Population Statistics for Particle Swarm Optimization on Problems Subject to Noise

by

Juan Rada-Vilela

A thesis submitted to the Victoria University of Wellington in fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science.

Victoria University of Wellington
2014
Abstract

Particle Swarm Optimization (PSO) is a metaheuristic where a swarm of particles explores the search space of an optimization problem to find good solutions. However, if the problem is subject to noise, the quality of the resulting solutions significantly deteriorates. The literature has attributed such a deterioration to particles suffering from inaccurate memories and from the incorrect selection of their neighborhood best solutions. For both cases, the incorporation of noise mitigation mechanisms has improved the quality of the results, but the analyses beyond such improvements often fall short of empirical evidence supporting their claims in terms other than the quality of the results. Furthermore, there is not even evidence showing the extent to which inaccurate memories and incorrect selection affect the particles in the swarm. Therefore, the performance of PSO on noisy optimization problems remains largely unexplored.

The overall goal of this thesis is to study the effect of noise on PSO beyond the known deterioration of its results in order to develop more efficient noise mitigation mechanisms. Based on the allocation of function evaluations by the noise mitigation mechanisms, we distinguish three groups of PSO algorithms as: single-evaluation, which sacrifice the accuracy of the objective values over performing more iterations; resampling-based, which sacrifice performing more iterations over better estimating the objective values; and hybrids, which merge methods from the previous two. With an empirical approach, we study and analyze the performance of existing and new PSO algorithms from each group on 20 large-scale benchmark functions subject to different levels of multiplicative Gaussian noise. Throughout the search process, we compute a set of 16 population statistics
that measure different characteristics of the swarms and provide useful information that we utilize to design better PSO algorithms.

Our study identifies and defines \textit{deception}, \textit{blindness} and \textit{disorientation} as three conditions from which particles suffer in noisy optimization problems. The population statistics for different PSO algorithms reveal that particles often suffer from large proportions of deception, blindness and disorientation, and show that reducing these three conditions would lead to better results. The sensitivity of PSO to noisy optimization problems is confirmed and highlights the importance of noise mitigation mechanisms.

The population statistics for single-evaluation PSO algorithms show that the commonly used evaporation mechanism produces too much disorientation, leading to divergent behaviour and to the worst results within the group. Two better algorithms are designed, the first utilizes probabilistic updates to reduce disorientation, and the second computes a centroid solution as the neighborhood best solution to reduce deception.

The population statistics for resampling-based PSO algorithms show that basic resampling still leads to large proportions of deception and blindness, and its results are the worst within the group. Two better algorithms are designed to reduce deception and blindness. The first provides better estimates of the personal best solutions, and the second provides even better estimates of a few solutions from which the neighborhood best solutions are selected. However, an existing PSO algorithm is the best within the group as it strives to asymptotically minimize deception by sequentially reducing both blindness and disorientation.

The population statistics for hybrid PSO algorithms show that they provide the best results thanks to a combined reduction of deception, blindness and disorientation. Amongst the hybrids, we find a promising algorithm whose simplicity, flexibility and quality of results questions the importance of overly complex methods designed to minimize deception.

Overall, our research presents a thorough study to design, evaluate and tune PSO algorithms to address optimization problems subject to noise.
List of Publications


Acknowledgments

I want to express my eternal gratitude to my supervisors, Dr Mark Johnston and Prof Mengjie Zhang, for their immense help and support throughout the course of my PhD studies. I have learnt so much from both that my words fall short to ever thank you enough. I also want to express my gratitude and appreciation to Prof Winston Seah for his vast efforts and patience throughout the first two years.

Thank you to the Victoria PhD Scholarship and Prof Mengjie Zhang for the financial support provided over the past three years.

Thank you to the Evolutionary Computation Research Group, the community of Venezuelans in Wellington, and my personal friends for their constant support and encouragement over the course of my PhD studies. In particular, thank you to Bing Xue, Harsha Raja, Maarten Lightband, Mario Alayón, and Su Nguyen.

The most special thank you goes to my family for their unconditional love and support at all times. In particular, thank you to my mother, my sisters and my nieces for always being my infinite source of love and inspiration.
CONTENTS

2.4.2 Network Topology ........................................... 25

2.5 Optimization Problems Subject to Noise .................. 28
  2.5.1 Classification according to Uncertainty ............... 29
  2.5.2 Direction of the Optimization Problem ............... 29
  2.5.3 Simulation-Based Optimization ......................... 30

2.6 PSO for Noisy Optimization Problems ..................... 33
  2.6.1 PSO under Local and Global Certainty ............... 34
  2.6.2 PSO with Single-Evaluation Methods ................. 34
  2.6.3 PSO with Resampling Methods ....................... 37

2.7 Related Work .................................................. 40
  2.7.1 PSO and Noise ............................................ 41
  2.7.2 PSO with Single-Evaluation Methods ................. 42
  2.7.3 PSO with Resampling Methods ....................... 44
  2.7.4 EAs and Noise ........................................... 47

2.8 Benchmark Functions ........................................ 50
  2.8.1 Basic Functions .......................................... 51
  2.8.2 Vector Operations ...................................... 51
  2.8.3 Separable Functions \(F_{01-03}\) ...................... 55
  2.8.4 Single-Group of \(m\) Non-Separable Functions \(F_{04-08}\) 55
  2.8.5 \(\frac{d}{2m}\) Groups of \(m\) Non-Separable Functions \(F_{09-13}\) 57
  2.8.6 \(\frac{d}{m}\) Groups of \(m\) Non-Separable Functions \(F_{14-18}\) 58
  2.8.7 Non-Separable Functions \(F_{19,20}\) .................... 59

2.9 Summary ....................................................... 60

3 Deception, Blindness and Disorientation .................... 63

3.1 Introduction .................................................. 63

3.2 Population Statistics for PSO ............................... 65
  3.2.1 Deception ............................................... 66
  3.2.2 Blindness ................................................ 66
  3.2.3 Disorientation .......................................... 67
  3.2.4 Regular Operation ...................................... 68
  3.2.5 Lifetime .................................................. 69
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.6</td>
<td>Quality of Results</td>
<td>70</td>
</tr>
<tr>
<td>3.3</td>
<td>Design of Experiments</td>
<td>70</td>
</tr>
<tr>
<td>3.4</td>
<td>Results and Discussions</td>
<td>73</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Population Statistics</td>
<td>73</td>
</tr>
<tr>
<td>3.4.2</td>
<td>PSO with and without Resampling</td>
<td>75</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Local and Global Certainty without Resampling</td>
<td>81</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Local and Global Certainty upon Resampling</td>
<td>84</td>
</tr>
<tr>
<td>3.4.5</td>
<td>Further Discussions</td>
<td>85</td>
</tr>
<tr>
<td>3.5</td>
<td>Summary</td>
<td>89</td>
</tr>
<tr>
<td>3.A</td>
<td>Population Statistics</td>
<td>92</td>
</tr>
<tr>
<td>4</td>
<td>Single-Evaluation Methods</td>
<td>103</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>103</td>
</tr>
<tr>
<td>4.2</td>
<td>Single-Evaluation PSO algorithms</td>
<td>105</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Stochastic vs Dynamic Optimization Problems</td>
<td>106</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Evaporation Factor</td>
<td>107</td>
</tr>
<tr>
<td>4.2.3</td>
<td>PSO with Probabilistic Updates</td>
<td>108</td>
</tr>
<tr>
<td>4.2.4</td>
<td>PSO with Average Neighborhoods</td>
<td>109</td>
</tr>
<tr>
<td>4.3</td>
<td>Design of Experiments</td>
<td>111</td>
</tr>
<tr>
<td>4.4</td>
<td>Results and Discussions</td>
<td>113</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Population Statistics</td>
<td>114</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Low and High Evaporation Factors</td>
<td>116</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Low and High Probabilities to Update</td>
<td>120</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Average Neighborhoods and Regular PSO</td>
<td>122</td>
</tr>
<tr>
<td>4.4.5</td>
<td>Further Discussions</td>
<td>124</td>
</tr>
<tr>
<td>4.5</td>
<td>Summary</td>
<td>127</td>
</tr>
<tr>
<td>5</td>
<td>Resampling Methods</td>
<td>145</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>145</td>
</tr>
<tr>
<td>5.2</td>
<td>Resampling-Based PSO Algorithms</td>
<td>147</td>
</tr>
<tr>
<td>5.2.1</td>
<td>PSO with Extended Equal Resampling</td>
<td>148</td>
</tr>
</tbody>
</table>
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2.2 PSO with ER and Allocations to the Top-N Solutions</td>
<td>149</td>
</tr>
<tr>
<td>5.3 Design of Experiments</td>
<td>150</td>
</tr>
<tr>
<td>5.4 Results and Discussions</td>
<td>152</td>
</tr>
<tr>
<td>5.4.1 Population Statistics</td>
<td>154</td>
</tr>
<tr>
<td>5.4.2 PSO-OCBA and PSO-ERN</td>
<td>155</td>
</tr>
<tr>
<td>5.4.3 PSO-EER1 and PSO-EER2</td>
<td>159</td>
</tr>
<tr>
<td>5.4.4 PSO-ERGC and PSO-ER</td>
<td>161</td>
</tr>
<tr>
<td>5.4.5 Further Discussions</td>
<td>162</td>
</tr>
<tr>
<td>5.5 Summary</td>
<td>163</td>
</tr>
<tr>
<td>5.A Population Statistics</td>
<td>167</td>
</tr>
<tr>
<td>6 Hybrid Methods</td>
<td>177</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>177</td>
</tr>
<tr>
<td>6.2 Hybrid PSO Algorithms</td>
<td>180</td>
</tr>
<tr>
<td>6.3 Design of Experiments</td>
<td>182</td>
</tr>
<tr>
<td>6.4 Results and Discussions</td>
<td>184</td>
</tr>
<tr>
<td>6.4.1 Population Statistics</td>
<td>185</td>
</tr>
<tr>
<td>6.4.2 Resampling-Based and Hybrid Algorithms</td>
<td>187</td>
</tr>
<tr>
<td>6.4.3 Hybrid Algorithms</td>
<td>192</td>
</tr>
<tr>
<td>6.4.4 Hybrid Algorithms and PSO-OCBA</td>
<td>193</td>
</tr>
<tr>
<td>6.4.5 Hybrid Algorithms and PSO-ERGC</td>
<td>195</td>
</tr>
<tr>
<td>6.4.6 Single-Evaluation against Resampling-Based and Hybrid Algorithms</td>
<td>196</td>
</tr>
<tr>
<td>6.4.7 Budget Allocation: from PSO-AN to PSO-EER1s</td>
<td>197</td>
</tr>
<tr>
<td>6.5 Summary</td>
<td>204</td>
</tr>
<tr>
<td>6.A Population Statistics</td>
<td>207</td>
</tr>
<tr>
<td>6.B Population Statistics for PSO-AN and Hybrid Variants</td>
<td>218</td>
</tr>
<tr>
<td>6.C Ackley Benchmark Functions</td>
<td>230</td>
</tr>
<tr>
<td>6.D Rastrigin Benchmark Functions</td>
<td>236</td>
</tr>
<tr>
<td>7 Conclusions</td>
<td>243</td>
</tr>
<tr>
<td>7.1 Achieved Objectives</td>
<td>244</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Particle Swarm Optimization .......................... 24
2.2 Simulated operation of Particle Swarm Optimization .... 26
2.3 Ring and star topologies ................................ 27
2.4 PSO with Evaporation (PSO-E) .......................... 36
2.5 Illustration of evaporation in a minimization problem ... 37
2.6 PSO with Equal Resampling (PSO-ER) ..................... 39
2.7 PSO with Optimal Computing Budget Allocation (PSO-OCBA) ... 41

3.1 Population statistics on benchmark function $F_{01}$ ...... 76
3.2 Conditions leading to blindness ............................ 80
3.3 Average number of disoriented particles ................... 81
3.4 Conditions leading to disorientation ......................... 82
3.5 Quality of Results ($A$, $B$, $E$) .......................... 94
3.6 Quality of Results ($C$, $D$) ............................... 95
3.7 Lifetime .................................................... 96
3.8 Deception .................................................. 97
3.9 Regular Operation, Blindness and Disorientation ........ 98
3.10 Regular Updates and Discards ............................. 99
3.11 Causes of Blindness ...................................... 100
3.12 Causes of Disorientation .................................. 101

4.1 PSO with Probabilistic Updates ............................ 110
4.2 PSO with Average Neighborhoods ......................... 111
4.3 Population statistics on benchmark function $F_{01}$ ....... 117
LIST OF FIGURES

4.4 Ranked Deception........................................... 123
4.5 Quality of Results (A, B, E)................................. 134
4.6 Quality of Results (C, D).................................. 135
4.7 Binary Deception........................................... 136
4.8 Ranked Deception........................................... 137
4.9 Regular Operation, Blindness and Disorientation........ 138
4.10 Regular Updates and Discards.............................. 139
4.11 Causes of Blindness........................................ 140
4.12 Causes of Disorientation................................... 141
4.13 Lifetime.................................................... 142
4.14 Effect of Disorientation and Blindness.................... 143
4.15 Optimization Curves........................................ 144

5.1 PSO with Extended Equal Resampling (PSO-EER)........... 149
5.2 PSO with Equal Resampling and Allocations to the Top-N Solutions (PSO-ERN). .......................................................... 151
5.3 Population statistics on benchmark function \( F_{13} \)........ 156
5.4 Quality of Results (A, B, E)................................. 168
5.5 Quality of Results (C, D).................................. 169
5.6 Binary Deception........................................... 170
5.7 Ranked Deception........................................... 171
5.8 Regular Operation, Blindness and Disorientation.......... 172
5.9 Regular Updates and Discards.............................. 173
5.10 Causes of Blindness........................................ 174
5.11 Causes of Disorientation................................... 175
5.12 Lifetime.................................................... 176

6.1 The Centroid Solution......................................... 181
6.2 Hybrid PSO-EER............................................... 182
6.3 Hybrid PSO-ERN............................................. 183
6.4 Hybrid PSO-OCBA............................................ 183
6.5 Population statistics on benchmark function \( F_{13} \)........ 189
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.6</td>
<td>Quality of Results for PSO-AN, its hybrid variants and hybrid PSO-OCBA on $F_{13}$</td>
<td>201</td>
</tr>
<tr>
<td>6.7</td>
<td>Quality of Results (A, B, E)</td>
<td>208</td>
</tr>
<tr>
<td>6.8</td>
<td>Quality of Results (C, D)</td>
<td>209</td>
</tr>
<tr>
<td>6.9</td>
<td>Binary Deception</td>
<td>210</td>
</tr>
<tr>
<td>6.10</td>
<td>Ranked Deception</td>
<td>211</td>
</tr>
<tr>
<td>6.11</td>
<td>Regular Operation, Blindness and Disorientation</td>
<td>212</td>
</tr>
<tr>
<td>6.12</td>
<td>Regular Updates and Discards</td>
<td>213</td>
</tr>
<tr>
<td>6.13</td>
<td>Causes of Blindness</td>
<td>214</td>
</tr>
<tr>
<td>6.14</td>
<td>Causes of Disorientation</td>
<td>215</td>
</tr>
<tr>
<td>6.15</td>
<td>Lifetime</td>
<td>216</td>
</tr>
<tr>
<td>6.16</td>
<td>Effect of Disorientation and Blindness</td>
<td>217</td>
</tr>
<tr>
<td>6.17</td>
<td>Quality of Results (A, B, E)</td>
<td>220</td>
</tr>
<tr>
<td>6.18</td>
<td>Quality of Results (C, D)</td>
<td>221</td>
</tr>
<tr>
<td>6.19</td>
<td>Binary Deception</td>
<td>222</td>
</tr>
<tr>
<td>6.20</td>
<td>Ranked Deception</td>
<td>223</td>
</tr>
<tr>
<td>6.21</td>
<td>Regular Operation, Blindness and Disorientation</td>
<td>224</td>
</tr>
<tr>
<td>6.22</td>
<td>Regular Updates and Discards</td>
<td>225</td>
</tr>
<tr>
<td>6.23</td>
<td>Causes of Blindness</td>
<td>226</td>
</tr>
<tr>
<td>6.24</td>
<td>Causes of Disorientation</td>
<td>227</td>
</tr>
<tr>
<td>6.25</td>
<td>Lifetime</td>
<td>228</td>
</tr>
<tr>
<td>6.26</td>
<td>Effect of Disorientation and Blindness</td>
<td>229</td>
</tr>
<tr>
<td>6.27</td>
<td>Quality of Results on $F_{03}$</td>
<td>232</td>
</tr>
<tr>
<td>6.28</td>
<td>Quality of Results on $F_{06}$</td>
<td>233</td>
</tr>
<tr>
<td>6.29</td>
<td>Quality of Results on $F_{11}$</td>
<td>234</td>
</tr>
<tr>
<td>6.30</td>
<td>Quality of Results on $F_{16}$</td>
<td>235</td>
</tr>
<tr>
<td>6.31</td>
<td>Quality of Results on $F_{02}$</td>
<td>238</td>
</tr>
<tr>
<td>6.32</td>
<td>Quality of Results on $F_{05}$</td>
<td>239</td>
</tr>
<tr>
<td>6.33</td>
<td>Quality of Results on $F_{10}$</td>
<td>240</td>
</tr>
<tr>
<td>6.34</td>
<td>Quality of Results on $F_{15}$</td>
<td>241</td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Separability of the benchmark functions</td>
<td>50</td>
</tr>
<tr>
<td>2.2</td>
<td>Composition of the benchmark functions</td>
<td>52</td>
</tr>
<tr>
<td>2.3</td>
<td>Equations of basic functions</td>
<td>53</td>
</tr>
<tr>
<td>3.1</td>
<td>Parameter settings</td>
<td>72</td>
</tr>
<tr>
<td>3.2</td>
<td>Summary of statistical significance tests</td>
<td>87</td>
</tr>
<tr>
<td>3.3</td>
<td>Average population statistics</td>
<td>89</td>
</tr>
<tr>
<td>4.1</td>
<td>Evaporation factors after unsuccessful iterations</td>
<td>109</td>
</tr>
<tr>
<td>4.2</td>
<td>Parameter settings</td>
<td>112</td>
</tr>
<tr>
<td>4.3</td>
<td>Summary of statistical significance tests</td>
<td>125</td>
</tr>
<tr>
<td>4.4</td>
<td>Summary of contradictory statistical significance tests</td>
<td>126</td>
</tr>
<tr>
<td>4.5</td>
<td>Average population statistics</td>
<td>128</td>
</tr>
<tr>
<td>5.1</td>
<td>Parameter settings</td>
<td>153</td>
</tr>
<tr>
<td>5.2</td>
<td>Summary of statistical significance tests</td>
<td>164</td>
</tr>
<tr>
<td>5.3</td>
<td>Average population statistics</td>
<td>165</td>
</tr>
<tr>
<td>6.1</td>
<td>Parameter settings</td>
<td>185</td>
</tr>
<tr>
<td>6.2</td>
<td>Average population statistics</td>
<td>188</td>
</tr>
<tr>
<td>6.3</td>
<td>Summary of statistical tests between resampling-based and hybrid PSO algorithms</td>
<td>190</td>
</tr>
<tr>
<td>6.4</td>
<td>Summary of statistical tests between hybrid PSO algorithms</td>
<td>192</td>
</tr>
<tr>
<td>6.5</td>
<td>Summary of statistical tests between hybrid PSO algorithms and PSO-OCBA</td>
<td>193</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>6.6</td>
<td>Summary of statistical tests between hybrid PSO algorithms and PSO-ERGC</td>
<td>196</td>
</tr>
<tr>
<td>6.7</td>
<td>Summary of statistical tests of PSO-AN against resampling-based and hybrid algorithms</td>
<td>198</td>
</tr>
<tr>
<td>6.8</td>
<td>Benchmarks on which PSO-AN is better than PSO-ERGC</td>
<td>199</td>
</tr>
<tr>
<td>6.9</td>
<td>Average population statistics for PSO-AN and hybrid algorithms</td>
<td>201</td>
</tr>
<tr>
<td>6.10</td>
<td>Average population statistics on $F_{03}$</td>
<td>232</td>
</tr>
<tr>
<td>6.11</td>
<td>Average population statistics on $F_{06}$</td>
<td>233</td>
</tr>
<tr>
<td>6.12</td>
<td>Average population statistics on $F_{11}$</td>
<td>234</td>
</tr>
<tr>
<td>6.13</td>
<td>Average population statistics on $F_{16}$</td>
<td>235</td>
</tr>
<tr>
<td>6.14</td>
<td>Average population statistics on $F_{02}$</td>
<td>238</td>
</tr>
<tr>
<td>6.15</td>
<td>Average population statistics on $F_{05}$</td>
<td>239</td>
</tr>
<tr>
<td>6.16</td>
<td>Average population statistics on $F_{10}$</td>
<td>240</td>
</tr>
<tr>
<td>6.17</td>
<td>Average population statistics on $F_{15}$</td>
<td>241</td>
</tr>
</tbody>
</table>
List of Acronyms

Chapter 3
PSO Particle Swarm Optimization
PSO-LC PSO with Local Certainty
PSO-GC PSO with Global Certainty
PSO-ER PSO with Equal Resampling
PSO-ERLC PSO with Equal Resampling and Local Certainty
PSO-ERGC PSO with Equal Resampling and Global Certainty

Chapter 4
PSO-E PSO with Evaporation
PSO-LE PSO with Low Evaporation
PSO-HE PSO with High Evaporation
PSO-PU PSO with Probabilistic Updates
PSO-LP PSO with a Low Probability to Update
PSO-HP PSO with a High Probability to Update
PSO-AN PSO with Average Neighborhoods

Chapter 5
PSO-EER PSO with Extended Equal Resampling
PSO-EER1 PSO-EER with One Evaluation to Personal Best Solutions
PSO-EER2 PSO-EER with Two Evaluations to Personal Best Solutions
PSO-ERN PSO with Equal Resampling and Allocations to Top-N Solutions
PSO-OCBA PSO with Optimal Computing Budget Allocation
Chapter 6

PSO-EER1\(^5\)  PSO-EER1 with Centroid Solution from Best Five Solutions
PSO-ERN\(^5\)  PSO-ERN with Centroid Solution from Best Five Solutions
PSO-OCBA\(^5\)  PSO-OCBA with Centroid Solution from Best Five Solutions
PSO-AN\(_{xy}\)  PSO-AN with \(x\) Evaluations to the Current Solutions and \(y\) Evaluations to the Personal Best Solutions

PSO-AN\(_{11}\)  PSO-AN\(_{xy}\) with \(x = 1\) and \(y = 1\)
PSO-AN\(_{21}\)  PSO-AN\(_{xy}\) with \(x = 2\) and \(y = 1\)
PSO-AN\(_{31}\)  PSO-AN\(_{xy}\) with \(x = 3\) and \(y = 1\)
PSO-AN\(_{41}\)  PSO-AN\(_{xy}\) with \(x = 4\) and \(y = 1\)
PSO-AN\(_{51}\)  PSO-AN\(_{xy}\) with \(x = 5\) and \(y = 1\)
LIST OF ACRONYMS
Chapter 1

Introduction

Particle Swarm Optimization (PSO) is a metaheuristic in which a swarm of particles explores the search space of an optimization problem to find good solutions. Designed by Eberhart and Kennedy [42, 71], it takes inspiration from swarming theory [93] and social models [102] by having particles interact with each other in order to improve the quality of their solutions. Each particle has a position that encodes a potential solution to the problem at hand, a velocity that will change its position at the next iteration, and a memory to remember where it found the best solution. Particles start at random positions and iteratively adjust their velocities such that they become partially attracted to the positions of the best solutions found by themselves and their neighbors. At each step, particles evaluate their newly found positions and decide whether to utilize them to replace their memories. This is the regular PSO algorithm that has been adapted to address many optimization problems in different fields of research [6, 35, 43, 114, 115].

One characteristic that has remained largely unexplored in PSO is the deterioration of its performance when optimization problems are subject to noise. In this type of problem, the objective values that determine the quality of the solutions are corrupted by the effect of sampling noise, hence resulting in differently estimated objective values every time the solutions
are evaluated. As a consequence, particles eventually fail to distinguish good from bad solutions, leading to other issues that will ultimately end up deteriorating the quality of the best solutions found. Two such issues have been recognized in the literature as the problem of particles having inaccurate memories \cite{30,31,32,50} and not selecting the true best solutions from their neighborhoods \cite{7,106,113,154}. The first issue is that particles with inaccurate memories may store solutions whose very inaccurately estimated objective values will potentially prevent them from finding better solutions, or at least affect the frequency at which they do. The second issue is that particles will select the best solution from their neighbors based on their estimated objective values, and hence they will most likely not select the true best solution amongst them. In both cases, particles end up partially attracted towards sub-optimal regions of the search space, which reflects on the poor quality of the solutions found. However, it is not certain as to what extent do the inaccurate memories or the incorrect selection affect the particles in PSO, and which is more important to address first.

The deterioration of the quality of the solutions found by PSO on optimization problems subject to noise prompts the need to incorporate noise mitigation mechanisms in order to prevent (or at least reduce) such a deterioration. We distinguish two conceptually different approaches of noise mitigation mechanisms based on their use of the computational budget of function evaluations available to the PSO algorithm. On the one hand, noise mitigation mechanisms based on resampling methods consist of performing multiple evaluations of each solution in order to better estimate its objective value by taking a sample mean over the evaluations \cite{7,106,113,154}. On the other hand, noise mitigation mechanisms based on single-evaluation methods do not perform additional evaluations to the solutions, but instead deal with the solutions having their objective values estimated from a single evaluation as the PSO algorithm dictates \cite{30,31,32,50,125}. As such, considering a fixed and limited computational budget of func-
tion evaluations, resampling methods better estimate the objective values of the solutions at the cost of performing fewer iterations, whereas single-evaluation methods perform more iterations at the cost of dealing with solutions whose objective values are (very) inaccurately estimated. While previous works have compared PSO algorithms with different noise mitigation mechanisms [7 30 31 32 50 106 154], these have always utilized the same method, that is, either resampling or single-evaluation. Consequently, it is not certain whether it is worthwhile to sacrifice the exploration of new solutions over improving the accuracy of the existing solutions, let alone the possibility of merging approaches from the different methods.

1.1 Motivation

Real-world optimization problems are often subject to uncertainty [3 4 12 56 57 58 69] as variables may be affected by imprecise measurements, modeled by probability distributions, or just corrupted by other factors such as communication errors. In either case, uncertainty is an inherent characteristic of the problem and therefore needs to be considered when tailoring metaheuristics to find good solutions. Noise is a class of uncertainty that corrupts the objective values of the solutions at each evaluation, and it has been shown to significantly deteriorate the performance of different metaheuristics such as Genetic Algorithms [62 89], Evolutionary Strategies [5 111], Differential Evolution [74 94], and others [12 17 29 139]. The effect of noise on PSO is not different as the quality of its solutions also deteriorate in the presence of noise [7 30 31 32 50 106 154], but it may be even worse because, unlike the individuals from other evolutionary algorithms, particles have a memory where they store the estimated best solutions. Consequently, the effect of noise on each solution will persist over the iterations potentially driving the swarm towards stagnation on what may not even be a local optimum.
Previous attempts to incorporate noise mitigation mechanisms into PSO have shown significant improvements upon the quality of its results [7, 30, 31, 32, 50, 106, 154]. However, the analyses of such improvements often fall short of empirical evidence supporting their claims about the underlying reasons for their achievements in terms other than the quality of the results, not to mention that the experiments have been performed only on very few benchmark functions with no more than ten dimensions. More importantly, there is not even evidence showing the extent to which inaccurate memories affect the particles in the swarm or the frequency at which particles fail to select their true neighborhood best solutions. Therefore, given the lack of such evidence and the state of the art, we want to investigate the effect of noise on PSO beyond the deterioration of its results on a variety of large-scale benchmark functions. Furthermore, we expect that such information will not only help to choose the best noise mitigation mechanism, but will also help to design even better ones.

The intricacies of understanding the effect of noise on PSO, beyond the quality of its results, presents a significant challenge due to the intractability caused by the interactions between the particles, the pseudo-randomness associated with the exploration of the search space, the characteristics of the particles, and the characteristics of the swarm itself. Therefore, instead of following a theoretical study like in [26, 104, 105, 146], we will perform an empirical study of PSO algorithms on the set of 20 large-scale benchmark functions designed in [141], whose objective values are subject to the effect of different levels of multiplicative Gaussian noise. The choice of benchmark functions provides not only different challenges to optimization algorithms but also a common reference specifically designed for the scientific community. The choice of Gaussian noise is a reasonable assumption because it accurately models many noisy domains [89, 90, 91]. The study will cover existing single-evaluation and resampling-based PSO algorithms as well as the new single-evaluation and resampling-based PSO algorithms that we propose. Furthermore,
considering the tradeoffs of both methods, we are encouraged to create the first PSO algorithms that utilize hybrid methods to potentially bring the best of both approaches and thereby significantly improve the quality of the results.

1.2 Research Goals

The overall goal of this thesis is to study and understand the effect of noise on PSO beyond the known deterioration of the quality of its results in order to systematically design more effective noise mitigation mechanisms that will reduce such a deterioration. Specifically, this thesis will address the following research questions:

(i) To what extent do the inaccurate memories and the incorrect selection of the neighborhood best solutions affect particles in PSO?

Particles in optimization problems subject to noise are prone to have inaccurate memories and to fail to select their true neighborhood best solutions. Both issues have been addressed independently from each other with algorithms that indeed find better solutions than the regular PSO on such problems \[7, 30, 31, 32, 50, 106, 154\]. However, the extents to which these issues affect the particles have never been reported in the literature. Consequently, the importance of specifically addressing one issue over the other is not known, and hence it is uncertain which method of noise mitigation mechanism would perform better.

(ii) Which of the single-evaluation methods yields the best quality of results to optimization problems subject to noise?

The evaporation mechanism \[31\] is a single-evaluation method that, incorporated into PSO, helps particles to overcome the stagnation caused by having solutions stored in memory with very inaccurately estimated objective values. While other algorithms have followed
similar approaches and reported better solutions than those obtained without the evaporation mechanism [30, 31, 32, 50, 51], PSO with Evaporation (PSO-E) and its variants have the potential to make the swarms exhibit divergent behaviours that will deteriorate the quality of the results. This threatening condition has been ignored thus far because research has only focused on justifying their approaches based on the quality of the solutions found without considering the performance during the search process. Furthermore, there is no empirical evidence on exactly how PSO-E and its variants mitigate the effect of noise. Hence, it is important to further investigate single-evaluation methods in order to better understand their advantages and disadvantages, and thus design more effective single-evaluation PSO algorithms.

(iii) Which of the resampling methods yields the best quality of results to optimization problems subject to noise?

The Optimal Computing Budget Allocation (OCBA) [25] is a resampling method that, incorporated into PSO, aims to maximize the correct selection of the neighborhood best solutions [7, 106, 154]. Different works have reported that PSO with Optimal Computing Budget Allocation (PSO-OCBA) yields a quality of results superior to any other resampling-based PSO algorithm [7, 106, 154], and this has been attributed to the higher probability of particles correctly selecting their neighborhood best solutions [106, 154] and generally making correct decisions regarding the replacement of their memories [2]. However, just like in single-evaluation PSO algorithms, there is no empirical evidence showing how often particles fail to select their true neighborhood best solutions, let alone showing the hypothetical improvements had the particles always correctly selected the true best solutions. Hence, it is important to further investigate resampling-based methods in order to better understand their advantages and disadvantages, and thus design more effective resampling-
1.2. RESEARCH GOALS

-based PSO algorithms.

(iv) Would a hybrid between single-evaluation and resampling methods yield a better quality of results than that obtained by both methods separately? While single-evaluation and resampling methods provide opposite tradeoffs, that does not imply that they cannot be merged to work together. On the contrary, the efforts of single-evaluation methods to address the inaccurate memories could be utilized to further reinforce the efforts of resampling methods and vice versa.

The research questions will be answered by addressing the following objectives which, together as a whole, will fulfill our research goal.

1. Design a set of population statistics to measure the extents to which inaccurate memories and incorrect selection of the neighborhood best solutions affect the particles throughout the search process on optimization problems subject to noise.

The design of a set of population statistics first requires a formal definition of what inaccurate memories and incorrect selection of the neighborhood best solutions represent. Once these conditions are clearly defined, we will quantify their presence throughout the search process for the regular PSO and PSO with Equal Resampling (PSO-ER) with the goal of understanding the effect of different levels of noise on the particles. Besides quantifying these conditions, we will design other population statistics to measure different characteristics of the swarm that will help not only to explain the effect of noise beyond the deterioration of the quality of the results, but also to reveal correlations and interactions between the different statistics. Furthermore, we will also compare the population statistics under the assumptions of local and global certainty, that is, when particles always make correct decisions with respect to their personal and neighborhood best solutions (respectively). In doing so, we will be
able to distinguish the importance of addressing the problem of inaccurate memories and incorrect selection of the neighborhood best solutions.

2. Study the population statistics for PSO algorithms incorporating single-evaluation methods as noise mitigation mechanisms on optimization problems subject to noise.

Single-evaluation methods refer to PSO algorithms that address the effect of noise without performing additional function evaluations to the solutions. The first of these algorithms was PSO-E, which was proposed to gradually encourage particles to forget the quality of their personal best solutions. Reports have suggested that PSO-E succeeds at finding better solutions than the regular PSO in different stochastic and dynamic optimization problems [30, 31, 32, 50, 51]. However, PSO-E depends on an evaporation factor whose value is not only determined empirically, but also puts the swarm at risk of exhibiting a divergent behaviour. Hence, we will design a method to better estimate a priori the evaporation factor for PSO-E, and will also propose a new PSO with Probabilistic Updates (PSO-PU) to reduce the risk of divergence. In addition, we will explore a different approach to mitigate the effect of noise by averaging the neighborhood best solutions. The performance of the algorithms will be compared with respect to their population statistics in order to properly support our discussions with empirical results.

3. Study the population statistics for PSO algorithms incorporating resampling methods as noise mitigation mechanisms on optimization problems subject to noise.

Resampling methods mitigate the effect of noise by performing multiple evaluations to the solutions and better estimating their true objective values with a sample mean over the evaluations. PSO-ER is the most basic resampling-based PSO algorithm given that each of
the current solutions is subject to the same number of re-evaluations, thereby better estimating their true objective values. However, two disadvantages of such an allocation method is that, first, it sacrifices the potential accuracy of the better solutions to improve that of the worse solutions, and second, particles still have the problem of inaccurate memories. Therefore, we will explore the advantages of allocating the computational budget differently such that evaluations can also be allocated to the personal best solutions as an attempt to simultaneously address the problem of inaccurate memories. As for allocation methods, previous works have suggested that PSO-OCBA yields the best quality of results amongst other resampling-based PSO algorithms \[7, 106, 154\], for which we will compute its population statistics in order to find the characteristics that makes it superior, understand the benefits of its rather complex internal operation, and perhaps provide some insights that could help the design of simpler and better resampling-based PSO algorithms.

4. Study the population statistics for PSO algorithms incorporating hybrid methods as noise mitigation mechanisms on optimization problems subject to noise.

Hybrid methods are a new approach that we propose to improve the quality of the results of PSO on optimization problems subject to noise. Hybrid methods consist of merging approaches from single-evaluation and resampling methods into PSO, thereby aiming to obtain the best of the two approaches and potentially find better solutions. Different hybrid PSO algorithms will be proposed based on the best single-evaluation and resampling-based algorithms studied, and we will contrast their population statistics against the population statistics for their corresponding parts independently. Furthermore, as part of this multiple comparison, the best single-evaluation and resampling-based algorithms will also be compared to deter-
mine which method (if any) is better suited to address optimization problems subject to noise.

### 1.3 Major Contributions

This thesis makes the following major contributions.

1. This thesis identifies and clearly defines *deception*, *blindness* and *disorientation* as three conditions from which particles in PSO suffer when optimization problems are subject to noise. Particles suffer from *deception* when they fail to select their true neighborhood best solutions, from *blindness* when they miss out on opportunities to improve upon their personal best solutions, and from *disorientation* when they mistakenly prefer worse solutions. While deception is just a new name to refer to the incorrect selection of neighborhood best solutions, blindness and disorientation are new concepts that clearly distinguish the problem of inaccurate memories. In addition, this thesis presents a set of *population statistics* to measure the extents to which these conditions and other characteristics of the swarm affect the particles throughout the search process. The population statistics computed for the regular PSO algorithm and PSO-ER reveal that particles suffer from large proportions of deception and blindness, confirm that PSO is sensitive to optimization problems subject to noise, highlights that it is important to incorporate resampling methods into PSO, and shows that reducing blindness and disorientation should be addressed first before reducing deception. This contribution has been submitted for publication in [119], and partial results have been published in [127].

2. This thesis studies the population statistics for PSO algorithms with single-evaluation methods to mitigate the effect of noise. Specifically, this thesis presents a method to better estimate the evapora-
1.3. MAJOR CONTRIBUTIONS

tion factors for PSO-E, two new algorithms that find better solutions than PSO-E, and shows that the regular PSO without a noise mitigation mechanism finds better solutions than PSO-E. The method to estimate the evaporation factors utilizes the number of unsuccessful iterations before particles are expected to replace their personal best solutions with the current solutions. The newly proposed PSO-PU algorithm provides more exploitation of the search space, which in turn prevents divergent behaviour and helps finding better solutions. However, the solutions found with the regular PSO are still better than those found with PSO-PU mostly because particles suffer less from disorientation. In addition, the newly proposed PSO with Average Neighborhoods (PSO-AN) blurs the effect of noise on the selection of the neighborhood best solution by selecting instead a new solution that is created computing the centroid of the solutions within the neighborhoods. Hence, PSO-AN is able to find better solutions than the regular PSO because the centroid solutions are better on average than any other solution particles could select given their inaccurately estimated objective values. This contribution has been submitted for publication in [120], and a preliminary study has been published in [126].

3. This thesis studies the population statistics for PSO algorithms with resampling methods to mitigate the effect of noise. Specifically, this thesis confirms that PSO-ER finds better solutions than the regular PSO, presents a PSO with Extended Equal Resampling (PSO-EER) to better estimate the personal best solutions, the PSO with Equal Resampling and Allocations to the Top-$N$ Solutions (PSO-ERN), and confirms PSO-OCBA as the best resampling-based PSO algorithm. The better quality of results of PSO-ER over regular PSO is mostly due to particles suffering less from deception and blindness. PSO-EER finds better solutions than PSO-ER because its particles suffer even less from blindness and deception. PSO-ERN finds better solutions
than PSO-EER because it selects better neighborhood best solutions and has smaller proportions of disorientation. Lastly, PSO-OCBA finds the best solutions amongst the algorithms because it reduces the proportions of blindness and disorientation of the most important particles, and thereby minimizes the proportions of deception in the swarm. This contribution has been accepted for publication in [121], and partial results have been published in [123].

4. This thesis studies the population statistics for PSO algorithms with hybrid methods to mitigate the effect of noise. The hybrid methods consist of utilizing the centroid solutions of PSO-AN within the resampling-based PSO algorithms. Different from PSO-AN, the centroid solutions in hybrid algorithms are created from the estimated best five solutions within the neighborhoods given that their objective values are already better estimated via resampling methods. In general, the hybrid PSO algorithms find significantly better solutions than their purely resampling-based and single-evaluation counterparts mostly thanks to the better quality of the neighborhood best solutions. The hybrid PSO-OCBA finds the best solutions amongst the hybrid algorithms because the centroid solutions are computed from better solutions than those in the other algorithms. However, the hybrid PSO-EER has a much simpler design than the hybrid PSO-OCBA and the quality of its results is mostly not significantly different. Therefore, given the simplicity and flexibility of the hybrid PSO-EER, we find it as a promising algorithm as its quality of results can be further improved by adjusting the algorithm according to its population statistics.

1.4 Thesis Organization

The remainder of this thesis is organized as follows.
1.4. THESIS ORGANIZATION

Chapter 2 presents a literature review covering the fundamentals of PSO, together with general topics on evolutionary computation, optimization problems subject to noise, works related to this thesis, and the benchmark functions on which the algorithms in this thesis will be evaluated.

Chapter 3 defines deception, blindness and disorientation, develops a set of population statistics, and studies the population statistics for the regular PSO and PSO-ER, both additionally under the assumptions of local and global certainty. The set of population statistics measures the proportions of deception, blindness and disorientation, the proportions of blindness caused by memory and by the environment, the proportions of disorientation caused by memory and by the environment, the proportions of regular operations including those of regular updates and regular discards, the quality of results and the lifetime of the swarm.

Chapter 4 focuses on single-evaluation methods in PSO, proposes the PSO-PU and PSO-AN algorithms, designs a method to better estimate the evaporation factors for PSO-E, and studies the population statistics for PSO-AN, PSO-E and PSO-PU utilizing low and high evaporation factors and update probabilities. In addition, the following four population statistics are designed: ranked deception to measure the quality of the neighborhood best solution with respect to the other solutions, optimization curves to measure the average quality of results at each iteration, deterioration caused by disorientation to measure the effect of disorientation on the objective values, and the hypothetical improvements missed due to blindness to measure the effect of blindness on the objective values.

Chapter 5 focuses on resampling methods in PSO, proposes PSO-EER and PSO-ERN, and studies the population statistics for PSO-EER, PSO-ERN and PSO-OCBA. Two different allocations of the computational budget are explored for PSO-EER, thus studying the effect of different accuracy tradeoffs between the current and personal best solutions.

Chapter 6 focuses on hybrid methods in PSO, where resampling-based PSO algorithms are combined with the centroid solution of the single-
evaluation PSO-AN. The population statistics for the hybrid PSO algorithms are compared to those for their purely single-evaluation and resampling-based counterparts. In addition, the population statistics for single-evaluation and resampling-based PSO algorithms are also compared.

Finally, Chapter 7 presents the main conclusions from this thesis, provides further discussions on the topics covered, and suggests promising research topics for future works.
Chapter 2

Literature Review

This chapter starts with an overview of Evolutionary Computation and its two main families of algorithms, namely Evolutionary Algorithms and Swarm Intelligence, in order to then focus on the fundamentals of Particle Swarm Optimization (PSO) and its operation. The chapter then introduces optimization problems subject to noise, their classification from two different perspectives, the direction of optimization problems based on the characteristics of noise, and an overview of simulation-based optimization within the field of Operations Research, where different real-world problems subject to noise are being addressed. The chapter then proceeds with the most relevant PSO algorithms proposed to address optimization problems subject to noise, which we classify into two groups according to the noise mitigation mechanism as single-evaluation and resampling-based PSO algorithms. The remainder of this chapter presents related work on PSO and other metaheuristics for optimization problems subject to noise, followed by the set of benchmark functions on which the algorithms in this thesis will be evaluated.
CHAPTER 2. LITERATURE REVIEW

2.1 Overview of Evolutionary Computation

Evolutionary Computation (EC) is a field of research within Artificial Intelligence that comprises population-based stochastic algorithms inspired by Nature to solve optimization problems by approximation. These algorithms are based on population models where individuals encode potential solutions to the problem at hand. Such solutions are randomly created at first and then iteratively improved by means of different search operators until a satisfactory solution is found within the population or another criterion is met.

The application of EC algorithms requires a formal definition of the optimization problem as an objective function $f : x \rightarrow \mathbb{R}^m$ that determines the quality of every possible solution $x$ in an $m$-dimensional objective space. The goal of EC algorithms is to explore the solution search space of such a clearly defined optimization problem in order to find a solution $x$ that minimizes (or maximizes) its $m$ objective values. Specifically, our focus is on single-objective optimization problems ($m = 1$) and, without any loss of generality, we refer only to minimization problems.

The design of EC algorithms consists mainly of selecting a proper representation of the solutions and a set of search operators to modify them. The solutions are represented by data structures such as vectors and trees, where all of its elements are fundamental pieces to construct a specific solution. The search operators modify the solutions at each iteration in order to explore the solution search space. As such, search operators ultimately determine the tradeoff between the exploration and exploitation of the search space according to the magnitude of the changes they perform. On the one hand, search operators that produce large changes to the solutions lead to the exploration of different regions of the solution search space at the cost of potentially missing good solutions in between. On the other hand, search operators that produce small changes to the solutions lead to potentially better solutions within a small region of the
search space at the cost of missing the exploration of other regions which may have better solutions.

EC algorithms can be classified according to their source of inspiration, where evolutionary algorithms and swarm intelligence are the two main categories.

2.2 Evolutionary Algorithms

Evolutionary Algorithms (EAs) are inspired by the Darwinian principle of natural selection or survival of the fittest [48, 118]. In these algorithms, each individual represents a single solution whose objective value (or fitness) influences its probability of survival for the next iteration (or generation). Those individuals with better objective values will be more likely to survive the next iteration and will contribute via reproduction to the creation of new individuals whose solutions will potentially be better.

Typical algorithms in this category are genetic algorithms [63, 66], genetic programming [73, 116], evolutionary strategies [11, 129], and evolutionary programming [53, 54], all of which have in common the following three search operators. The selection operator chooses the individuals from the population that will either reproduce or survive the next iteration, favouring those with better objective values. The crossover operator creates new offspring solutions based on the selected individuals which act as parents. The mutation operator introduces random modifications to the new offspring in order to increase the diversity of the population. The fundamentals of these three operators are the same, but each needs to be specialized to deal with the intrinsic characteristics of the algorithms such as the representation of the solutions.
2.2.1 Genetic Algorithms

Genetic Algorithms (GAs) [63, 66] is a metaheuristic in which the individuals of a population are referred to as chromosomes. Each chromosome encodes a potential solution to the problem at hand in a vector of bits. At each iteration, a new population of chromosomes is created by means of selection, reproduction and mutation. The selection operator selects two chromosomes from the population to serve as parents. The reproduction operator utilizes the selected parents to create two new chromosomes as offspring that will be added to the new population. Finally, the mutation operator performs changes to the new chromosomes. These operators are utilized in that sequence until a new population of chromosomes is created.

The basic configuration of a GA utilizes chromosomes with fixed-length vectors and the following search operators. The roulette-wheel selection [61] selects the parents with probabilities proportional to their objective values. The single-point crossover operator creates two offspring by combining the parent vectors with respect to a randomly selected point. The bit mutation operator probabilistically swaps the bits of the offspring. Other configurations that have been proposed in the literature are n-tournament selection [90], where n chromosomes from the population are selected and the best therein becomes a parent; representation of the solutions with vectors of real numbers [150] requiring specialized search operators; variable-length vectors [66]; and multiple-point crossover operators [66].

2.2.2 Genetic Programming

Genetic Programming (GP) [73] is a metaheuristic in which the individuals of a population are referred to as programs. In the tree-based GP, each program is encoded as an expression tree whose internal nodes are functions over their respective child nodes, and the terminal nodes are variables or constants. At each iteration, a new population of programs is created by
2.2. EVOLUTIONARY ALGORITHMS

means of the selection, reproduction and mutation operators (similar to GAs). However, given the different encoding of the solutions, the search operators are specialized to deal with expression trees. Specifically, the reproduction and mutation operators need to ensure that the resulting offsprings are valid expression trees, and may also need to control their resulting size in order to avoid the bloating problem as trees become too large to handle and hardly incorporate relevant information [73].

A basic configuration of GP requires the definition of a set of functions for the internal nodes and utilizes the following search operators. The roulette-wheel selection [116] selects programs according to probabilities proportional to their objective values. The subtree crossover operator [73] creates two offspring by merging two randomly selected subtrees from the parents. The subtree mutation randomly selects a subtree of the parent and replaces it with a new subtree. Other configurations proposed in the literature involve approaches to initialize the population of programs [73], n-tournament selection [116], and different strategies to address the bloating problem [73]. Other variants include linear, cartesian and grammar-based GP [14, 86, 92]. Unlike other EC algorithms, GP is also often referred to as a hyperheuristic [21].

2.2.3 Evolutionary Strategies

Evolutionary Strategies (ES) [11, 129] is a metaheuristic in which the individuals of a population encode solutions to the problem at hand as vectors of bits, integers or real numbers. At each iteration, the population of individuals undergo specialized search operators for selection, reproduction and mutation. The selection operator selects $\rho$ individuals from the population to become parents. The reproduction operator creates a single offspring from the selected parents. The mutation operator introduces changes to the offspring. The main difference of ES from GA is that its individuals encode additional information relative to external parameters.
(or strategies) that will provide self-adaptation up to a certain extent. However, the reproduction and mutation operators on these strategies may need to handle special cases as, for example, the individuals may encode probabilities which operators will need to ensure always remain within $[0.0, 1.0]$ in the resulting offspring.

ES often utilizes the following notation $(\mu/\rho{\dagger}\lambda)$, where $\mu$ is the size of the population, $\rho$ is the number of parents that get selected each time to create a single offspring, $\lambda$ is the number of offspring that needs to be created, and $\dagger$ specifies whether the parents at each iteration are available for selection at the next iteration ($\dagger$) or only the offspring are ($\cdot$).

The basic configuration of ES utilizes the following search operators. The $n$-tournament selection selects the best from $n$ individuals randomly selected, and the tournament is repeated $\rho$ times to determine the parents for the next iteration. The dominant recombination [11] creates new offspring by randomly selecting elements from the parents. The mutation operator [11] probabilistically introduces changes to the offspring depending on the representation of the solution, for example, random samples from a Gaussian distribution to vectors of real numbers, bit swaps to binary vectors, or index swaps to integer vectors. Other configurations may involve a preference between $\dagger$ selection, intermediate recombination [11] to create offspring based on the average elements from the parents, or mutation operators [11] to change the neighborhoods of the elements within the vectors.

2.2.4 Evolutionary Programming

Evolutionary Programming (EP) [53, 54] is a metaheuristic in which the individuals of the population are referred to as species, and each encodes a finite state machine initially created at random with a predefined number of states. At each iteration, the worst species in the population are selected for removal according to a selection operator, which may do so
deterministically or probabilistically. For each species in the population, an offspring is created after random mutations on the parent solution. The mutation operator can add a new state, delete an existing state, change the start state, change an output symbol, or change a state transition, all of which with equal probabilities. EP is one of the earliest EC algorithms, and it is related to the more recent ES, GAs, and GP. According to [53], the original EP has not changed much over the years, but recent additions such as probabilistic tournament selection and self-adaptive parameters blur some of the differences between EP, ES and GAs.

2.3 Swarm Intelligence

Swarm Intelligence (SI) is the category of EC whose algorithms are inspired by the collective behaviour of organisms and the interactions between its individuals. Like EAs, each individual encodes or builds a potential solution to the problem at each iteration, and some sort of communication takes place between the individuals that will influence the exploration of the solution search space. The underlying concept of SI algorithms is that its individuals are rather simple agents with limited capabilities, but as these agents interact with others, complex behaviours emerge from their local interactions, resulting in a collective behaviour that is greater than the individual sum of its parts [9, 156, 157]. Examples in Nature are found in termite societies building complex structures [156], ant colonies foraging for food [39], and bee colonies collecting and processing nectar [143], amongst others.

Typical SI algorithms are ant colony optimization [37, 39] and particle swarm optimization [42, 71], the former of which is reviewed in this section, while the latter is described in the next section in more detail as it is the main subject of this thesis.
2.3.1 Ant Colony Optimization

Ant Colony Optimization (ACO) \[37,39\] is a metaheuristic inspired by the behaviour of ants finding the shortest path between their nest and a source of food. In ACO, its population is referred to as a colony, its individuals as ants, and the communication between ants is by means of pheromones. The solution search space is modeled as a graph for ants to explore. At each iteration, ants start at a certain node and incrementally build a solution as a sequence of connected nodes in the graph. Once the ants have built their solutions, their respective objective values are computed and utilized to deposit an amount of pheromone across the links that connect the nodes of their solutions. The pheromone trails deposited are utilized by all ants at the next iteration to build their solutions, and those trails with the most pheromone will be more likely to be followed by the ants. At each node, ants compute the probabilities of selecting every possible node according to a transition rule, which considers the amount of pheromone between the nodes and (optionally) a heuristic function that provides additional information about the problem. At the end of each iteration, an evaporation factor proportionally reduces the amount of pheromones in order to prevent early convergence and encourage the exploration of new solutions.

The first ACO algorithm was the Ant System (AS) \[38\]. In AS, the pheromone trail \(\tau_{ij}\) refers to the amount of pheromone between nodes \(i\) and \(j\), and it is computed according to Equation (2.1), where \(\rho\) is the evaporation factor, \(\Gamma\) is the number of ants in the colony, and \(\Delta\tau^k_{ij}\) is the amount of pheromone that ant \(k\) deposits between nodes \(i\) and \(j\) according to the quality of its solution. The transition rule is computed as the probability of ant \(k\) to visit node \(j\) from node \(i\) according to Equation (2.2), where \(\eta_{ij}\) is a heuristic function that specifies the preference of ant \(k\) at node \(i\) to visit \(j\), \(\alpha\) and \(\beta\) are factors that weight the influence of the pheromone and the heuristic function (respectively), and \(\Omega\) is the set of nodes available from \(i\).
2.4. PARTICLE SWARM OPTIMIZATION

\[ \tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^{\Gamma} \Delta \tau_{ij}^{k} \]  \hspace{1cm} (2.1)

\[ p_{ij}^{k} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{u \in \Omega} \tau_{iu}^{\alpha} \cdot \eta_{iu}^{\beta}} \]  \hspace{1cm} (2.2)

2.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [42, 71] is a metaheuristic inspired by swarming theory [93] and social models [102] such as bird flocking and fish schools. Its population is referred to as a swarm, its individuals as particles, and the communication between the particles is established by means of a network topology. Each particle \(i\) consists of a position vector \(x_{ij}^{t}\) at iteration \(t\) that encodes a solution to the problem, a velocity vector \(v_{ij}^{t}\) to change \(x_{ij}^{t}\) in order to explore new solutions, and a memory vector \(y_{ij}^{t}\) that stores the (personal) best position found. As such, large values in the velocity vector will favour the exploration of the solution search space because the changes to the position vector will be greater, whereas small values will favour exploitation by producing smaller changes. Additionally, particle \(i\) has access to the personal best positions found by other particles within its neighborhood \(N_{i}\), and the neighborhood best position is selected and referred to as \(\hat{y}_{ij}^{t}\). Notice that, since particles only update their personal best positions with better positions, \(\hat{y}_{ij}^{t}\) represents the best position found by any particle within \(i\)'s neighborhood throughout the iterations. With this information, particles determine their velocities and positions according to Equations (2.3) and (2.4), respectively,

\[ v_{ij}^{t+1} = wv_{ij}^{t} + c_{1}r_{1j}^{t}[y_{ij}^{t} - x_{ij}^{t}] + c_{2}r_{2j}^{t}[\hat{y}_{ij}^{t} - x_{ij}^{t}] \]  \hspace{1cm} (2.3)

\[ x_{ij}^{t+1} = x_{ij}^{t} + v_{ij}^{t+1} \]  \hspace{1cm} (2.4)

where \(w\) refers to the inertia of the particle [134], \(c_{1}\) and \(c_{2}\) are positive
CHAPTER 2. LITERATURE REVIEW

foreach iteration $t$ do
  foreach particle $i$ in swarm $S$ do
    if $\tilde{f}(x^t_i) < \tilde{f}(y^t_{i-1})$
      then $y^t_i \leftarrow x^t_i$ else $y^t_i \leftarrow y^t_{i-1}$
  foreach particle $i$ in swarm $S$ do
    $\hat{y}^t_i \leftarrow y^t_{\omega} \mid \omega = \arg \min_{j \in N_i} \tilde{f}(y^t_j)$
  foreach particle $i$ in swarm $S$ do
    update $v^{t+1}_i$ and $x^{t+1}_i$ using (2.3) and (2.4)

Figure 2.1: Particle Swarm Optimization.

acceleration coefficients that determine the influence of its personal and neighborhood best positions, $r^1_{ij}$ and $r^2_{ij}$ are random values sampled from a uniform distribution $U(0,1)$, $y^t_{ij}$ is the value of dimension $j$ of its personal best position, and $\hat{y}^t_{ij}$ is the value of dimension $j$ of its neighborhood best position. These equations are utilized in the PSO algorithm for a minimization problem as shown in Figure 2.1 where $\tilde{f}(x)$ is the objective value of the solution represented by position $x$. Hereafter, we refer to the positions of the particles mostly as solutions.

2.4.1 Operation

The operation of PSO is illustrated in Figure 2.2 as follows.

Figure 2.2a shows the objective space of a two-dimensional optimization problem defined as $F_{\text{sphere}}(a,b) = a^2 + b^2$. The objective is to find the values of $a$ and $b$ such that $F_{\text{sphere}}$ results in its minimum value. The optimum values for $a$ and $b$ are located at $(0,0)$ and are represented by the darkest point in the middle of the figure.

Figure 2.2b shows the initialization of PSO. Each particle encodes in its position a potential solution to the problem as a two-dimensional vector corresponding to the values $a$ and $b$. The positions of all particles are
2.4. PARTICLE SWARM OPTIMIZATION

initialized with random values obtained from a uniform distribution, and their velocities are all initialized to zero [47].

Figures 2.2c to 2.2f simulate the positions of the particles after exploring the search space for a number of iterations. At each iteration, every particle evaluates the objective value of its current solution and compares it against the objective value of its personal best solution. If the current solution is better, then the personal best solution is updated with such a solution. Otherwise, the current solution will be discarded at the next iteration via Equation (2.4). After particles have decided whether to update their personal best solutions, particles select the best solution within their own neighborhoods. Lastly, particles update their velocities and positions according to Equations (2.3) and (2.4) such that they become partially attracted towards their personal and neighborhood best solutions. This operation is repeated at each iteration until a certain criterion is met.

2.4.2 Network Topology

The network topology of the swarm defines the neighborhoods to which particles belong, thereby establishing links between the particles from which they can select their neighborhood best solutions. The most commonly used topologies are the ring and the star [87], but others have also been proposed in the literature [1, 48, 65, 68, 75, 144].

Ring Topology

The ring topology defines the neighborhood \( N_i \) of particle \( i \) as the set particles adjacent to \( i \) according to Equation (2.5), where \( \text{mod} \) refers to the euclidean modulo operator, \( n \) is the number of neighbors, and \( S \) refers to the set of particles in the swarm. The neighborhood \( N_i \) includes particle \( i \) and its \( n \) adjacent neighbors by index from the set of particles in the swarm. Hence, adjacent neighborhoods overlap with each other by having particles in common between them. The typical case of the ring topology has
CHAPTER 2. LITERATURE REVIEW

(a) Optimization problem  (b) Random initialization  (c) $t = 5$ iterations

(d) $t = 10$ iterations  (e) $t = 15$ iterations  (f) $t = 20$ iterations

Figure 2.2: Simulated operation of Particle Swarm Optimization.
2.4. PARTICLE SWARM OPTIMIZATION

\[
N_i = \bigcup_{j=i-m}^{i+m} j \mod |S|, \quad \text{with } m = \left\lfloor \frac{n}{2} \right\rfloor
\]  

(2.5)

Star Topology

The star topology defines the neighborhood \( N_i \) of particle \( i \) as the set of all the particles in the swarm. The star topology is a special case of the ring topology when \( n = |S| \). The star topology is shown as a graph in Figure 2.3b where particles are represented as nodes according to their index in the swarm.

The Effect of the Network Topology

The network topology defines the neighborhoods from which particles will select the best solutions to be partially attracted to. Hence, the network topology will influence the quality of the neighborhood best solutions and, ultimately, the diversity of solutions in the swarm. For instance, the network topology will provide different tradeoffs between the exploration and exploitation of the search space as the ring topology with \( n = 2 \) expands to become the star topology with \( n = |S| \). Specifically, smaller neighborhood sizes will encourage exploration as more particles will be partially attracted towards different neighborhood best solutions, whereas
larger neighborhood sizes will encourage exploitation as more particles will be partially attracted towards the same neighborhood best solutions.

The best tradeoff between exploration and exploitation of the search space will depend on the difficulty of the objective space of the optimization problem. Smaller neighborhood sizes will make the swarm resilient to stagnation in local minima at the cost of a slow convergence, and hence are preferred for difficult optimization problems. Conversely, larger neighborhood sizes will make the swarm converge faster at the cost of vulnerability to stagnation in local minima, and hence are preferred for easier optimization problems. Other works in the literature have suggested that changing the neighborhood size \([1, 68, 95, 137]\) may potentially improve the quality of the solutions found.

2.5 Optimization Problems Subject to Noise

Optimization problems subject to noise are a challenging type of problem because the evaluation of the solutions will rarely (if ever) reflect their true objective values. Instead, the effect of noise will lead to underestimations or overestimations that will deteriorate the performance of metaheuristics in general. This uncertainty is usually modeled in controlled environments as sampling noise from a Gaussian distribution \([69]\), and the severity of noise is determined by its effect on the objective values and by the standard deviation of the noise distribution. For example, common types of noise are modeled as follows,

\[
\hat{f}_+ (x) = f(x) + N(0, \sigma^2) \quad (2.6)
\]
\[
\hat{f}_\times (x) = f(x) \times N(1, \sigma^2) \quad (2.7)
\]

where \(\hat{f}_+\) and \(\hat{f}_\times\) refer to additive and multiplicative noise (respectively), \(f(x)\) is the true objective value of solution \(x\), and \(N(\mu, \sigma^2)\) is a random
value sampled from a Gaussian distribution with mean $\mu$ and standard deviation $\sigma$. Hereinafter, the true objective value of solution $x$ is represented as $f(x)$, a single noisy evaluation of solution $x$ is represented as $\hat{f}(x)$, and the estimated objective value of solution $x$ is represented as $\tilde{f}(x)$. Thus, in the absence of noise, $f(x) = \hat{f}(x) = \tilde{f}(x)$.

2.5.1 Classification according to Uncertainty

The classification proposed in [69] divides optimization problems according to the type of uncertainty into four classes, namely noise, robustness, objective approximation and time-varying objective functions. The category of noise covers the problems whose objective space is subject to noise. The category of robustness covers the problems whose solution space is subject to noise. The category of objective approximation avoids evaluating costly objective functions and instead creates and updates a model which estimates the objective values of the solutions from previously collected data. The category of time-varying objective functions covers the dynamic optimization problems whose objective space changes in time and yet remains deterministic at any given instant.

In the category of noisy optimization problems, the noise mitigation mechanisms are classified into explicit averaging when resampling methods are involved, implicit averaging when methods other than resampling are involved (e.g. increasing the population size), and as modifying selection when the individuals are discarded from being selected unless they satisfy certain requirements regarding the effect of noise [69].

2.5.2 Direction of the Optimization Problem

The direction of the optimization problem [111, 112] refers to the challenge for algorithms when the objective space is subject to levels of noise that vary depending on the location of the solutions. The direction of an optimization problem is backwards when the global optimum can be found by
directing the search towards the low-noise regions. Conversely, the direction is *forwards* when the global optimum can only be located by traversing the regions subject to high levels of noise. According to these definitions [111, 112], multiplicative noise on the objective values will define a direction for the optimization problem given that the severity of noise will change according to the objective values of the solutions. Thus, the additional challenge of multiplicative noise over additive noise is a larger corruption of the objective values whose magnitude changes across the search space proportionally to the objective values of the solutions.

In optimization problems subject to multiplicative noise, the direction of the problem will depend on the objective of the optimization and on the range of the objective space. Specifically, on minimization problems whose objective space is only positive, the objective values of better solutions will be affected by a smaller severity of noise. Conversely, in maximization problems, the objective values of better solutions will be affected by a larger severity of noise. Hence, the direction of these two instances of optimization problems are backwards and forwards, respectively.

### 2.5.3 Simulation-Based Optimization

In the discipline of Operations Research, optimization problems subject to noise are comprised within the general field of *simulation-based optimization* [2, 3, 4, 36, 56, 57, 58, 103], which is defined as the process of finding the best parameter values for a system whose performance is evaluated from a stochastic simulation model [139]. The formulation of a general stochastic simulation model is commonly found as Equation (2.8), where \( x \) is a solution in the search space \( \Theta \), \( \hat{f}(x) \) is the objective function to optimize, \( \omega \) represents a simulation replication, \( L \) is the sample performance measure, and \( E \) is the expected objective value.

\[
\min_{x \in \Theta} \hat{f}(x) = E [L(x, \omega)] \tag{2.8}
\]
Some examples of real-world problems that are modeled as stochastic simulation models are the design and operations of call centers [57], inventory policies for orders in inventory control systems [57], scheduling of manufacturing cells for the production of aircraft and gas turbine engines [139], and scheduling of camshaft machining lines [139]. In all these cases, the optimization is performed on stochastic simulation models whose samples could be obtained, for example, over a period of time on an online system or from already known probability distributions.

**Design and Operations of a Call Centre**

A call centre provides support to customers via telephone. The design of a call centre depends on many variables that will define its operation towards the handling of the calls. The variables can be obtained from Customer Relationship Management (CRM) systems that store information about customers such as inquiries, requests, complaints, and priorities [57]. Based on these variables, different policies determine the assignment of available operators to the calls depending on other variables such as skills of the operator, routing algorithms and types of queues [57]. The optimization of the design and operations of a call centre consists of finding potential solutions with optimal settings for the variables involved. The objective values of these solutions may consist of metrics relative to the customers or operators such as reducing waiting times, operational costs, network usage, abandonment rates of calls, and operators usage [57].

**Inventory Control System**

An inventory control system manages the availability of items in stock by automatically placing new orders when the stock falls below a certain level. A simple example considers the optimization of two parameters, namely the re-order level and the order-up-to level [57]. The re-order level
refers to the minimum availability of an item in stock before a new order is placed to increase it. The order-up-to level refers to the amount of items that the new order will place to stock up with. Solutions to this problem will find the optimal values to both levels such that a cost objective function is minimized. Examples of the objective functions involve reducing ordering and holding costs, and lost sales components, amongst others [57].

Scheduling of Manufacturing Cells

The scheduling of a real-world manufacturing cell of components for aircraft and gas turbine engines was addressed in [139]. The cell consists of five machines and five burring stations that operate differently according to the components that arrive at the cell. If multiple components arrive simultaneously, a priority function \( p \) determines their precedence based on their due times and that results in a critical ratio value \( p < 1.0 \) if the component is behind schedule, \( p = 1.0 \) if it is on schedule, and \( p > 1.0 \) if it is ahead of schedule, where those components with smaller ratio values take precedence. The due time for each type of component is determined by an inter-arrival time that specifies the frequency at which the type of component enters the system. The optimization problem is to find the optimal values of inter-arrival times for the different types of components such that the utilization of the cell is maximized and the tardiness is minimized.

Scheduling of Camshaft Machining Lines

The scheduling of a real-world camshaft machining line was addressed in [139]. The machining line produces 15 different variants of camshafts, and the tasks to be performed on each camshaft are allocated between 14 specialized groups containing 34 machines in total that operate in parallel. Each machine has a processing time, physical capability and limitations,
2.6. PSO FOR NOISY OPTIMIZATION PROBLEMS

and a variable number of failures and set-ups. The production takes place with a number of batches containing 50 camshafts of the same type, each of which is moved to a storage area once it is finished. The optimization problem is to find an optimal setting to match batch types and machines such that product shortage is minimized while the throughput of the line is maximized.

2.6 PSO for Noisy Optimization Problems

One characteristic that has remained largely unexplored in PSO is the deterioration of the quality of its results on optimization problems subject to noise. In this type of problem, the objective values of the solutions are corrupted by the effect of sampling noise, thus causing solutions to have inaccurate objective values that change every time these are evaluated. As a consequence, particles eventually fail to distinguish good from bad solutions, leading in turn to other issues that ultimately end up deteriorating the quality of its results. Two such issues recognized in the literature are the inaccurate memories of the particles and the incorrect selection of their neighborhood best solutions, both of which have been addressed respectively by incorporating evaporation mechanisms [30, 31, 32, 50, 51] and resampling methods [7, 106, 113, 154] into PSO.

The most relevant works on PSO for noisy optimization problems are presented next, and we group them according to the noise mitigation mechanisms as follows: a) hypothetical perfect noise mitigation that achieve either local or global certainty, b) noise mitigation mechanisms where the objective values of the solutions are estimated based on a single evaluation, and c) noise mitigation mechanisms where the objective values of the solutions are estimated based on resampling methods.
2.6.1 PSO under Local and Global Certainty

Local and global certainty refers to two assumptions under which PSO algorithms are evaluated on optimization problems subject to noise. On the one hand, particles in PSO with Local Certainty (PSO-LC) operate utilizing the true objective values of their current and personal best solutions, hence they always make the right decisions when it comes to discarding or updating them. On the other hand, particles in PSO with Global Certainty (PSO-GC) operate utilizing the true objective values of the personal best solutions from the particles within their neighborhoods, hence they always select their true neighborhood best solutions. Under these assumptions, the evaluation of PSO provides insights about the importance of certainty at both levels and the consequences of the lack thereof.

Thus far, the performance of PSO on optimization problems subject to noise under the assumptions of local and global certainty has only been explored in [7], where PSO-LC is described as offering a superior performance due to its continuous progress throughout the optimization process and its resilience to noise and stagnation up to a certain extent. Conversely, the observations in [7] consider PSO-GC to suffer from stagnation without converging to an optimum due to particles being misled by the effect of noise. Unfortunately, the study in [7] did not include any quantitative results to support their analyses on this matter. Nonetheless, their findings are in line with our expectations given that the neighborhood best solutions are selected from the personal best solutions, and therefore the quality of the neighborhood best solutions will depend on the quality of the personal best solutions, and the quality of the results will ultimately depend on the quality of both personal and neighborhood best solutions.

2.6.2 PSO with Single-Evaluation Methods

We refer as single-evaluation methods in PSO to those noise mitigation mechanisms whose operation does not involve performing additional eval-
2.6. PSO FOR NOISY OPTIMIZATION PROBLEMS

uations to any of the solutions in the swarm, thereby settling with having their objective values estimated upon a single evaluation that is performed by the PSO algorithm. Consequently, single-evaluation methods need to assume that the objective values of the solutions are very inaccurate and cannot be better estimated like resampling methods do with multiple evaluations and a sample mean over them. Despite that in resampling-based PSO algorithms the objective values of the solutions are better estimated, the leverage of single-evaluation PSO algorithms is that they perform more iterations and hence have more opportunities to find better solutions.

**PSO with Evaporation**

PSO with Evaporation (PSO-E) \[31\] incorporates an evaporation mechanism into PSO as a single-evaluation method to have particles worsen the estimated objective values of their personal best solutions in order to encourage their replacement with new solutions. The evaporation takes place in a particle whenever its current solution is not better than its personal best solution, a case that hereinafter we refer to as an unsuccessful iteration of the particle. The PSO-E algorithm for a minimization problem is described in Figure 2.4, where particle \(i\) estimates the objective value of its personal best solution based on the number of unsuccessful iterations \(u_i\) and the evaporation factor \(\rho \in [0.0, 1.0]\) according to Equation 2.9. Thus, for example, a particle with \(\rho = 1.0\) will worsen the estimated objective value of its personal best solution by doubling it after a single unsuccessful iteration.

\[
\hat{f}(\mathbf{y}_i^t) = \hat{f}(\mathbf{y}_i^{t-u_i})(1 + \rho)^{u_i}
\]  

(2.9)

The goal of the evaporation mechanism in minimization problems subject to noise is to correct the potentially underestimated objective values of the personal best solutions such that they do not affect the frequency at
foreach iteration $t$ do

foreach particle $i$ in swarm $S$ do

if $f(x^t_i) < f(y^t_{t-1})$

then $y^t_i \leftarrow x^t_i$, $u_i \leftarrow 0$

else $y^t_i \leftarrow y^t_{t-1}$, $u_i \leftarrow u_i + 1$

foreach particle $i$ in swarm $S$ do

$\hat{y}^t_i \leftarrow y^t_\omega \mid \omega = \arg \min_{j \in N_i} \tilde{f}(y^t_j)$

foreach particle $i$ in swarm $S$ do

update $v^{t+1}_i$ and $x^{t+1}_i$ using (2.3) and (2.4)

Figure 2.4: PSO with Evaporation (PSO-E).

which particles update them [31]. For example, Figure 2.5 depicts the underestimated objective value of the personal best solution $y$, and its worsening towards infinity in proportion to the evaporation factor $\rho$ after each unsuccessful iteration. The goal of the evaporation mechanism is the same for dynamic minimization problems, where worsening the estimated objective values will encourage the replacement of the personal best solutions with new solutions that have their objective values more recently estimated, hence reflecting any possible changes of the objective space that will partially attract the particles towards current (and potentially better) regions of the search space.

The evaporation mechanism proposed in [31] is utilized on a variant of PSO where the particles store a copy of the neighborhood best solutions, which are replaced whenever better solutions are found within the neighborhoods. Therefore, the original evaporation mechanism was proposed with two evaporation factors, one for the personal best solutions and another for the copies of the neighborhood best solutions. However, for the sake of simplicity, in our study we consider only the incorporation of the evaporation mechanism into the regular PSO algorithm, for which we only require one evaporation factor $\rho$ to worsen the estimated objective values.
2.6. PSO FOR NOISY OPTIMIZATION PROBLEMS

2.6.3 PSO with Resampling Methods

Resampling methods in PSO are noise mitigation mechanisms that estimate the true objective values of the solutions by performing multiple evaluations to the solutions and taking a sample mean over the evaluations. As such, the more re-evaluations performed to a solution, the more accurate its objective value will be because the standard error will be reduced proportional to $1/\sqrt{n}$ after $n$ evaluations. However, since the additional evaluations performed are extracted from a fixed and limited computational budget $B$ available to the algorithm, incorporating a resampling method into PSO creates a tradeoff between the number of re-evaluations and the number of iterations. Consequently, resampling-based PSO algo-
algorithms need to decide upon a balance between the accuracy of the estimated objective values of the existing solutions (i.e. more re-evaluations) and the exploration of new solutions (i.e. more iterations). Once this trade-off has been decided upon, the problem is then to allocate the computational budget $B^t$ amongst the solutions at iteration $t$.

Resampling-based PSO algorithms divide the computational budget $B^t$ in budgets $B^t_\alpha$ and $B^t_\beta$, one allocated after the other. The first budget, $B^t_\alpha$, is equally divided and allocated between the current solutions in the swarm, and the second budget, $B^t_\beta$, is allocated according to the criteria of the algorithm. The following sections detail the most relevant resampling-based PSO algorithms available in the literature for a minimization problem (without any loss of generality) assuming that the computational budget $B^t$ is the same at every iteration.

**PSO with Equal Resampling**

PSO with Equal Resampling (PSO-ER) allocates both computational budgets $B^t_\alpha$ and $B^t_\beta$ equally between the current solutions in the swarm, and then PSO proceeds as usual utilizing a sample mean over the evaluations allocated to the current solutions as their respective objective values. PSO-ER is the most basic resampling-based PSO algorithm that different works in the literature have used as a baseline reference to compare against other algorithms [7, 106, 113, 123]. While the quality of its results are generally worse than that of any other resampling-based algorithm, the quality of its results are still significantly better than the regular PSO [7, 106, 154]. Besides simplicity, the advantage of PSO-ER is that the objective values of the current solutions are estimated more accurately. However, in doing so, PSO-ER sacrifices the potential accuracy of better solutions to improve that of the worse solutions. Nonetheless, PSO-ER is still a useful baseline to compare against in order to ensure that new resampling-based PSO algorithms find better solutions.

The PSO-ER algorithm is presented in Figure 2.6, where $b_\alpha$ and $b_\beta$ are
foreach iteration \( t \) do

foreach particle \( i \) in swarm \( S \) do

\( \mathcal{X}_i^t \leftarrow \bigcup_{b_0}^{b_0 + b_0} \hat{f}(x_i^t) \)

if \( \text{mean}(\mathcal{X}_i^t) < \text{mean}(\mathcal{Y}_i^{t-1}) \) then 

\( y_i^t \leftarrow x_i^t \), \( \mathcal{Y}_i^t \leftarrow \mathcal{X}_i^t \)

else \( y_i^t \leftarrow y_i^{t-1} \), \( \mathcal{Y}_i^t \leftarrow \mathcal{Y}_i^{t-1} \)

foreach particle \( i \) in swarm \( S \) do

\( \hat{y}_i^t \leftarrow y_{i, \omega}^t \mid \omega = \arg \min_{j \in \mathcal{N}_i} \text{mean}(\mathcal{Y}_j^t) \)

foreach particle \( i \) in swarm \( S \) do

update \( v_i^{t+1} \) and \( x_i^{t+1} \) using (2.3) and (2.4)

Figure 2.6: PSO with Equal Resampling (PSO-ER).

the individual computational budgets allocated to each particle from \( B_\alpha^t \) and \( B_\beta^t \) (respectively), \( \hat{f}(x_i^t) \) is a noisy evaluation of solution \( x_i^t \), \( \mathcal{X}_i^t \) is the set of evaluations performed on solution \( x_i^t \), and \( \mathcal{Y}_i^t \) is the set of evaluations performed on the personal best solution \( y_i^t \).

PSO with Optimal Computing Budget Allocation

PSO with Optimal Computing Budget Allocation (PSO-OCBA) was proposed in [106] to asymptotically maximize the estimated probability for particles to correctly select their neighborhood best solutions within a star topology. PSO-OCBA equally divides and allocates \( B_\alpha^t \) between the current solutions to estimate their respective means and variances, both of which are utilized to allocate the additional computational budget \( B_\beta^t \) according to the Optimal Computing Budget Allocation (OCBA) [25].

OCBA divides \( B_\beta^t \) into groups of \( b_\Delta \) evaluations to be allocated sequentially between the estimated best solutions in the swarm, that is, between the current and personal best solutions that are likely to be the true best solutions. After each allocation, the means and variances are re-estimated and utilized again to allocate the next group of \( b_\Delta \) evaluations according
to the $z$-scores of the solutions with respect to the estimated neighborhood best solution. Specifically, $z$-scores favour the allocation of evaluations to those solutions whose estimated objective values have the lower means and higher variances, hence improving the accuracy of the objective values of the solutions that are more likely to be the best solutions in the swarm. The allocation of $B^t_\beta$ is determined according to Equations (2.10) and (2.11) in order to maximize the probability of correct neighborhood best selection [25],

$$\frac{(\bar{e}_q - \bar{e}_a)}{s_q/\sqrt{b_q}} = \frac{(\bar{e}_p - \bar{e}_a)}{s_p/\sqrt{b_p}}, \quad p, q \in \{1, 2, \ldots, k\} \text{ and } p \neq q \neq a$$

(2.10)

$$b_a = s_a \sqrt{\sum_{p=1, p \neq a}^{k} \frac{b_p^2}{s_p^2}}$$

(2.11)

where $b_p$ is the number of evaluations allocated to solution $p$, $\bar{e}_p$ and $s_p$ are the mean and standard deviation of the set of evaluations performed to solution $p$, $a$ refers to the estimated best solution (i.e. the one with the lowest mean), and $k = 2 |S|$ is the number of both current and personal best solutions in the swarm. The PSO-OCBA algorithm is presented in Figure 2.7, where $b_\alpha$ is the individual computational budgets allocated to each particle from $B^t_\alpha$, $\hat{f}(x^t_i)$ is a noisy evaluation of solution $x^t_i$, $X^t_i$ is the set of evaluations performed to solution $x^t_i$, and $Y^t_i$ is the set of evaluations performed to the personal best solution $y^t_i$. The reader is encouraged to refer to [24, 25] for further details on OCBA.

### 2.7 Related Work

The literature addressing optimization problems subject to noise with PSO is not large, unlike the cases of other metaheuristics such as GAs and ES. In the following sections, we present most of the work on PSO in opti-
2.7. RELATED WORK

mization problems subject to noise, and also the most relevant works with other metaheuristics.

2.7.1 PSO and Noise

Parsopoulos and Vrahatis [108, 109] performed the earliest studies about the effect of noise on PSO. They evaluated the performance of PSO in terms of the success rate in finding the optimal solution to a three-dimensional benchmark function and nine bi-dimensional benchmark functions all subject to multiplicative Gaussian noise. In their first work [108], they utilized different levels of noise with standard deviations $\sigma \in [0.1, 0.9]$ and achieved success rates of over 88% as long as $\sigma \leq 0.3$, and worse otherwise. In their second work [109], they considered $\sigma \in [0.1, 0.3]$ and achieved success rates of 100% in the majority of the problems. Based on these results, they concluded that PSO is stable and efficient in the presence of noise, and that noise may even be beneficial for PSO. However, we question

```plaintext
foreach iteration $t$ do
    foreach particle $i$ in swarm $S$ do
        $X^t_i \leftarrow \bigcup_i^b \hat{f}(x^t_i)$
    ocba($S$, $B^t_B$, $b_\Delta$)
    foreach particle $i$ in swarm $S$ do
        if $\text{mean}(X^t_i) < \text{mean}(Y^{t-1}_i)$
            then $y^t_i \leftarrow x^t_i$, $Y^t_i \leftarrow X^t_i$
        else $y^t_i \leftarrow y^{t-1}_i$, $Y^t_i \leftarrow Y^{t-1}_i$
    foreach particle $i$ in swarm $S$ do
        $\hat{y}^t_i \leftarrow y^t_\omega \mid \omega = \arg\min_{j \in N_i} \text{mean}(Y^t_j)$
    foreach particle $i$ in swarm $S$ do
        update $v^{t+1}_i$ and $x^{t+1}_i$ using (2.3) and (2.4)
```

Figure 2.7: PSO with Optimal Computing Budget Allocation (PSO-OCBA).
CHAPTER 2. LITERATURE REVIEW

such high success rates because the optimization problems were subject to very high levels of multiplicative noise, especially the levels in [108] where a value of $\sigma = 0.9$ could corrupt the objective values by factors of up to $1.0 \pm 3\sigma$.

A variety of other works have utilized PSO to address different noisy optimization problems [19, 83, 107, 133, 147, 148, 155], generally favouring the PSO algorithms over the other techniques compared against. However, none of the algorithms incorporates noise mitigation mechanisms or addressess the effect of noise in any other way. Some of these works acknowledge that the quality of the results of PSO deteriorates due to the presence of noise [147, 148], while others suggest that their PSO algorithms are tolerant to noise up to a certain extent [19, 83, 107, 133, 155].

2.7.2 PSO with Single-Evaluation Methods

Cui et al. [31] proposed the evaporation mechanism for PSO to improve its performance on optimization problems subject to noise. PSO with Evaporation (PSO-E) consists of worsening the estimated objective values of the personal and neighborhood best solutions whenever particles are unable to find better replacements. The estimated objective values are worsened by an evaporation factor $\rho$ on their personal best solutions and another evaporation factor $\hat{\rho}$ on the neighborhood best solutions. The PSO variant presented by the authors has each particle store in memory a copy of the neighborhood best solutions, for which the evaporation factor $\hat{\rho}$ is required. The experiments were designed to determine the utility of the evaporation mechanism and the effect of different factors on the quality of the solutions found. The experiments were performed on the three-dimensional sphere function whose solution space is subject to additive Gaussian noise with $\sigma \in \{0.01, 0.025, 0.05\}$, hence classifying the problem as a search for robust solutions according to the classification in [69]. In the absence of noise, the quality of the results obtained with PSO-E us-
2.7. RELATED WORK

ing $\rho = \hat{\rho} = 0.36$ was not significantly different from that obtained with the regular PSO. However, in the presence of noise, the quality of the results obtained with PSO-E was significantly better. In addition, based on further experiments, they suggest that the best evaporation factors are $\rho \in [0.44, 0.54]$ and $\hat{\rho} \in [0.27, 0.36]$.

Cui and Potok [32] experimented with PSO-E on dynamic optimization problems whose objective functions are additionally subject to multiplicative Gaussian noise. They compared the regular PSO, PSO-E with $\rho = 0.42$ and $\hat{\rho} = 0.86$, and two other algorithms whose particles periodically reset their memories [23, 44]. Experiments on the two-dimensional benchmark function DF1 [96] showed that the quality of the results obtained with PSO-E is better than that of the other algorithms, for which they concluded that the evaporation mechanism allows particles to keep track of the solutions. These conclusions were further supported with a similar set of experiments in [30].

Fernandez-Marquez and Arcos [50] incorporated the evaporation mechanism into the Multi-Quantum Swarm Optimization (MQSO) algorithm, which divides the swarm into multiple subswarms that repel each other in order to explore different regions of the search space. In addition, they proposed another evaporation mechanism in which the solutions are worsened by adding the evaporation factor instead of multiplying it. They explored the performance of MQSO utilizing different evaporation factors, both additive and multiplicative, on the five-dimensional Moving Peak Benchmark (MPB) [15] whose objective space changes periodically after a certain number of iterations and is additionally subject to different levels of uniform additive noise $U(0, \gamma)$ as $\gamma \in \{0, \ldots, 30\}$. Experiments showed that better results are obtained with evaporation factors $\rho = 0.5$ and $\rho = 0.011$ in additive and multiplicative evaporation mechanisms (respectively), but the quality of the results between them was found not to be significantly different. Later on, they further improved upon these results by utilizing a dynamic evaporation factor whose value changes according
to the velocity of the particles [51].

Rada-Vilela et al. [126] studied the effect of different levels of noise and evaporation factors on the quality of the results of two PSO algorithms. The algorithms under evaluation were the regular PSO and a variant named Random Asynchronous PSO (RA-PSO), in which particles are selected at random and perform asynchronous updates [124]. Experiments on 20 large-scale optimization problems subject to different levels of multiplicative Gaussian noise showed a superior quality of results for RA-PSO, which also exhibited a positive correlation between the evaporation factor and the level of noise. Differently, the evaporation mechanism in regular PSO often produced worse results and no correlation was found at all. Later on, they improved upon these results with a heterogeneous swarm based on RA-PSO in which particles have different evaporation factors [125].

### 2.7.3 PSO with Resampling Methods

Liu et al. [82] addressed the permutation flow shop scheduling problem with a PSO-ER variant that incorporates Simulated Annealing (SA) [72] and hypothesis testing. At each iteration, their algorithm utilizes SA to perform local search around the best solution in the swarm to improve it. Hypothesis testing is utilized in particles to decide whether to replace their personal best solutions with the current solutions based on a statistical test between their respective sets of evaluations. In addition, they create two other PSO algorithms that utilize SA, but do not perform resampling or hypothesis testing. Experiments are performed on eight benchmark functions of up to 20 dimensions and their objective spaces are subject to two different levels of uniform additive noise computed according to the problem. It is not certain whether the comparison is based on the quality of the results obtained after a fixed computational time or number of function evaluations performed. The proposed algorithms are compared based
on the quality of the results obtained on the different benchmark functions, where the resampling-based algorithm outperformed the other two. Different settings of their resampling-based algorithm show that more re-evaluations to the solutions and lower levels of noise improve the quality of the results.

Pan et al. [106] integrated the OCBA resampling method [25] into PSO to maximize the probability of particles to correctly select their true neighborhood best solutions. PSO-OCBA allocates a computational budget at every iteration amongst the estimated best solutions in a sequential manner, hence improving the accuracy of the most relevant solutions that could be the true neighborhood best solutions. They also developed another algorithm based on PSO-OCBA to encourage the diversity of the swarm by utilizing hypothesis testing between the solutions starting from the best solution to the worst solution. If the statistical difference between any two solutions is not significant, the worse solution is discarded and replaced with a new randomly created solution. They compared PSO-OCBA, PSO-OCBA with hypothesis testing and PSO-ER on six benchmark functions with 2 to 6 dimensions subject to additive Gaussian noise. The results showed that PSO-OCBA with hypothesis testing finds better solutions than PSO-OCBA, and both find better solutions than PSO-ER.

Bartz-Beielstein et al. [7] compared the quality of the results obtained with PSO-ER and PSO-OCBA on the 10-dimensional sphere function subject to additive and multiplicative Gaussian noise. They found that PSO-OCBA finds better solutions than PSO-ER because its particles have a higher probability to correctly select the neighborhood best solutions as well as to correctly update their personal best solutions. Both algorithms were favourably compared against the regular PSO, even after tuning some of their parameters via Sequential Parameter Optimization [8]. Besides these experiments, they also evaluated PSO-ER under the assumptions of local and global certainty, for which they mentioned that better results are obtained under local certainty, but the authors did not include the results.
supporting such observations.

Zhang et al. [154] developed two new algorithms and compared them against PSO-OCBA and PSO-ER. The new algorithms divide the swarm into two halves according to the quality of the current solutions. Particles on the better half utilize a social-only model [70] which updates the velocities of the particles ignoring the influence of the personal best solutions. Particles on the worse half utilize as personal best solutions the nearest solutions (in search space) from the better half. One of their algorithms utilizes equal resampling to re-evaluate the current solutions, whereas the other utilizes a specifically designed setting of OCBA that takes into consideration the number of times that particles in the better half are referenced by those in the worse half. Experiments on two two-dimensional benchmark functions subject to additive Gaussian noise showed that the quality of the results obtained with PSO-OCBA and their OCBA-based algorithm do not differ significantly, but the quality of the results from both algorithms is significantly better than that from the other two on both problems.

Piperagkas et al. [113] considered the incorrect selection of the neighborhood best solutions to be the main challenge to address in PSO when optimization problems are subject to noise. They developed two PSO algorithms which incorporate reinforcement learning to allocate more evaluations to those solutions likely to maximize the probability of correctly selecting the neighborhood best solutions. The allocation is performed sequentially utilizing roulette wheel selection in one algorithm and stochastic independent decisions in the other, and both selection methods utilize a set of weights that is controlled by reinforcement learning. Experiments were performed on five benchmark functions with 5, 15 and 40 dimensions, and the objective space is subject to low levels of multiplicative Gaussian noise with \( \sigma \in \{0.01, 0.03, 0.05\} \). The results showed that their new algorithms find solutions which are not significantly different from each other, but the solutions from the two algorithms are significantly bet-
ter than those obtained with PSO-ER.

Rada-Vilela et al. [122] developed a variant of PSO-OCBA in which particles update their personal best solutions to their current ones only if the difference between them is statistically significant according to a $t$-test. They compared the new variant against PSO-OCBA on 20 large-scale benchmark functions [141] subject to different levels of multiplicative Gaussian noise. Experiments were designed for both algorithms to perform resampling at different iteration frequencies ranging from every single iteration to every 10 iterations, thereby balancing between resampling more often at the cost of fewer iterations and less often but more iterations. The results showed that PSO-OCBA finds better solutions than their new variant when resampling at every iteration. Failing to resample at one or more iterations deteriorates the quality of the results from both algorithms, but their variant finds better solutions than PSO-OCBA in these cases and this is attributed to the protection provided by the $t$-test.

### 2.7.4 EAs and Noise

This section presents some of the most relevant works on EA for optimization problems subject to noise. We only focus on GAs and ES in order to provide insights about the different challenges that the effect of noise has in such metaheuristics. A more detailed and comprehensive review can be found in [12].

#### Genetic Algorithms

In GAs, the effect of noise will not hinder its performance when utilizing the roulette-wheel selection given that individuals are selected probabilistically according to their objective values, hence rendering such a scheme noise invariant [91]. However, when utilizing $n$-tournament selection, the effect of noise will definitely hinder the correct selection of the best solution from $n$ individuals randomly selected. The $n$-tournament selection
may be preferred over the roulette-wheel selection when a specific selection pressure is desired. The selection pressure in GAs refers to the probability of better individuals to be selected. The higher the selection pressure, the more it favours the selection of better individuals, with elitism being the highest selection pressure possible given that the best individual of the population is selected. In GAs, the selection pressure affects the tradeoff between the exploration and exploitation of the search space. Specifically, a higher selection pressure leads to more exploitation and earlier convergence, whereas a lower selection pressure leads to more exploration and later convergence.

The following are perhaps the most notable works on GAs handling noise. Fitzpatrick and Grefenstette [52] evaluated the performance of GAs on a 30-dimensional function subject to additive Gaussian noise. They utilized a computational budget of 2,000 function evaluations at each iteration, and evaluated the performance of different tradeoffs between population size and number of samples to estimate the objective values. The algorithms perform 200 iterations and the results show that the best performing tradeoff is to utilize a population size of 2,000 whose solutions are estimated upon a single evaluation. They also address a problem of image restoration experimenting with the different tradeoffs, favouring smaller populations with better estimated objective values. However, in the image restoration problem, larger population sizes perform significantly fewer iterations due to the overhead caused to the GA.

Miller and Goldberg [90, 91] derived a model to predict the selection pressure of tournament selection based on the variance of the objective values within the population. They start from the assumption that mean objective values of the population are normally distributed, and from there they estimate the objective values of the individuals. Under such a model, the effect of noise on the objective values contributes to the variance of the population and the model remains valid considering normally distributed additive noise. However, the model requires previous knowledge about
2.7. RELATED WORK

the problem such as the level of noise and differences to be detected in the objective values. They validated their model on three benchmark functions with 25 dimensions, each represented within the chromosome as a binary block of 4 dimensions resulting in vectors of 100 elements.

Evolution Strategies

In ES, the effect of noise affects the selection operator just as in GAs, but it may have further repercussions if parameters relevant to noise are also subject to reproduction and mutation. The following are perhaps the most relevant works on ES for optimization problems subject to noise. Di Pietro [111] coined the direction of the optimization problem as backwards when the the global optimum could be found by directing the search towards low-noise regions, and forwards when it is necessary to traverse the high-noise regions. Likewise, he distinguishes backward and forward setups depending on whether higher resampling rates are utilized for low-noise or high-noise regions, respectively. He proposed five different resampling-based ES algorithms and evaluated their performance on nine optimization problems with both backward and forward directions. One of the best performing algorithms required the level of noise to be divided into $m = 2$ levels and map each level to different resampling rates. The algorithm then reevaluates the solutions according to their respective levels of noise. While the algorithm requires to set $2m + 1$ parameters, the results obtained were significantly better than using equal resampling.

Markon et al. [84] presented a work, where the offspring is required to be better than their parents by a predefined threshold in order to ensure that they have indeed a truly better solution. Beyer [10] provided further guidelines based on resampling, population sizing, and the inheritance of mutations in order to improve the performance of ES on problems subject to noise.
Table 2.1: Separability of the benchmark functions.

<table>
<thead>
<tr>
<th>Set</th>
<th>Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>[F_{01-03}]</td>
<td>separable, each dimension can be independently optimized from the others</td>
</tr>
<tr>
<td>B</td>
<td>[F_{04-08}]</td>
<td>partially separable, only a single group of (m) dimensions are non-separable</td>
</tr>
<tr>
<td>C</td>
<td>[F_{09-13}]</td>
<td>partially separable, (\frac{d}{2m}) groups of (m) dimensions are non-separable</td>
</tr>
<tr>
<td>D</td>
<td>[F_{14-18}]</td>
<td>partially separable, (\frac{d}{m}) groups of (m) dimensions are non-separable</td>
</tr>
<tr>
<td>E</td>
<td>[F_{19-20}]</td>
<td>fully non-separable, any two dimensions cannot be optimized independently</td>
</tr>
</tbody>
</table>

### 2.8 Benchmark Functions

The benchmark functions that we utilize throughout this thesis to evaluate the algorithms belong to the suite presented at the CEC’2010 Special Session and Competition on Large-Scale Global Optimization [141], which was specifically designed to evaluate the performance of optimization algorithms on a variety of challenging problems. This suite comprises 20 large-scale minimization functions whose objective values are always positive and the global minimum is \(f(x) = 0\). The benchmark functions are classified into five sets according to their degree of separability as shown in Table 2.1, where parameter \(d = 1000\) refers to the total number of dimensions of the function and \(m = 50\) to the number of dimensions which are not separable. The values for \(d\) and \(m\) are suggested in [141].
2.8. **BENCHMARK FUNCTIONS**

2.8.1 **Basic Functions**

The composition of the benchmark functions consists primarily of the classical optimization functions shown in Table 2.2, where the objective spaces are presented to illustrate the intrinsic challenges of the problems to optimization algorithms. In particular, `ackley` and `rastrigin` are the most challenging benchmarks because the former has little gradient information to find the global minimum and the latter has numerous local minima surrounding the global minimum. The objective spaces in Table 2.2 are computed within the following boundaries on each dimension [141]: sphere ($\pm 5.12$), elliptic ($\pm 100$), rastrigin ($\pm 5$), ackley ($\pm 32$), schwefel ($\pm 100$), and rosenbrock ($\pm 100$). The global minimum of each function is located at the centre of the figure, whose darker colours indicate lower and (hence) better objective values. The sets $A$ and $E$ are shown together just for presentation purposes.

The equations of the basic benchmark functions are presented in Table 2.3, where the `sphere`, `elliptic`, `rastrigin`, and `ackley` are fully separable, whereas `schweffel` and `rosenbrock` are not separable. For example, expanding the `sphere` function in two dimensions yields $F_{\text{sphere}} = x_1^2 + x_2^2$, showing that terms $x_1$ and $x_2$ can be optimized independently. Conversely, expanding `schwefel` yields $F_{\text{schwefel}} = x_1^2 + (x_1 + x_2)^2$, showing the dependency of the two variables for computing the second term $(x_1 + x_2)^2$.

2.8.2 **Vector Operations**

The benchmark functions combine the basic functions in Table 2.2 with different operators to modify the solution vectors, and especially to shift the centric location of the global minimum in order to avoid introducing biases that could favour the performance of the algorithms. The benchmark functions and their details were originally presented in [141], but here we introduce a different notation aiming to provide a less verbose
Table 2.2: Composition of the benchmark functions.

<table>
<thead>
<tr>
<th>A/E</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>elliptic $F_{01}$ $F_{04}$ $F_{09}$ $F_{14}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>rastrigin $F_{02}$ $F_{05}$ $F_{10}$ $F_{15}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ackley $F_{03}$ $F_{06}$ $F_{11}$ $F_{16}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>schwefel $F_{19}$ $F_{07}$ $F_{12}$ $F_{17}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>rosenbrock $F_{20}$ $F_{08}$ $F_{13}$ $F_{18}$</td>
<td></td>
</tr>
</tbody>
</table>
Table 2.3: Equations of basic functions.

\[ F_{\text{sphere}}(x) = \sum_{i=1}^{d} x_i^2 \quad (2.12) \]
\[ F_{\text{elliptic}}(x) = \sum_{i=1}^{d} (10^6)^{\left[\frac{i-1}{d-1}\right]} x_i^2 \quad (2.13) \]
\[ F_{\text{rastrigin}}(x) = \sum_{i=1}^{d} \left[ x_i^2 - 10 \cos(2\pi x_i) + 10 \right] \quad (2.14) \]
\[ F_{\text{ackley}}(x) = -20 \exp\left(-0.2\sqrt{a}\right) - \exp(b) + 20 + e \quad (2.15) \]

where \( a = \frac{1}{d} \sum_{i=1}^{d} x_i^2 \) and \( b = \frac{1}{d} \sum_{i=1}^{d} \cos(2\pi x_i) \)

\[ F_{\text{schwefel}}(x) = \sum_{i=1}^{d} \left( \sum_{j=1}^{i} x_i \right)^2 \quad (2.16) \]
\[ F_{\text{rosenbrock}}(x) = \sum_{i=1}^{d-1} 100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \quad (2.17) \]
mathematical representation. As such, we define the operators as follows.

**Group** ($G$) creates a vector $x^g$ based on the contiguous elements from an input vector given the inclusive starting position (lower index) and the exclusive ending position (upper index). The group operation $G_0^m x$ produces a vector $x^g = \{x_0, x_1, \ldots, x_{m-1}\}$.

**Permute** ($P$) creates a vector $x^p$ by randomly shuffling the positions of the input vector. The permute operation $P x$, where $x = \{x_0, x_1, x_2\}$, could produce $x^p = \{x_2, x_0, x_1\}$ depending on the random seed. The random seed is defined in the implementation details in [141].

**Rotate** ($R$) creates a vector $x^r$ which is the result of multiplying the input vector by an orthogonal matrix $M$, thus converting a function from separable to non-separable over $x^r$. The rotate operation $F_{\text{elliptic}}(R x)$ makes the elliptic function non-separable. The orthogonal matrix $M$ is defined in the implementation details in [141].

**Shift** ($S$) creates a vector $x^s = x - o^g$ that translates the global minimum of the function. The vector $o^g$ is defined as $o^g = G_{i}^{|x|} o$, where $o$ is a vector of size $d$, and $i$ indicates the starting index of vector $o^g$ such that it matches the size of vector $x$. The shift operator takes a lower index to define $i$. The vector $o$ is randomly generated and defined in the implementation details in [141].

Additionally, combinations of the operators are possible by concatenation. For example, considering $F(x) = F_{\text{elliptic}}(R S G P x)$, the sequence of operations is as follows: a) shuffle vector $x$ producing $x^p = P x$, b) truncate vector $x^p$ to size $m$ producing $x^g = G_0^m x^p$, c) subtract vector $G_0^{|x|} o$ from $x^g$ producing $x^s = S_0^{|x|} x^g$, d) rotate vector $x^g$ producing $x^r = R x^s$, and e) compute the elliptic function on $x^r$ producing $F(x) = F_{\text{elliptic}}(x^r)$.

In this section, we present each of the 20 benchmark functions in [141] using our proposed notation, and detailing the modality of the function, the boundaries of each dimension of the solutions, the location of the optimum solution, and the objective value of the optimum solution.
2.8. BENCHMARK FUNCTIONS

2.8.3 Separable Functions \((F_{01-03})\)

\(F_{01}:\) shifted elliptic function

\[
F_{01}(x) = F_{\text{elliptic}}(\mathbb{S}x)_0
\]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal</td>
<td>(x \in [-100, 100]^d)</td>
<td>(x^* = 0)</td>
<td>(F_{01}(x^*) = 0)</td>
</tr>
</tbody>
</table>

\(F_{02}:\) shifted rastrigin's function

\[
F_{02}(x) = F_{\text{rastrigin}}(\mathbb{S}x)_0
\]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>(x \in [-5, 5]^d)</td>
<td>(x^* = 0)</td>
<td>(F_{02}(x^*) = 0)</td>
</tr>
</tbody>
</table>

\(F_{03}:\) shifted ackley's function

\[
F_{03}(x) = F_{\text{ackley}}(\mathbb{S}x)_0
\]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>(x \in [-32, 32]^d)</td>
<td>(x^* = 0)</td>
<td>(F_{03}(x^*) = 0)</td>
</tr>
</tbody>
</table>

2.8.4 Single-Group of \(m\) Non-Separable Functions \((F_{04-08})\)

\(F_{04}:\) single-group shifted and \(m\)-rotated elliptic function

\[
F_{04}(x) = 10^6 \times F_{\text{elliptic}}(\mathbb{R} \mathbb{S} \mathbb{G}^m \mathbb{P}_0 x) + F_{\text{elliptic}}(\mathbb{S}^d \mathbb{G}^m \mathbb{P}_m x)
\]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal</td>
<td>(x \in [-100, 100]^d)</td>
<td>(x^* = 0)</td>
<td>(F_{04}(x^*) = 0)</td>
</tr>
</tbody>
</table>
CHAPTER 2. LITERATURE REVIEW

F₀₅: single-group shifted and m-rotated rastrigin’s function

\[ F_{05}(x) = 10^6 \times F_{\text{rastrigin}} \left( \mathbf{R} \mathbf{S} \mathbf{G} \mathbf{P} x \right) + F_{\text{rastrigin}} \left( \mathbf{S} \mathbf{G} \mathbf{P} x \right) \]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>( x \in [-5, 5]^d )</td>
<td>( x^* = 0 )</td>
<td>( F_{05}(x^*) = 0 )</td>
</tr>
</tbody>
</table>

F₀₆: single-group shifted and m-rotated ackley’s function

\[ F_{06}(x) = 10^6 \times F_{\text{ackley}} \left( \mathbf{R} \mathbf{S} \mathbf{G} \mathbf{P} x \right) + F_{\text{ackley}} \left( \mathbf{S} \mathbf{G} \mathbf{P} x \right) \]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>( x \in [-32, 32]^d )</td>
<td>( x^* = 0 )</td>
<td>( F_{06}(x^*) = 0 )</td>
</tr>
</tbody>
</table>

F₀₇: single-group shifted m-dimensional schwefel’s problem 1.2

\[ F_{07}(x) = 10^6 \times F_{\text{schwefel}} \left( \mathbf{S} \mathbf{G} \mathbf{P} x \right) + F_{\text{sphere}} \left( \mathbf{S} \mathbf{G} \mathbf{P} x \right) \]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal</td>
<td>( x \in [-100, 100]^d )</td>
<td>( x^* = 0 )</td>
<td>( F_{07}(x^*) = 0 )</td>
</tr>
</tbody>
</table>

F₀₈: single-group shifted m-dimensional rosenbrock’s function

\[ F_{08}(x) = 10^6 \times F_{\text{rosenbrock}} \left( \mathbf{S} \mathbf{G} \mathbf{P} x \right) + F_{\text{sphere}} \left( \mathbf{S} \mathbf{G} \mathbf{P} x \right) \]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>( x \in [-100, 100]^d )</td>
<td>( x^* = 1 + \mathbf{G} \mathbf{P} 0 )</td>
<td>( F_{08}(x^*) = 0 )</td>
</tr>
</tbody>
</table>
2.8. BENCHMARK FUNCTIONS

2.8.5 $\frac{d}{2m}$ Groups of $m$ Non-Separable Functions ($F_{09-13}$)

$F_{09}$: $\frac{d}{2m}$-group shifted and $m$-rotated elliptic function

$$F_{09}(x) = \sum_{k=0}^{d/2m-1} F_{\text{elliptic}}\left(\mathbb{R}^\frac{(k+1)\times m}{k\times m} G x\right) + F_{\text{elliptic}}\left(\frac{d}{d/2 d/2} P x\right)$$

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal</td>
<td>$x \in [-100, 100]^d$</td>
<td>$x^* = 0$</td>
<td>$F_{09}(x^*) = 0$</td>
</tr>
</tbody>
</table>

$F_{10}$: $\frac{d}{2m}$-group shifted and $m$-rotated rastrigin’s function

$$F_{10}(x) = \sum_{k=0}^{d/2m-1} F_{\text{rastrigin}}\left(\mathbb{R}^\frac{(k+1)\times m}{k\times m} G x\right) + F_{\text{rastrigin}}\left(\frac{d}{d/2 d/2} P x\right)$$

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>$x \in [-5, 5]^d$</td>
<td>$x^* = 0$</td>
<td>$F_{10}(x^*) = 0$</td>
</tr>
</tbody>
</table>

$F_{11}$: $\frac{d}{2m}$-group shifted and $m$-rotated ackley’s function

$$F_{11}(x) = \sum_{k=0}^{d/2m-1} F_{\text{ackley}}\left(\mathbb{R}^\frac{(k+1)\times m}{k\times m} G x\right) + F_{\text{ackley}}\left(\frac{d}{d/2 d/2} P x\right)$$

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>$x \in [-32, 32]^d$</td>
<td>$x^* = 0$</td>
<td>$F_{11}(x^*) = 0$</td>
</tr>
</tbody>
</table>

$F_{12}$: $\frac{d}{2m}$-group shifted and $m$-rotated schwefel’s problem 1.2

$$F_{12}(x) = \sum_{k=0}^{d/2m-1} F_{\text{schwefel}}\left(\mathbb{R}^\frac{(k+1)\times m}{k\times m} G x\right) + F_{\text{sphere}}\left(\frac{d}{d/2 d/2} P x\right)$$

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal</td>
<td>$x \in [-100, 100]^d$</td>
<td>$x^* = 0$</td>
<td>$F_{12}(x^*) = 0$</td>
</tr>
</tbody>
</table>
**F_{13}: \frac{d}{2m} \text{-group shifted and } m\text{-rotated rosenbrocks’s function}**

\[
F_{13}(x) = \sum_{k=0}^{\frac{d}{2m}-1} F_{\text{rosenbrock}} \left( \begin{array}{c} S \\ k \times m \end{array} \right) \left( \begin{array}{c} (k+1) \times m \\ G \times k \times m \end{array} \right) x + F_{\text{sphere}} \left( \begin{array}{c} d \\ \frac{d}{2} \end{array} \right) \left( \begin{array}{c} G \times m \\ k \times m \end{array} \right) x
\]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>(x \in [-100, 100]^d)</td>
<td>(\frac{d}{2} G P x^* = 1 + G P o) (\frac{d}{2})</td>
<td>(F_{13}(x^*) = 0)</td>
</tr>
</tbody>
</table>

**2.8.6 \(\frac{d}{m}\) Groups of \(m\) Non-Separable Functions \((F_{14-18})\)**

**F_{14}: \frac{d}{m} \text{-group shifted and } m\text{-rotated elliptic function}**

\[
F_{14}(x) = \sum_{k=0}^{\frac{d}{m}-1} F_{\text{elliptic}} \left( \begin{array}{c} R \\ k \times m \end{array} \right) \left( \begin{array}{c} (k+1) \times m \\ S \times k \times m \end{array} \right) x
\]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal</td>
<td>(x \in [-100, 100]^d)</td>
<td>(x^* = o)</td>
<td>(F_{14}(x^*) = 0)</td>
</tr>
</tbody>
</table>

**F_{15}: \frac{d}{m} \text{-group shifted and } m\text{-rotated rastrigin’s function}**

\[
F_{15}(x) = \sum_{k=0}^{\frac{d}{m}-1} F_{\text{rastrigin}} \left( \begin{array}{c} R \\ k \times m \end{array} \right) \left( \begin{array}{c} (k+1) \times m \\ S \times k \times m \end{array} \right) x
\]

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>(x \in [-5, 5]^d)</td>
<td>(x^* = o)</td>
<td>(F_{15}(x^*) = 0)</td>
</tr>
</tbody>
</table>

**F_{16}: \frac{d}{m} \text{-group shifted and } m\text{-rotated ackley’s function}**

\[
F_{16}(x) = \sum_{k=0}^{\frac{d}{m}-1} F_{\text{ackley}} \left( \begin{array}{c} R \\ k \times m \end{array} \right) \left( \begin{array}{c} (k+1) \times m \\ S \times k \times m \end{array} \right) x
\]
### 2.8. BENCHMARK FUNCTIONS

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>$x \in [-32, 32]^d$</td>
<td>$x^* = o$</td>
<td>$F_{16}(x^*) = 0$</td>
</tr>
</tbody>
</table>

$F_{17}$: $\frac{d}{m}$-group shifted and $m$-dimensional schwefel’s problem 1.2

$$F_{17}(x) = \sum_{k=0}^{d/m-1} F_{\text{schwefel}} \left( \sum_{k \times m}^{(k+1)\times m} \mathbb{S} \mathbb{G} \mathbb{P} x \right)$$

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal</td>
<td>$x \in [-100, 100]^d$</td>
<td>$x^* = o$</td>
<td>$F_{17}(x^*) = 0$</td>
</tr>
</tbody>
</table>

$F_{18}$: $\frac{d}{m}$-group shifted $m$-dimensional rosenbrocks’s function

$$F_{18}(x) = \sum_{k=0}^{d/m-1} F_{\text{rosenbrock}} \left( \sum_{k \times m}^{(k+1)\times m} \mathbb{S} \mathbb{G} \mathbb{P} x \right)$$

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimodal</td>
<td>$x \in [-100, 100]^d$</td>
<td>$x^* = o + 1$</td>
<td>$F_{18}(x^*) = 0$</td>
</tr>
</tbody>
</table>

#### 2.8.7 Non-Separable Functions ($F_{19,20}$)

$F_{19}$: schwefel’s problem 1.2

$$F_{19}(x) = F_{\text{schwefel}} \left( \sum_{0} \mathbb{S} x \right)$$

<table>
<thead>
<tr>
<th>Modality</th>
<th>Boundaries</th>
<th>Solution</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal</td>
<td>$x \in [-100, 100]^d$</td>
<td>$x^* = o$</td>
<td>$F_{19}(x^*) = 0$</td>
</tr>
</tbody>
</table>

$F_{20}$: shifted rosenbrock’s function

$$F_{20}(x) = F_{\text{rosenbrock}} \left( \sum_{0} \mathbb{S} x \right)$$
### 2.9 Summary

This chapter has presented an overview of EC with an emphasis on SI, which is the class of algorithms to which PSO belongs given its source of inspiration. Moreover, a comprehensive overview of optimization problems subject to noise has been covered and distinguished from other optimization problems according to the type of uncertainty present, thereby clearly delimiting the scope of the problems addressed in this thesis. The relevance of noisy optimization problems has been explored in the field of simulation-based optimization, where solutions to real-world problems are optimized based on evaluations that contain stochastic components due to simulation models. In this type of problem, the performance of PSO has been acknowledged to present two issues, namely the inaccurate memories of the particles and their most likely incorrect selection of the neighborhood best solutions. These challenges make the effect of noise on PSO inherently different from that on other metaheuristics such as GAs and ES. First, particles store in memory solutions with inaccurately estimated objective values, and that will affect the frequency at which they find better solutions. Second, particles select the neighborhood best solutions from the solutions stored in memory, for which poor decisions made by an individual particle will potentially affect the decisions of those within the neighborhood.

The noise mitigation mechanisms incorporated into PSO can be classified based on whether they perform additional evaluations to the solutions to better estimate them. If they perform additional evaluations, we refer to them as resampling-based PSO algorithms, and several works have been explored in the literature suggesting PSO-OCBA is the best algorithm available to deal with this type of problem. Conversely, if the
PSO algorithm does not perform any additional evaluation, we classify such an approach as a single-evaluation method given that the estimated objective values of the solutions are based solely on the single evaluation that the PSO algorithm dictates. Several works have also explored the performance of single-evaluation methods, all of which make use of the evaporation mechanism to worsen the quality of the personal best solutions and thereby encourage their replacement. While both resampling and single-evaluation methods have shown improvements on the quality of the results, their experimental design has not been too rigorous and, more often than not, their conclusions end up based mostly on intuition supporting hypothetical expectations rather than confirming them based on empirical results.

Lastly, considering the few benchmark functions upon which previous works have evaluated the performance of their algorithms, this chapter has presented the 20 large-scale benchmark functions on which we will evaluate the algorithms in this thesis. The benchmark functions make up the entire suite presented at the CEC’2010 Special Session and Competition on Large-Scale Global Optimization [141], but have been presented with a different notation aiming to provide a less verbose mathematical representation than the one originally proposed.
Chapter 3

Deception, Blindness and Disorientation

This chapter identifies and clearly defines deception, blindness and disorientation as three conditions responsible for the deterioration of the quality of the results of PSO on optimization problems subject to noise. Additionally, this chapter introduces the concept of population statistics to measure the extents to which these conditions (and other characteristics) affect the particles in the swarm throughout the search process. The population statistics are first studied for the regular PSO and PSO-ER, both additionally under the assumptions of local and global certainty.

This chapter is structured as follows. Section 3.1 introduces this chapter. Section 3.2 presents the population statistics and the definitions of deception, blindness and disorientation. Section 3.3 describes the design of experiments. Section 3.4 presents the results and discussions. Finally, Section 3.5 ends this chapter with a summary.

3.1 Introduction

One characteristic that has remained largely unexplored in PSO is the deterioration of the quality of its results on optimization problems subject
to noise. In this type of problem, the objective values of the solutions are corrupted by the effect of sampling noise, thus causing solutions to have inaccurate objective values that change every time these are evaluated. As a consequence, particles eventually fail to distinguish good from bad solutions, leading in turn to other issues that ultimately end up deteriorating the quality of the results. The literature has recognized two such issues as particles having inaccurate memories and particles failing to select of their true neighborhood best solutions, both of which have been addressed respectively by incorporating evaporation mechanisms [30, 31, 32, 50, 51] and resampling methods [7, 106, 113, 154] into PSO.

The incorporation of noise mitigation mechanisms into PSO has shown to provide significant improvements on the quality of the results [7, 30, 31, 32, 50, 51, 106, 113, 154]. However, the analyses behind such improvements often fall short of empirical evidence supporting the claims made about the underlying reasons for their achievements in terms other than the quality of the results. More importantly, there is not even evidence showing the extents to which inaccurate memories affect the particles in the swarm or the frequency at which particles fail to select their true neighborhood best solutions. Therefore, given the lack of such evidence, we want to investigate the effect of noise on PSO beyond the deterioration of its results. Furthermore, we expect that such information will not only help to choose the best noise mitigation mechanism, but will also help to design even better ones.

In this chapter, we distinguish the effect of noise on particles as deception when they fail to select their true neighborhood best solutions, blindness when they miss out on opportunities to improve upon their own best solutions, and disorientation when they mistakenly prefer worse solutions. While deception is just a new name to refer to the incorrect selection of neighborhood best solutions, blindness and disorientation are new concepts that clearly define the problem of inaccurate memories. Furthermore, we design in this chapter a set of population statistics to count the
frequency at which particles suffer from these conditions throughout the search process.

**Chapter Goals**

The overall goal of this chapter is to study the population statistics for PSO and PSO-ER on 20 large-scale benchmark functions subject to different levels of multiplicative Gaussian noise. Specifically, we will focus on the following objectives.

- Define the conditions that lead particles to suffer from deception, blindness and disorientation.
- Design a useful set of population statistics for PSO algorithms.
- Compare the population statistics for regular PSO and PSO-ER.
- Compare the population statistics for regular PSO and PSO-ER under the assumptions of local and global certainty.

### 3.2 Population Statistics for PSO

*Population statistics* is the term by which we refer to a set of indicators providing information about the particles in the swarm throughout the search process. The goal of these statistics is to provide empirical evidence about the effect of noise on PSO beyond the known deterioration of the quality of its results. To this end, we identify *deception, blindness* and *disorientation* as the three conditions responsible for such a deterioration, all of which originate because the objective values of all the solutions are either underestimated or overestimated due to the effect of noise. Sections 3.2.1, 3.2.2 and 3.2.3 define these conditions and present their respective statistics. Additionally, we define the *regular operation* of a particle and its statistic in Section 3.2.4, the *lifetime* of the swarm in Section 3.2.5, and the *quality*
of results in Section 3.2.6. The population statistics presented in this section are for a swarm $S$ after performing $t_{\text{max}}$ iterations on a minimization problem. The computation of the population statistics requires the true objective values of the solutions to be known in order to properly identify and quantify the effect of noise on the swarm.

### 3.2.1 Deception

Particle $i$ is deceived by its neighbors in $N_i$ at iteration $t$ when $i$ selects its neighborhood best solution $\hat{y}_t^i$ based on the estimated objective values of their personal best solutions, but $\hat{y}_t^i$ does not correspond to the true best solution found in $N_i$. Consequently, a deceived particle will be partially attracted towards a sub-optimal region of the search space. The statistic for deception (3.1) corresponds to the average number of iterations at which a particle is deceived.

\[
I_D(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} \hat{D}_t^i, \quad \text{where } \hat{D}_t^i = \begin{cases} 
1, & \text{if } \hat{y}_t^i \neq \arg \min_{y^t \in N_i} f(y^t) \\
0, & \text{otherwise}
\end{cases} 
\] (3.1)

Deception is a condition that has been previously addressed in the literature as the problem of correctly selecting the neighborhood best solution [7, 106, 113, 122, 123, 154].

### 3.2.2 Blindness

Particle $i$ is blinded at iteration $t$ when the true objective value of its current solution $x_t^i$ is better than that of its personal best solution $y_t^i$, but the effect of noise on either solution suggests otherwise and hence the particle discards $x_t^i$ by deciding not to update $y_t^i$. Consequently, a blinded particle misses out on an opportunity to improve upon its personal best solution. The concept of blindness is new to the PSO domain, and its
3.2. POPULATION STATISTICS FOR PSO

Statistic (3.2) corresponds to the average number of iterations at which a particle is blinded.

\[ I_B(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} B^t_i, \quad \text{where } B^t_i = \begin{cases} 1, & \text{if } f(x^t_i) < f(y^t_i) \\ 0, & \text{otherwise} \end{cases} \]  

(3.2)

Additionally, for a blinded particle \( i \), we attribute the cause of its blindness to the environment if the objective value of its current solution \( x^t_i \) is less accurate than that of its personal best solution \( y^t_i \). Otherwise, its blindness is due to its own memory. Hence, the cause of blindness provides information to design noise mitigation mechanisms to better estimate either the current or the personal best solutions. The statistics for blindness caused by the environment (3.3) and by memory (3.4) correspond to the average number of iterations at which each of these causes is attributable to a blinded particle.

\[ I_{B_e}(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} B^t_i \cdot B^t_{e_i}, \quad \text{where } B^t_{e_i} = \begin{cases} 1, & \text{if } \left| f(x^t_i) - \hat{f}(x^t_i) \right| > \left| f(y^t_i) - \hat{f}(y^t_i) \right| \\ 0, & \text{otherwise} \end{cases} \]  

(3.3)

\[ I_{B_m}(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} B^t_i \cdot B^t_{m_i}, \quad \text{where } B^t_{m_i} = \begin{cases} 1, & \text{if } \left| f(x^t_i) - \hat{f}(x^t_i) \right| \leq \left| f(y^t_i) - \hat{f}(y^t_i) \right| \\ 0, & \text{otherwise} \end{cases} \]  

(3.4)

3.2.3 Disorientation

Particle \( i \) is disoriented at iteration \( t \) when the true objective value of its current solution \( x^t_i \) is worse than that of its personal best solution \( y^t_i \), but the effect of noise on either solution suggests otherwise and hence the particle
decides to update $y^t_i$. Consequently, a disoriented particle mistakenly updates its personal best solution to a worse solution. The concept of disorientation is also new to the PSO domain, and its statistic (3.5) corresponds to the average number of iterations at which a particle is disoriented.

$$I_D(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} D^t_i,$$  \quad \text{where} \quad D^t_i = \begin{cases} 1, & \text{if } f(y^t_i) > f(y^{t-1}_i) \\ 0, & \text{otherwise} \end{cases} \quad (3.5)$$

Additionally, for a disoriented particle $i$, we attribute the cause of its disorientation to the *environment* if the objective value of its current solution $x^t_i$ is less accurate than that of its personal best solution $y^t_i$. Otherwise, its disorientation is due to its own *memory*. Hence, the cause of disorientation provides information to design noise mitigation mechanisms to better estimate either the current or the personal best solutions. The statistics for disorientation caused by the environment (3.6) and by memory (3.7) correspond to the average number of iterations at which each of these causes is attributable to a disoriented particle.

$$I_{D_e}(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} D^t_i \cdot D_{ei}^t,$$  \quad \text{where} \quad D_{ei}^t = \begin{cases} 1, & \text{if } \left| f(x^t_i) - \tilde{f}(x^t_i) \right| > \left| f(y^t_i) - \tilde{f}(y^t_i) \right| \\ 0, & \text{otherwise} \end{cases} \quad (3.6)$$

$$I_{D_m}(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} D^t_i \cdot D_{mi}^t,$$  \quad \text{where} \quad D_{mi}^t = \begin{cases} 1, & \text{if } \left| f(x^t_i) - \tilde{f}(x^t_i) \right| \leq \left| f(y^t_i) - \tilde{f}(y^t_i) \right| \\ 0, & \text{otherwise} \end{cases} \quad (3.7)$$

### 3.2.4 Regular Operation

Particle $i$ presents a *regular operation* when it is neither blinded nor disoriented, that is, when it correctly discards its current solution provided it
3.2. POPULATION STATISTICS FOR PSO

is truly worse than its personal best solution, or when it updates its personal best solution provided that the current solution is truly better. Thus, particles always have a regular operation in the absence of noise, but not necessarily in its presence. The statistic for regular operation (3.8) corresponds to the average number of iterations at which a particle performs a regular update or discard, each of which is represented in statistics (3.9) and (3.10), respectively.

\[ I_R(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} (R_{ui}^t + R_{di}^t) \]  

(3.8)

\[ I_{R_u}(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} R_{ui}^t, \quad \text{where } R_{ui}^t = \begin{cases} 
1, & \text{if } x_i^t = y_i^t \land f(x_i^t) < f(y_i^{t-1}) \\
0, & \text{otherwise}
\end{cases} \]  

(3.9)

\[ I_{R_d}(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} R_{di}^t, \quad \text{where } R_{di}^t = \begin{cases} 
1, & \text{if } x_i^t \neq y_i^t \land f(x_i^t) \geq f(y_i^{t-1}) \\
0, & \text{otherwise}
\end{cases} \]  

(3.10)

3.2.5 Lifetime

The lifetime of the swarm refers to the average last iteration at which a particle updates its personal best solution. Its statistic (3.11) provides useful information to determine the moment just before the swarm either converges to a solution, stagnates due to blindness, or simply cannot find better solutions any more.

\[ I_L(S) = \frac{1}{|S|} \sum_{i \in S} L_i^{t_{\text{max}}}, \quad \text{where } L_i^t = \begin{cases} 
t, & \text{if } y_i^t \neq y_i^{t-1} \\
L_i^{t-1}, & \text{otherwise}
\end{cases} \]  

(3.11)
3.2.6 Quality of Results

The quality of results is measured by the true objective value of the true best solution found and retained by the swarm over the iterations. While it may seem unrealistic to correctly select the true best solution in optimization problems subject to noise, our goal is to report the objective values of the true best solution in order to provide a baseline reference where any other solution in the swarm will certainly be worse. Furthermore, since particles operate with estimated objective values in the presence of noise, the global optimum solution could be found and yet be ignored due to blindness, or could be stored and yet be later replaced due to disorientation. Hence, the quality of results is made up of the true best solution that is stored in memory at the last iteration (3.12). Notice that, in the absence of noise, the true best solution will never be ignored or replaced and hence will always remain in memory after being found by a particle.

\[ I_Q(S) = \min_{i \in S} f(y_{i}^{t_{\text{max}}}) \]  

(3.12)

3.3 Design of Experiments

The algorithms for which we want to study the population statistics on optimization problems subject to noise are the regular PSO and PSO-ER, both additionally under the assumptions of local and global certainty. Specifically, we are interested in understanding the population statistics for these algorithms in order to provide insights about the effect of noise beyond the quality of their results. Thus, the algorithms to compare are PSO, PSO-LC, PSO-GC, PSO-ER, PSO with Equal Resampling and Local Certainty (PSO-ERLC), and PSO with Equal Resampling and Global Certainty (PSO-ERGC).

The configuration of the swarms is rather minimalistic in order to reduce the number of variables that influence the performance of the al-
3.3. DESIGN OF EXPERIMENTS

The swarms are all made up of 50 particles to provide a better coverage of the search space than the 30 or fewer particles suggested in the literature \([18,49,145]\). The neighborhoods are defined by the star topology in order to have a single neighborhood best solution to which all particles are partially attracted. The acceleration coefficients \(c_1 = c_2 = 1.49618\) and the inertia \(w = 0.729844\) of the particles are set according to the guidelines in \([145]\). Particles start with their velocity vectors initialized to zero as suggested in \([47]\) and limit their velocities utilizing Equation (3.13) in order to reduce the sensitivity of setting a maximum velocity \(v_{\text{max}}\) \([48]\), which is determined as shown in Equation (3.14) according to the limits \(x_{\text{min}}\) and \(x_{\text{max}}\) of the optimization problem. The total computational budget is set for each swarm to 30,000 function evaluations given that it divides nicely into \(30,000 / 50 = 600\) iterations for PSO without resampling, and \(30,000 / (6 \times 50) = 100\) iterations for PSO-ER performing \(b_i = 6\) function evaluations on every current solution at each iteration, amongst other settings. The computational budget of 30,000 evaluations is shown to be large enough for the problems addressed as it is discussed in Section 3.4.2 (page 75) and Section 7.3.3 (page 259). The number of function evaluations of PSO-ER is set to \(b_i = 6\) considering the guidelines in \([24,25]\), which suggest to utilize at least \(b_i = 5\) function evaluations to estimate the objective values of the solutions. The list of parameter settings is shown in Table 3.1.

\[
\dot{v}_{ij}^t = v_{\text{max}} \cdot \tanh \left( \frac{v_{ij}^t}{v_{\text{max}}} \right) \quad (3.13)
\]

\[
v_{\text{max}} = 0.25 \cdot |x_{\text{max}} - x_{\text{min}}| \quad (3.14)
\]

The Random Number Generators (RNGs) are strictly controlled within the experiments not only to ensure reproducibility, but also to create the exact same conditions for all the algorithms such that they are fairly compared. Each swarm has an RNG to randomly create the initial solutions of every independent run. The particles of each swarm have two RNGs from which the random values \(r_1^t\) and \(r_2^t\) of Equation (2.5) are sampled, and
Table 3.1: Parameter settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent runs</td>
<td>50 with 30000 function evaluations</td>
</tr>
<tr>
<td>Number of particles</td>
<td>50 in $\mathbb{R}^{1000}$ with star topology</td>
</tr>
<tr>
<td>Acceleration</td>
<td>Static with $c_1 = c_2 = 1.49618$</td>
</tr>
<tr>
<td>Inertia</td>
<td>Static with $w = 0.729844$</td>
</tr>
<tr>
<td>Severity of noise</td>
<td>$\sigma \in {0.00, 0.06, 0.12, 0.18, 0.24, 0.30}$</td>
</tr>
<tr>
<td>Resampling budget</td>
<td>$b_i = 6$</td>
</tr>
</tbody>
</table>

their respective seeds change at every independent run. Thus, at every independent run, each swarm has different independent RNGs that are exactly the same for every other swarm, thereby ensuring that all the algorithms are under the exact same conditions and that the results obtained are due to the internal operation of the algorithms and not to exclusive circumstantial conditions caused by RNGs.

The algorithms are evaluated on the benchmark functions presented in Section 2.8 subject to multiplicative Gaussian noise, thereby converting them into backward optimization problems. The algorithms perform 50 independent runs on each benchmark at different levels of noise ranging from very low to very high as $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$, and PSO without resampling is additionally evaluated in the absence of noise. At each level of noise, the noise samples are ensured to lie within $3\sigma$ (by resampling, if needed) in order to keep the objective values only positive. Thus, for example, the objective values of a solution will be corrupted at most by a factor of $1.0 \pm 0.9$ (i.e. up to 90% of its true objective value) when $\sigma = 0.30$.

The population statistics are computed for the swarm of each algorithm utilizing the results collected in all independent runs. Thus, at each level of noise, the population statistics for PSO, PSO-LC and PSO-GC are
3.4 Results and Discussions

Our discussions are based on the population statistics for the algorithms on benchmark function $F_{01}$ because these reflect most of the trends found in the other benchmark functions, all of which are presented in Appendix 3.A (page 92). The results and discussions are structured as follows. In Section 3.4.1, we present the population statistics for the algorithms on $F_{01}$. In Section 3.4.2, we discuss the statistics of PSO and PSO-ER. In Sections 3.4.3 and 3.4.4 we contrast the statistics of PSO and PSO-ER each under the assumptions of local and global certainty. Finally, in Section 3.4.5 we present and discuss a summary of the statistics on all the benchmark functions.

3.4.1 Population Statistics

The population statistics on $F_{01}$ are presented in Figure 3.1, where the algorithms are abbreviated as (a) PSO, (al) PSO-LC, (ag) PSO-GC, (e) PSO-ER, (el) PSO-ERLC, and (eg) PSO-ERGC. The statistics in this figure are referred to hereinafter by their names in italics, and these are presented as follows.

Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) on $F_{01}$ subject to the different levels of noise (top axis). The subdivisions at each level of noise separate the algorithms without resampling from those with resampling. The algorithms in the first subdivision correspond to (a) PSO, (l) PSO-LC and (g) PSO-GC, and those in the second subdivision to (e) PSO-ER, (l) PSO-ERLC and (g) PSO-ERGC. The boxplots are...
coloured as follows to ease the comparison: PSO and PSO-ER (white), PSO-LC and PSO-ERLC (light gray), and PSO-GC and PSO-ERGC (dark gray). The benchmark functions are minimization problems and therefore lower objective values indicate better solutions.

**Lifetime.** The barplots represent the normalized average lifetime (left axis) of a particle for each algorithm (bottom axis) on $F_{01}$ subject to the different levels of noise (bars coloured from light to dark gray). A longer lifetime is better when the swarm does not converge to the global optimum. The normalized average lifetime refers to the ratio between the average lifetime and the number of iterations performed by the algorithm.

**Deception.** The barplots represent the average proportion of iterations (left axis) in which a particle for each algorithm (bottom axis) is deceived by its neighbors on $F_{01}$ subject to the different levels of noise (bars coloured from light to dark gray). Smaller proportions are better.

**Regular Operation, Blindness and Disorientation.** The stacked barplots represent the average proportions (left axis) of regular operations (dark gray), blindness (medium gray) and disorientation (light gray) experienced by a particle for each algorithm (bottom axis) on $F_{01}$ subject to the different levels of noise (bars from left to right). Larger proportions of regular operations and smaller proportions of blindness and disorientation are better.

**Regular Updates and Discards.** The stacked barplots represent the average proportions (left axis) of regular updates (dark gray) and discards (light gray) with respect to the regular operations experienced by a particle for each algorithm (bottom axis) on $F_{01}$ subject to the different levels of noise (bars from left to right). Larger proportions of regular updates and smaller proportions of regular discards are better.

**Causes of Blindness.** The stacked barplots represent the average proportions of blindness (left axis) caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on $F_{01}$ subject to the different levels of noise (bars from left to right).
3.4. RESULTS AND DISCUSSIONS

*Causes of Disorientation.* The stacked barplots represent the average proportions of disorientation (left axis) caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on $F_{01}$ subject to the different levels of noise (bars from left to right).

From these statistics, notice that PSO in the absence of noise is also presented and therefore PSO (a) has an additional bar which is relevant in *lifetime, regular operation* and *regular updates and discards*. Additionally, the bars in *deception* for PSO-GC (ag) and PSO-ERGC (eg) are empty because their particles do not suffer from deception as they have global certainty. Likewise, the bars in *causes of blindness* and *causes of disorientation* for PSO-LC (al) and PSO-ERLC (el) are empty because their particles do not suffer from blindness or disorientation as they have local certainty.

### 3.4.2 PSO with and without Resampling

The *quality of results* shows that PSO in the absence of noise finds better solutions than any of the other algorithms in the presence of noise. However, in the presence of noise, PSO-ER is significantly better than PSO even when it performs 100 iterations compared to the 600 that PSO performs. Nonetheless, both algorithms deteriorate in the presence of noise and higher levels of noise lead to further deterioration. These results show the sensitivity of PSO to optimization problems subject to noise and the importance of improving the accuracy of the objective values in order to provide a significantly better quality of results. The underlying reasons for such a difference are mostly found within the following statistics.

The *lifetime* of PSO is generally shorter than that of PSO-ER by about 10%, thus indicating that particles in PSO tend to stagnate earlier than those in PSO-ER. We attribute such a shorter lifetime partly to the larger proportions of blindness in PSO that prevent particles from updating to better solutions, and partly to the larger number of iterations performed by PSO which renders its convergence more likely. Notice that the dif-
Figure 3.1: Population statistics on benchmark function $F_{01}$. The presentation details of each statistic are described in Section 3.4.1 (page 73).
ference in lifetime between both algorithms is expressed in proportions, and as such, it means that particles in PSO spend a much larger number of iterations without finding better solutions. Moreover, as the level of noise becomes higher, the lifetime in both algorithms further shortens due to the increasing proportions of blindness. In general, the lifetime of the swarms suggests that they reach convergence far before performing 100% of the iterations, thereby supporting our choice of setting the computational budget to $30\,000$ function evaluations.

The proportions of deception in both algorithms are very high (generally over 95%), but those in PSO-ER are slightly smaller (over 85%). While we expected PSO-ER to present a much smaller proportion of deception because the objective values of its solutions are more accurate, these results show that particles require even more re-evaluations to better estimate their objective values and hence increase the chance of correctly selecting the neighborhood best solutions. Nonetheless, the greater accuracy of PSO-ER does lead particles to generally select better neighborhood solutions than those in PSO, and this can be confirmed by utilizing a deception indicator measuring ranks instead of the binary selection presented here.

The proportions of regular operations are larger in PSO-ER (over 50%) than in PSO (below 50%) because the estimated objective values are more accurate in the former. Consequently, particles in PSO-ER are more likely to correctly distinguish good from bad solutions, and thereby correctly either update their personal best solutions or discard their current solutions. In the absence of noise, particles in PSO do not suffer from blindness or disorientation and hence all of them present only regular operations. However, as the level of noise increases, the proportions of regular operations in PSO-ER progressively reduce as expected, but those in PSO show subtle differences with a small increase at $\sigma = 0.30$. While we would need to study in more detail the reasons for such a sudden increase of regular operations in PSO, the subtle differences just indicate that very low levels of noise already disrupt the regular operation of PSO in $F_{01}$, and hence
higher levels of noise will not have a major influence except for the quality of the results which deteriorates significantly.

The proportions of regular updates and discards in the absence of noise show that particles in PSO fail to find better solutions in over 50% of the iterations performed. Such a high proportion of failure (without finding the global optimum solution) is due to the exploration mechanism determined by the velocity update in Equation (2.3) and its respective clamping. This suggests that it would be useful to explore the performance of other mechanisms such as the constriction factor proposed in [26] or design new mechanisms including more information about the problem upon updating the velocity (e.g. proportionally weighting the personal and neighborhood best solutions according to their objective values). The proportions of regular updates in the presence of noise plummet below 10% for PSO, thus remarking the sensitivity of PSO to optimization problems subject to noise. Differently, those proportions in PSO-ER are over 30%, decreasing as the level of noise increases, yet remaining over 10%.

The proportions of blindness are smaller in PSO-ER (below 50%) than in PSO (over 50%) due to the greater accuracy to which the objective values of solutions in the former are estimated. The proportions of blindness in both algorithms are rather high considering that these are missed opportunities to improve upon the best solutions found. The underlying reason to such high proportions of blindness is that particles are always prone to suffer from blindness at the next iteration regardless of what they do at the current iteration. Specifically, if a particle suffers from blindness or disorientation, or even if it has a regular operation at iteration $t$, the particle is still prone to suffer from blindness to specific ranges of solutions at $t + 1$ (see Figure 3.2). The most common cause of blindness is blindness by memory, which is responsible for over 98% of the cases in PSO, and it ranges between 85–96% in PSO-ER increasing with the level of noise. Blindness by memory is vastly more common than blindness by the environment because the direction of the optimization problem is backwards,
and hence the current solutions which are truly better than the personal best solutions will generally be more accurate as the proportional effect of noise will be smaller.

The proportions of disorientation in PSO-ER are larger (below 5%) than those in PSO (below 1%), but their presence in both algorithms is rather small compared to the proportions of regular operations and blindness. These proportions refer to those particles which mistakenly update their personal best solutions to worse solutions, thus causing a setback to the search process. The underlying reasons for such small proportions of disorientation are a) the direction of the optimization problem, b) the underestimation of objective values from the personal best solutions (pre-existing blindness), and c) the landscape of the objective search space affected by the noise therein. Firstly, the direction of the optimization problem being backwards causes better solutions to be less corrupted by noise, and hence the overlapping of the probability density functions (representing the noisy objective values) of adjacent solutions will be smaller. As such, the proportions of disorientation in backward optimization problems will occur mostly at early iterations when the objective values of the solutions are generally bad and hence more likely to overlap (see Figure 3.3). Secondly, pre-existing blindness creates an inverse correlation between blindness and disorientation because larger underestimations of the objective values from the personal best solutions will require even larger underestimations of those from the current solutions; this, together with the backward direction of the optimization problem, will make disorientation even less likely to happen (see Figure 3.4). Lastly, the landscape of the objective space and the noise therein will further determine the amount of overlapping between probability density functions of adjacent solutions in objective space. Regarding the causes of disorientation, the environment is responsible for over 95% of the cases of disorientation in PSO while it ranges between 83–93% in PSO-ER increasing with the level of noise. Disorientation by the environment is vastly more common than disorien-
Figure 3.2: Conditions leading to blindness. Figures 3.2a and 3.2b show conditions of regular operation and disorientation that will blind a particle at iteration $t + 1$. Figures 3.2c and 3.2d illustrate the causes of blindness for a particle at iteration $t$. The horizontal axis represents the objective values, and the vertical axis represents the probability density function of sampling noise. Gray areas cover the range of objective values to which a particle is (or will be) blinded. Solid and dashed lines represent the objective values of the current and personal best solutions, respectively. Vertical lines assume the true objective value of the solution is estimated correctly, e.g. $f(y^{t-1}) = \tilde{f}(y^{t-1})$. 
tation by memory because the direction of the optimization problem is backwards, and hence, when particles mistakenly update to worse solutions, their probability density functions will generally have larger standard deviations than their once personal bests.

Figure 3.3: Average number of disoriented particles in PSO-ER throughout the iterations on $F_{01}$ subject to different levels of noise. The horizontal axis represents the iterations, and the vertical axis represents the average number of disoriented particles from all independent runs.

3.4.3 Local and Global Certainty without Resampling

The quality of results shows that PSO-LC generally finds better solutions than PSO-GC, and both of them always find better solutions than PSO (except when $\sigma = 0.0$). On the one hand, particles in PSO-LC do not suffer from blindness or disorientation, hence allowing them to correctly improve upon their personal best solutions even when they are very likely to suffer from deception due to their neighbors estimates. On the other hand,
Figure 3.4: Conditions leading to disorientation subject to $f(x^t) < \tilde{f}(y^{t-1})$. Figures (a) and (b) show the necessary conditions for a particle to become disoriented at iteration $t$ under hypothetical cases of pre-existing blindness. The horizontal axis represents the objective values, and the vertical axis represents the probability of sampling noise. Light gray areas cover the range of objective values to which a particle is (or will be) blinded. Dark gray areas cover the range of objective values to which a particle will become disoriented. Solid and dashed lines represent the objective values of the current and personal best solutions, respectively. The vertical line in Figure (a) assumes that the true objective value of the personal best solution is estimated correctly, i.e., $f(y^{t-1}) = \tilde{f}(y^{t-1})$. 
particles in PSO-GC suffer from blindness and disorientation, which takes a toll on the quality of the personal best solutions from which the true neighborhood best solutions are selected. These results show that reducing blindness and disorientation (PSO-LC) is more important than reducing deception (PSO-GC). Nonetheless, reducing either of these conditions will significantly improve the quality of the results. The advantages and disadvantages of local and global certainty are reflected in the remaining statistics as follows.

The *lifetime* of PSO-LC is longer than PSO-GC because particles do not suffer from blindness and hence they keep improving upon their personal best solutions for more iterations, whereas blinded particles in PSO-GC eventually drive the swarm towards stagnation. The lifetime of PSO-GC is longer than that of PSO because selecting the true neighborhood best solution also provides more chances for particles to find better solutions and hence update their personal best solutions for more iterations.

The proportions of *deception* are similar between PSO and PSO-LC because particles in both algorithms select the neighborhood best solutions based on the noisy objective values. Hence, it is very unlikely that particles will correctly select the neighborhoods best solutions. Differently, particles in PSO-GC do not suffer from deception because global certainty is assumed and hence particles select the neighborhood best solutions based on their true objective values.

The proportions of *disorientation* are negligible in both PSO and PSO-GC for the same reasons explained in Section 3.4.2. Regarding the *causes of disorientation*, these are rather similar as expected because particles from both algorithms do not have local certainty and hence are subject to the same conditions that could lead to disorientation. The only difference in PSO is that particles additionally suffer from deception, whereas those in PSO-GC do not. Meanwhile, particles from PSO-LC do not suffer from disorientation because local certainty is assumed and hence particles correctly decide whether to update their personal best solutions or discard
their current solutions.

The proportions of regular operations are larger in PSO than in PSO-GC, while those of blindness are smaller. At best, we expected the opposite because particles in PSO-GC are partially attracted to better solutions than those in PSO thanks to global certainty, but we could have also understood similar results considering that both algorithms are under the same conditions that lead to blindness and disorientation. However, neither of our expectations was met and we would need a more detailed study to properly explain the underlying reasons to these statistics. Nonetheless, breaking them down into their respective regular updates and discards and causes of blindness, we find the following proportions that we did expect. Regular updates are more common in PSO-GC than in PSO because the former lacks deception, thereby increasing the chances to finding better solutions and hence updating more often. Blindness by memory is the major cause of blindness in both algorithms because, in backward optimization problems, the current solutions to which particles are blinded will be less corrupted by noise and hence more accurate.

3.4.4 Local and Global Certainty upon Resampling

The quality of results shows that PSO-ERGC finds significantly better solutions than PSO-ERLC, and both of them find better solutions than PSO-ER. On the one hand, the quality of their results under the assumptions of local and global certainty are the converse of those without resampling, thus suggesting that it is more important to focus on reducing deception only after blindness and disorientation have been reduced to a certain extent, but not before. This is not unexpected because disorientation leads to worse solutions and blindness leads to early stagnation, therefore, under such circumstances, the correct selection of the neighborhood best solutions will not lead to major improvements anyway. However, having reduced blindness and disorientation, particles will not only find better
solutions, but these will be further improved thanks to the extended lifetime of the swarm and to particles being partially attracted to the true neighborhood best solutions. On the other hand, the worse quality of results provided by PSO-ER is due to the presence of deception, blindness and disorientation that already deteriorate the results of PSO-ERLC and PSO-ERGC, only in this case these conditions are all present.

The remaining statistics analogously follow similar trends as PSO without resampling under local and global certainty, and the underlying reasons are the same as discussed earlier. The only difference is that particles in PSO-ERLC and PSO-ERGC estimate better the objective values of their solutions at the cost of performing just 100 iterations compared to the 600 performed without resampling. Consequently, the population statistics upon resampling contrast to those without resampling by showing an overall longer lifetime, smaller proportions of deception, larger proportions of regular operations, smaller proportions of blindness and (hence) larger proportions of disorientation, more regular updates, less blindness by memory, and less disorientation by the environment.

### 3.4.5 Further Discussions

The discussion presented thus far corresponds to the results obtained on benchmark function $F_{01}$, but these can be generalized to most of the trends in the remaining functions. Still, we find some exceptions on $F_{03}$, $F_{06}$, $F_{11}$ and $F_{16}$, in which increasing the levels of noise does not affect the quality of results, lifetime, or the proportions of blindness and disorientation present within the algorithms. We attribute these exceptions to the intrinsic challenges posed by the landscape of the ackley function from which these benchmarks are primarily composed of (see Table 2.2, page 52). However, we acknowledge that further studies are required to precisely identify the underlying characteristics of these problems to exhibit such exceptions.

The quality of results provided by the algorithms on most benchmark
functions are several orders of magnitude greater than the global optimum located in all of them at \( f(x) = 0 \). In our context, such a difference is no reason for concern because none of the algorithms have been designed to exploit particular traits of the benchmarks such as their large dimensionality or their composition. Instead, the algorithms are rather minimalistic in order to reduce the number of variables interacting during the search, and thereby allow to describe more accurately the role of key components on the population statistics. Moreover, we are not discussing the relevance of PSO to specifically address any of these problems, but rather the challenges to PSO that exist therein and in any other optimization problem subject to noise. Hence, our focus is on the relative performance between the algorithms and not on whether they find the global optimum.

The summary of statistical significance tests in Table 3.2 on the quality of the results between the algorithms further supports the generality of our discussions to the benchmark functions addressed. Specifically, PSO-ER generally finds better solutions than the regular PSO (or at least not worse), and the quality of both their solutions can only be improved by making correct decisions on either the personal or neighborhood best solutions.

The average population statistics for PSO and PSO-ER across all benchmark functions in the presence of noise are presented in Table 3.3. These statistics show similar trends to the population statistics on \( F_{01} \) discussed in the previous sections.

- The *lifetime* of PSO-ER (63.97%) is longer than that of PSO (58.58%) due its smaller proportions of blindness and smaller number of iterations.

- The *deception* suffered by particles in PSO-ER (92.74%) is smaller than that in PSO (96.99%) thanks to having the objective values of the solutions more accurately estimated, although both proportions are still very high.
3.4. RESULTS AND DISCUSSIONS

Table 3.2: Summary of statistical significance tests on the quality of results between relevant pairs of algorithms. The tables count the number of functions at which the quality of results from a PSO algorithm is similar (‘=’) or significantly better (‘−’) or worse (‘+’) than its PSO counterpart at each level of noise $\sigma$ according to the pairwise Wilcoxon test at $\alpha = 0.05$ with Holm [67] correction.

<table>
<thead>
<tr>
<th></th>
<th>ER vs. PSO</th>
<th>ERGC vs. ERLC</th>
<th>ERLC vs. ER</th>
<th>LC vs. GC</th>
<th>GC vs. PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>− = +</td>
<td>− = +</td>
<td>− = +</td>
<td>− = +</td>
<td>− = +</td>
</tr>
<tr>
<td>$\sigma_{06}$</td>
<td>10 10 0</td>
<td>13 7 0</td>
<td>16 4 0</td>
<td>4 8 8</td>
<td>19 1 0</td>
</tr>
<tr>
<td>$\sigma_{12}$</td>
<td>12 8 0</td>
<td>16 4 0</td>
<td>18 2 0</td>
<td>10 6 4</td>
<td>19 1 0</td>
</tr>
<tr>
<td>$\sigma_{18}$</td>
<td>14 6 0</td>
<td>16 4 0</td>
<td>19 1 0</td>
<td>11 4 5</td>
<td>19 1 0</td>
</tr>
<tr>
<td>$\sigma_{24}$</td>
<td>15 5 0</td>
<td>15 5 0</td>
<td>19 1 0</td>
<td>10 5 5</td>
<td>20 0 0</td>
</tr>
<tr>
<td>$\sigma_{30}$</td>
<td>15 5 0</td>
<td>16 4 0</td>
<td>20 0 0</td>
<td>7 8 5</td>
<td>20 0 0</td>
</tr>
<tr>
<td>Total</td>
<td>66 34 0</td>
<td>76 24 0</td>
<td>92 8 0</td>
<td>42 31 27</td>
<td>98 2 0</td>
</tr>
</tbody>
</table>
• The proportions of regular operation in PSO-ER (61.67%) are larger than those in PSO (46.51%) thanks to the greater accuracy of the objective values of its solutions. From these proportions, only 3.69% of the decisions in PSO lead particles to find better solutions while the remaining 96.31% of solutions are discarded. Differently, 16.06% of the decisions in PSO-ER lead to better solutions while the remaining 83.94% of the solutions are discarded.

• The proportions of blindness in PSO-ER (36.13%) are smaller than those in PSO (53.00%) thanks to the greater accuracy of the objective values of its solutions. From these proportions, over 90% are blindness caused by memory while the remaining are caused by the environment.

• The proportions of disorientation in PSO (0.49%) are smaller than those in PSO-ER (2.20%) because larger proportions of blindness render disorientation less likely to occur. From these proportions, approximately 90% or more are caused by the environment while the remaining are caused by memory.

The population statistics help to characterize the performance of the PSO algorithms on optimization problems subject to noise, and provide important insights for potential improvements regardless of the problem. For example, the proportions of blindness are largely caused by inaccurate objective values in the personal best solutions (i.e. blindness by memory), hence suggesting that algorithms which better estimate these solutions or deal with such inaccuracies will be able to reduce blindness and therefore potentially improve the quality of the results. As such, these population statistics can be utilized to measure the quality of other noise mitigation mechanisms for PSO such as evaporation [30, 31, 32, 50, 51] and other resampling methods [7, 106, 113, 154]. More importantly, the population statistics provide a guidance for tuning the PSO algorithms before addressing noisy real-world optimization problems, where it is not possible
### 3.5 Summary

Table 3.3: Average population statistics for PSO with and without resampling over all the benchmark functions, levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>PSO</th>
<th>PSO-ER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min. mean max.</td>
<td>min. mean max.</td>
</tr>
<tr>
<td>Lifetime(^+)</td>
<td>48.99 58.58 84.05</td>
<td>48.72 63.97 90.08</td>
</tr>
<tr>
<td>Deception(^-)</td>
<td>89.19 96.99 99.86</td>
<td>51.98 92.74 100.00</td>
</tr>
<tr>
<td>Regular Operation(^+)</td>
<td>32.79 46.51 79.47</td>
<td>46.20 61.67 94.15</td>
</tr>
<tr>
<td>Updates(^+)</td>
<td>1.87 3.69 12.03</td>
<td>6.82 16.06 43.21</td>
</tr>
<tr>
<td>Discards(^-)</td>
<td>87.97 96.31 98.13</td>
<td>56.79 83.94 93.18</td>
</tr>
<tr>
<td>Blindness(^-)</td>
<td>19.78 53.00 66.83</td>
<td>4.14 36.13 52.27</td>
</tr>
<tr>
<td>Memory</td>
<td>96.42 98.96 99.49</td>
<td>67.78 91.85 96.37</td>
</tr>
<tr>
<td>Environment</td>
<td>0.51 1.04 3.58</td>
<td>3.63 8.15 32.22</td>
</tr>
<tr>
<td>Disorientation(^-)</td>
<td>0.37 0.49 1.45</td>
<td>1.53 2.20 4.26</td>
</tr>
<tr>
<td>Memory</td>
<td>1.85 4.48 16.47</td>
<td>2.46 10.53 33.07</td>
</tr>
<tr>
<td>Environment</td>
<td>83.53 95.52 98.15</td>
<td>66.93 89.47 97.54</td>
</tr>
</tbody>
</table>

Larger\(^+\) or smaller\(^-\) is better. Units in percentages (%).

to know the true objective values of the solutions and hence the population statistics of deception, blindness and disorientation cannot be computed.

Particle Swarm Optimization is a population-based algorithm whose quality of results severely deteriorates in optimization problems subject to noise. In this type of problem, particles are often not able to correctly distinguish the quality of the solutions, and that leads them to suffer from three conditions which we have identified as deception, blindness and disorientation. Particles suffering from deception are partially attracted towards sub-optimal regions of the search space, those suffering from blindness
miss out on opportunities to improve upon their personal best solutions, and those suffering from disorientation mistakenly prefer worse solutions. Thus, under these conditions, the quality of the results obtained with particle swarms are clearly bound to deteriorate. Therefore, it is important to explicitly address these conditions with noise mitigation mechanisms, but first it is necessary to understand the extents to which deception, blindness and disorientation affect the particles. To this end, we developed the population statistics for PSO.

Population statistics is a term that we coin to refer to a set of indicators which measure, from a particle perspective, different attributes of the swarm throughout the search process. As such, the population statistics provide valuable information that complements, and explains to a certain extent, the quality of the results obtained. The population statistics utilized for the algorithms under study are composed of 12 indicators which measure not only the proportions of the particles affected by deception, blindness and disorientation, but also their causes and other characteristics such as the lifetime of the swarm and its regular operations. These indicators helped to explain the deterioration of PSO in the presence of noise, remark the importance of incorporating resampling methods, and assign priorities over addressing the conditions responsible for deteriorating the quality of the results.

The population statistics for PSO revealed that the quality of its results deteriorates in the presence of noise mostly due to the large proportions of blindness and deception affecting the swarm. Blindness not only prevents particles from improving their personal best solutions, but together with deception, both reduce the lifetime of the swarm and have the potential to even drive the swarm to stagnation. Another important factor to such a deterioration is the proportions of regular updates which reduce as the level of noise increases. The number of regular updates refers to particles improving their personal best solutions, and is the key to improving the quality of the results. However, the proportions of regular updates
will automatically increase by reducing blindness in the swarm. Disorientation, on the other hand, had a negligible presence in the swarms and therefore we find no interest in addressing it yet. Nonetheless, disorientation is still a condition that will deteriorate the quality of the results and therefore needs to be monitored.

The population statistics also remarked the sensitivity of PSO to optimization problems subject to noise given the better quality of the results obtained with PSO-ER. In this case, resampling was incorporated into PSO to more accurately estimate the objective values of its solutions at the cost of performing fewer iterations. In spite of such a tradeoff, PSO-ER found significantly better solutions than PSO. Furthermore, considering that particles in PSO-ER still suffer from deception, blindness and disorientation, we expect that further addressing these conditions will lead to even better results.

The population statistics for the swarms under the assumptions of local and global certainty showed that reducing blindness and disorientation has a higher priority than reducing deception. First, reducing blindness and disorientation will improve the quality of the personal best solutions from which the neighborhood best solutions will be selected. Second, having the objective values of the personal best solutions more accurately estimated will help to further reduce the proportions of deception. Lastly, reducing blindness will also extend the lifetime of the swarm and hence will also increase the chances of finding better solutions.

The population statistics for the algorithms presented in this chapter comprise the first of a series of studies which we will pursue in the following chapters to further understand the effect of noise on PSO. More importantly, we expect that the population statistics will help towards the design of more effective noise mitigation mechanisms for PSO.
Next Chapter

The next chapter will present a study on the population statistics for single-evaluation PSO algorithms on optimization problems subject to noise.

3.A Population Statistics

The following figures correspond to the population statistics on each benchmark function for the algorithms utilized in the design of experiments from this chapter (Section 3.3, page 70).
(This page is intentionally left blank)
Figure 3.5: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The algorithms are abbreviated as (a) PSO, (l) PSO-LC and (g) PSO-GC in the first subdivision, and those in the second subdivision are abbreviated as (e) PSO-ER, (l) PSO-ERLC and (g) PSO-ERGC. The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions. The boxplots corresponding to (a) in $F_{04}, F_{07}$ and $F_{08}$ at $\sigma \in \{0.12, 0.18, 0.24, 0.30\}$ have been downscaled to improve their presentation without changing their relative ordering.
Figure 3.6: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The algorithms are abbreviated as (a) PSO, (l) PSO-LC and (g) PSO-GC in the first subdivision, and those in the second subdivision are abbreviated as (e) PSO-ER, (l) PSO-ERLC and (g) PSO-ERGC. The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
Figure 3.7: Lifetime. The barplots represent the proportions of average lifetime (left axis) of a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (a) PSO, (al) PSO-LC, (ag) PSO-GC, (e) PSO-ER, (el) PSO-ERLC and (eg) PSO-ERGC. A longer lifetime is better when the swarm does not converge to the global optimum.
Figure 3.8: Deception. The barplots represent the average proportions of iterations (left axis) at which a particle is deceived by its neighbors for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (a) PSO, (al) PSO-LC, (ag) PSO-GC, (e) PSO-ER, (el) PSO-ERLC and (eg) PSO-ERGC. Particles in PSO-ERGC do not suffer from deception. Smaller proportions are better.
Figure 3.9: Regular Operation, Blindness and Disorientation. The stacked barplots represent the average proportions (left axis) of regular operation (dark gray), blindness (medium gray) and disorientation (light gray) experienced by a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (a) PSO, (al) PSO-LC, (ag) PSO-GC, (e) PSO-ER, (el) PSO-ERLC and (eg) PSO-ERGC. Larger proportions of regular operation and smaller proportions of blindness and disorientation are better.
3.A. POPULATION STATISTICS

Figure 3.10: Regular Updates and Discards. The stacked barplots represent the average proportions (left axis) of regular updates (dark gray) and discards (light gray) experienced by a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (a) PSO, (al) PSO-LC, (ag) PSO-GC, (e) PSO-ER, (el) PSO-ERLC and (eg) PSO-ERGC. Larger proportions of regular updates and smaller proportions of regular discards are better.
Figure 3.11: Causes of Blindness. The stacked barplots represent the average proportions (left axis) of blindness caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (a) PSO, (al) PSO-LC, (ag) PSO-GC, (e) PSO-ER, (el) PSO-ERLC and (eg) PSO-ERGC.
Figure 3.12: Causes of Disorientation. The stacked barplots represent the average proportions (left axis) of disorientation caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (a) PSO, (al) PSO-LC, (ag) PSO-GC, (e) PSO-ER, (el) PSO-ERLC and (eg) PSO-ERGC.
Chapter 4

Single-Evaluation Methods

This chapter studies the population statistics for single-evaluation PSO algorithms. The set of algorithms under study consists, on the one hand, of the PSO-E already proposed in the literature with evaporation factors estimated with a new method that we propose, and on the other hand, of two new algorithms proposed in this chapter that address differently the problem of inaccurate memories.

This chapter is structured as follows. Section 4.1 introduces this chapter. Section 4.2 presents the design of the new single-evaluation PSO algorithms and the method to estimate the evaporation factors for PSO-E. Section 4.3 describes the design of experiments. Section 4.4 presents the results and discussions. Finally, Section 4.5 ends this chapter with a summary.

4.1 Introduction

Single-evaluation methods are noise mitigation mechanisms for PSO whose operation is based on having the objective values of the solutions (inaccurately) estimated with a single evaluation. We refer to these methods as single-evaluation to contrast them from resampling methods, whose operation involves re-evaluating the solutions multiple times and estimat-
ing the objective values of the solutions by a sample mean over the evaluations. These two approaches present different tradeoffs with regards to the (fixed and limited) computational budget of function evaluations available to the PSO algorithm. On the one hand, single-evaluation PSO algorithms will be able to perform more iterations at the cost of dealing with solutions whose objective values are very inaccurately estimated. On the other hand, resampling-based PSO algorithms will be able to better estimate the true objective values of the solutions at the cost of performing fewer iterations.

The evaporation mechanism \[31\] is a single-evaluation method for PSO designed to reduce the presence of blindness in the swarm by addressing directly the problem of inaccurate memories. Specifically, PSO-E consists of worsening the estimated objective values of the personal best solutions by a constant known as the \textit{evaporation factor} whenever the particles are not able to find better solutions (see Section 2.6.2, page 35). As such, in minimization problems, PSO-E manages to correct the underestimated objective values of the personal best solutions up to a certain extent, thereby reducing blindness and preventing particles from reaching premature convergence. However, the evaporation mechanism will also overestimate already overestimated objective values, which will potentially lead the swarm to divergent behaviour. In either case, the operation of PSO-E depends on the value of the evaporation factor, which thus far has been determined empirically based on the quality of the results obtained on a few benchmark functions \[30, 31, 32, 50, 51\]. Still, previous works have reported that PSO-E finds better solutions than the regular PSO \[30, 31\]. Some researchers have utilized PSO-E to approach both stochastic and dynamic optimization problems \[32, 50\], and others have proposed different variants involving dynamic evaporation factors and heterogeneous swarms \[51, 125\]. However, these works often fall short of empirical evidence to support their claims about the underlying reasons for the obtained quality of results.
4.2 SINGLE-EVALUATION PSO ALGORITHMS

Chapter Goals

The overall goal of this chapter is to study and compare the population statistics for different PSO algorithms with single-evaluation methods on optimization problems subject to different levels of multiplicative Gaussian noise. Specifically, we will focus on the following objectives.

- Design a method to better determine the evaporation factors in PSO-E.
- Develop a new PSO algorithm where particles utilize probabilistic updates instead of evaporation factors in order to prevent the divergent behaviour of PSO-E.
- Develop a new PSO algorithm where particles compute the neighborhood best solutions as the centroid of the solutions within the neighborhoods in order to blur the effect of noise upon selection.

4.2 Single-Evaluation PSO algorithms

Single-evaluation methods in PSO are those noise mitigation mechanisms whose operation does not involve performing additional evaluations to any of the solutions in the swarm, thereby settling with having their objective values estimated by a single evaluation that is performed by the PSO algorithm. Consequently, single-evaluation methods need to assume that the objective values of the solutions are very inaccurate and the solutions cannot be re-evaluated multiple times to better estimate them as resampling methods do. Despite that resampling methods provide solutions with more accurately estimated objective values, the leverage of single-evaluation methods is that they search over more iterations, hence exploring more solutions and thereby increasing the chance of finding even better solutions.

The evaporation mechanism [31] is a single-evaluation method that has served as the basis for other studies [30, 32, 50, 51, 125]. In the literature,
CHAPTER 4. SINGLE-EVALUATION METHODS

the evaporation mechanism has been utilized to address both stochastic and dynamic optimization problems [30, 31, 32, 50], but the problems have been loosely referred to as dynamic optimization problems. While the evaporation mechanism is suitable for addressing both stochastic and dynamic optimization problems, it is not a requirement for noise mitigation mechanisms to address both types of problem. Hence, given the ambiguity introduced by enclosing stochastic problems as a type of dynamic optimization problems, we consider it appropriate to make a clear distinction between them such that the targets of noise mitigation mechanisms can be precisely identified.

The remainder of this section is structured as follows. Section 4.2.1 distinguishes between stochastic and dynamic optimization problems. Section 4.2.2 presents our method to determine the evaporation factors of PSO-E. Section 4.2.3 presents a new single-evaluation method based on probabilistic updates. Finally, Section 4.2.4 presents another single-evaluation method based on averaging the neighborhood best solutions.

4.2.1 Stochastic vs Dynamic Optimization Problems

Stochastic and dynamic optimization problems have in common that the objective space is subject to change due to external factors, and such a change can occur directly on the objective space or indirectly via the solution space. However, it is the model of the external factors that determine whether the optimization problem is stochastic, dynamic, or even both. In stochastic optimization problems, the external factors are modeled as sampling noise from a probability distribution. As such, despite that the true objective space of the problem will never change, multiple evaluations to any solution will yield differently estimated objective values due to the effect of sampling noise. In dynamic optimization problems, however, the external factors are instead deterministic functions that change over time. As such, at instant \( t \), any number of evaluations to the same solution will
always yield its true objective value, but it may no longer be the same objective value at $t + 1$.

The characteristics of both, stochastic and dynamic optimization problems, can be combined to derive even more challenging problems, but have in mind that these problems are already inherently complex on their own and each will probably require to be addressed with different strategies. For example, in stochastic optimization problems where the objective space is directly affected by noise, the goal is to better estimate the true objective values of the solutions and/or minimize the effect of their intrinsic inaccuracies. Differently, in dynamic optimization problems where changes directly occur on the objective space, the goal is instead to promptly identify these changes and react accordingly. Therefore, it is necessary to thoroughly consider the characteristics of the problem before deciding what strategy to use.

4.2.2 Evaporation Factor

The evaporation factor of PSO-E provides control over the exploration and exploitation of the search space as follows. Particles with low evaporation factors will slowly worsen the estimated objective values of their personal best solutions, hence favouring exploitation because these solutions will be more likely to remain in memory for more iterations. Conversely, high evaporation factors will quickly worsen the estimated objective values of the personal best solutions, hence favouring exploration because these solutions will be more likely to be replaced with the current solutions.

Previous works have suggested to use different values as evaporation factors \cite{30, 31, 32, 50}, but these values have always been derived empirically and the effect over the estimated objective values has only been clear for the immediately next unsuccessful iteration. Hence, in order to provide a long-term understanding of the effect of the evaporation factors, we propose to determine the evaporation factor on minimization prob-
lems according to the number of unsuccessful iterations a particle requires to double the estimated objective value of its personal best solution. The assumption is that the personal best solutions, before doubling their estimated objective values, will most likely be replaced with the current solutions. Thus, this doubling time provides some guidance on the tolerance that particles will have towards unsuccessful iterations and hence on the replacement of the personal best solutions. The method that we propose to determine the evaporation factors with respect to the doubling time is described as follows.

In minimization problems, the evaporation mechanism will cause an exponential growth of the estimated objective value of the personal best solution for any particle \( i \) over \( u \) unsuccessful iterations after \( t \). Such an exponential growth is depicted in Equation (4.1).

\[
\hat{f}(y_i^t) = \hat{f}(y_i^{t-u})(1 + \rho)^u
\]  

Therefore, given an evaporation factor \( \rho \), the required number of unsuccessful iterations \( u \) to double any objective value is determined by (4.2), from which we obtain (4.3) by isolating \( \rho \) in order to determine instead the evaporation factor required to double any objective value given \( u \) unsuccessful iterations. Thus, using (4.3), Table 4.1 presents the evaporation factors for different numbers of unsuccessful iterations.

\[
u = \frac{\ln(2)}{\ln(1 + \rho)}
\]

\[
\rho = \exp(\ln(2)/u) - 1 = 2^{1/u} - 1
\]

4.2.3 PSO with Probabilistic Updates

Probabilistic Updates is a single-evaluation method that we propose as an alternative to the evaporation mechanism. In PSO with Probabilistic Updates (PSO-PU), a particle decides probabilistically to update its personal
Table 4.1: Evaporation factors after $u$ unsuccessful iterations.

<table>
<thead>
<tr>
<th>$u$</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho$</td>
<td>1.0000</td>
<td>0.4142</td>
<td>0.1487</td>
<td>0.0718</td>
<td>0.0353</td>
<td>0.0140</td>
<td>0.0070</td>
<td>0.0035</td>
<td>0.0014</td>
</tr>
</tbody>
</table>

best solution whenever the current solution is not better. Thus, particles have an update probability $\theta$ that is analogous to the evaporation factor $\rho$ because both specify an approximate threshold to replace the personal best solutions after a number of unsuccessful iterations. However, there are three important differences between PSO-PU and PSO-E. First, the probability $\theta$ provides a more reliable criterion to replace the personal best solutions after $u$ unsuccessful iterations given the known uniform probability distribution, whereas the evaporation factor relies on the assumption that, by doubling the estimated objective values of the personal best solutions, these will likely be updated. Second, PSO-PU provides more exploitation than PSO-E because the estimated objective values remain unchanged, whereas the evaporation mechanism constantly worsens them hence lowering the requirements to find replacements. Third, the memory update in PSO-PU does not depend on the objective space because the update probabilities are independent, whereas such is not the case in PSO-E because the evaporation factors proportionally worsen the estimated objective values. The PSO-PU algorithm is described in Figure 4.1, where $r_i^t$ is a random value sampled from a uniform distribution $U(0,1)$, and $\theta$ determines the probability for particles to update their personal best solutions.

4.2.4 PSO with Average Neighborhoods

Average Neighborhoods is a single-evaluation method that we propose to reduce the effect of deception in the swarm, which affects the particles of the regular PSO in $96.99\%$ of the iterations (see Table 3.3, page 89). The
for each iteration \( t \) do
  for each particle \( i \) in swarm do
    if \( \hat{f}(x^t_i) < \hat{f}(y^t_{i-1}) \) or \( r^t_i < \theta \)
    then \( y^t_i \leftarrow x^t_i \) else \( y^t_i \leftarrow y^t_{i-1} \)
  for each particle \( i \) in swarm do
    \( \hat{y}^t_i \leftarrow y^t_{\omega} \) where \( \omega = \arg \min_{j \in N_i} \hat{f}(y^t_j) \)
  for each particle \( i \) in swarm do
    update \( v^t+1_i \) and \( x^t+1_i \) using (2.3) and (2.4)

Figure 4.1: PSO with Probabilistic Updates.

PSO with Average Neighborhoods (PSO-AN) is presented in Figure 4.2, where particles utilize as neighborhood best solutions the centroid of the personal best solutions of the particles within their neighborhoods. As such, the goal is to create a new solution that will be potentially better than any other solution selected based on its very inaccurately estimated objective value.

The PSO-AN is a special case of the Fully-Informed Particle Swarm (FIPS) [88] because the velocity equation of the particles is influenced by every other particle within the neighborhood. However, both algorithms differ with respect to the influence that particles within the neighborhood have. In FIPS, the neighborhood best solution is computed as the weighted centroid of the personal best solutions within the neighborhood, where the weight for each solution is just a random value that equally affects all the dimensions of the solution. Differently, in PSO-AN we want a particle to be partially attracted just to the centroid solution of the neighborhood, that is, without the effect of any random value. As such, PSO-AN can be seen as a special case of FIPS when every weight utilized to compute the centroid solution is equal to 1.0.
4.3 Design of Experiments

The algorithms to compare are the regular PSO, PSO-E, PSO-PU and PSO-AN on optimization problems subject to different levels of multiplicative Gaussian noise. Specifically, we compare the PSO-E and PSO-PU with different evaporation factors and update probabilities computed in order to have particles replace their personal best solutions after approximately $u \in \{50, 25, 10, 5, 2\}$ unsuccessful iterations. As such, the evaporation factors for $u$ are computed according to Equation (4.3) resulting in $\rho \in \{0.014, 0.028, 0.071, 0.149, 0.414\}$. Differently, we compute the update probabilities in PSO-PU as $\theta = 1/u$ resulting in $\theta \in \{0.02, 0.04, 0.1, 0.2, 0.5\}$.

The swarms for each algorithm are made up of 50 particles whose inertia and acceleration coefficients are set according to the guidelines in [145], and their neighborhoods are defined by the star topology. Particles limit their velocities utilizing the hyperbolic tangent function to reduce the sensitivity of setting a maximum velocity [48], which is set according to the limits $x_{\text{min}}$ and $x_{\text{max}}$ of the optimization problem. The computational budget available to each swarm is set to 30,000 function evaluations, thereby allowing the swarms to iterate for 600 iterations. In general, the settings are the same as those in Chapter 3 but with a different set of algorithms.
The complete list of parameter settings is presented in Table 4.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent runs</td>
<td>50 with 600 iterations</td>
</tr>
<tr>
<td>Number of particles</td>
<td>50 in $\mathbb{R}^{1000}$ with star topology</td>
</tr>
<tr>
<td>Acceleration</td>
<td>Static with $c_1 = c_2 = 1.49618$</td>
</tr>
<tr>
<td>Inertia</td>
<td>Static with $w = 0.729844$</td>
</tr>
<tr>
<td>Maximum velocity</td>
<td>$0.25 \cdot</td>
</tr>
<tr>
<td>Velocity clamping</td>
<td>$v^t_{ij} = v_{\text{max}} \cdot \tanh \left( \frac{v^t_{ij}}{v_{\text{max}}} \right)$</td>
</tr>
<tr>
<td>Severity of noise</td>
<td>$\sigma \in {0.00, 0.06, 0.12, 0.18, 0.24, 0.30}$</td>
</tr>
<tr>
<td>Unsuccessful iterations</td>
<td>$u \in {50, 25, 10, 5, 2}$</td>
</tr>
<tr>
<td>Evaporation factors</td>
<td>$\rho \in {0.014, 0.028, 0.071, 0.149, 0.414}$</td>
</tr>
<tr>
<td>Update probability</td>
<td>$\theta \in {0.02, 0.04, 0.1, 0.2, 0.5}$</td>
</tr>
</tbody>
</table>

The population statistics for each swarm are computed utilizing the information from all 50 particles throughout the 600 iterations that take place in each independent run from of a total of 50. As such, the population statistics are made up of $50 \times 600 \times 50 = 1,500,000$ observations as follows. The quality of results represents the objective values of the best solutions found by the swarm. The proportions of blindness and disorientation measure the extents to which these conditions affect the particles in the swarm. The proportions of blindness by memory and blindness by the environment identify the causes of blindness in the swarm. The proportions of disorientation by memory and disorientation by the environment identify the causes of disorientation in the swarm. The proportions of regular operations measure how often particles operate correctly, that is, how often particles replace the personal best solutions for truly better solutions and how often particles discard truly worse current solutions, both of which are
measured independently as the proportions of regular updates and regular discards, respectively. The population statistics are detailed in Section 3.2 (page 65).

In addition, we design the following four population statistics. The optimization curves (4.4) compute the average objective values of the personal best solutions from the particles in the swarm at each iteration, thereby creating an optimization curve when considering all the iterations. The deterioration caused by disorientation (4.5) computes the average magnitude of deterioration of the objective values when replacing the personal best solutions with worse solutions. The hypothetical improvement missed by blindness (4.6) computes the average magnitude of the improvement to the objective values had the personal best solutions correctly been replaced by the current solutions. The ranked deception (4.7), different from the hereinafter binary deception (3.1), computes the average rank of the neighborhood best solutions with respect to the personal best solutions in the swarm, for which a better ranked deception is equivalent to a better quality of the neighborhood best solutions.

\[
I_C(S) = \frac{1}{|S|} \sum_{i \in S} f(y_i^t)
\]

\[
I_{E_d}(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} D_i^t \left[ f(y_i^t) - f(y_i^{t-1}) \right] \sum_{t=1}^{t_{\text{max}}} D_i^t
\]

\[
I_{E_b}(S) = \frac{1}{|S|} \sum_{i \in S} \sum_{t=1}^{t_{\text{max}}} B_i^t \left[ f(y_i^t) - f(x_i^t) \right] \sum_{t=1}^{t_{\text{max}}} B_i^t
\]

\[
I_{D_r}(S) = \frac{1}{|S|} \sum_{i \in S} \frac{1}{t_{\text{max}}} \sum_{t=1}^{t_{\text{max}}} \text{rank}(f(\hat{y}_i^t), S)
\]

4.4 Results and Discussions

The different evaporation factors and update probabilities produced a large number of results with trends showing that, in most population statis-
tics, the performance of PSO-E and PSO-PU is better for larger values of $u$ (i.e. smaller evaporation factors and update probabilities, respectively). Hence, for the sake of conciseness, we have selected for discussion only those results corresponding to $u \in \{25, 5\}$, which still comprise the trends observed. Hereinafter, we refer to PSO-E with $\rho \in \{0.028, 0.149\}$ as PSO with Low Evaporation (PSO-LE) and PSO with High Evaporation (PSO-HE), respectively. Likewise, we refer to PSO-PU with $\theta \in \{0.04, 0.2\}$ as PSO with a Low Probability to Update (PSO-LP) and PSO with a High Probability to Update (PSO-HP).

Our discussions in this section are based on the population statistics for PSO-LE, PSO-HE, PSO-LP, PSO-HP, regular PSO, and PSO-AN on benchmark function $F_{01}$. These population statistics reflect most of the trends found in the other benchmark functions, all of which are presented in Appendix 4.A (page 132). This section is structured as follows. Section 4.4.1 presents the population statistics for the algorithms on benchmark function $F_{01}$. Section 4.4.2 contrasts the population statistics of PSO-LE and PSO-HE. Section 4.4.3 compares PSO-LP and PSO-HP and contrasts their population statistics against the evaporation counterparts. Section 4.4.4 compares the regular PSO and PSO-AN. Lastly, Section 4.4.5 presents further discussions about the algorithms and their overall performance across the benchmark functions.

### 4.4.1 Population Statistics

The population statistics for $F_{01}$ are presented in Figure 4.3, where the algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, and (n) PSO-AN. The statistics are presented as follows.

*Quality of Results.* The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) on $F_{01}$ subject to the different levels of noise (top axis). The boxplots are coloured from dark to light gray to ease the comparison between the algorithms.
4.4. RESULTS AND DISCUSSIONS

The benchmark functions are minimization problems and therefore lower objective values indicate better solutions.

**Binary Deception.** The barplots represent the average proportion of iterations (left axis) in which a particle for each algorithm (bottom axis) is deceived by its neighbors on $F_{01}$ subject to the different levels of noise (bars coloured from light to dark gray). The centroid solution of PSO-AN is evaluated only to compute this statistic. Smaller proportions are better.

**Ranked Deception.** The barplots represent the average normalized rank (left axis) of the neighborhood best solution chosen by a particle for each algorithm (bottom axis) on $F_{01}$ subject to the different levels of noise (bars coloured from light to dark gray). The ranks are normalized by computing the ratio between the rank of the solution with respect to the swarm and the number of particles in the swarm. The centroid solution of PSO-AN is evaluated only to compute this statistic. Smaller percentile ranks are better.

**Regular Operation, Blindness and Disorientation.** The stacked barplots represent the average proportions (left axis) of regular operation (dark gray), blindness (medium gray) and disorientation (light gray) experienced by a particle for each algorithm (bottom axis) on $F_{01}$ subject to the different levels of noise (bars from left to right). Larger proportions of regular operations and smaller proportions of blindness and disorientation are better.

**Regular Updates and Discards.** The stacked barplots represent the average proportions (left axis) of regular updates (dark gray) and discards (light gray) experienced by a particle for each algorithm (bottom axis) on $F_{01}$ subject to the different levels of noise (bars from left to right). Larger proportions of regular updates and smaller proportions of discards are better.

**Causes of Blindness.** The stacked barplots represent the average proportions (left axis) of blindness caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on
$F_{01}$ subject to the different levels of noise (bars from left to right). Particles do not suffer from blindness in the absence of noise.

*Causes of Disorientation.* The stacked barplots represent the average proportions (left axis) of disorientation caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on $F_{01}$ subject to the different levels of noise (bars from left to right). Particles from the regular PSO and PSO-AN do not suffer from disorientation in the absence of noise.

*Effect of Disorientation and Blindness.* The stacked barplots represent the normalized average magnitudes (left axis) of deterioration caused by disorientation (dark gray) and hypothetical improvement missed by blindness (light gray) on $F_{01}$ subject to the different levels of noise (bars from left to right). Particles from the regular PSO and PSO-AN do not suffer from blindness or disorientation in the absence of noise.

*Optimization Curves.* The plots represent the average objective values (left axis) of the personal best solutions over all the independent runs at each iteration (bottom axis) of PSO-LE (marked with ‘•’) and PSO-LP (no marks) on $F_{01}$ subject to levels of noise $\sigma_{00}$ (solid line), $\sigma_{06}$ (long-dashed line), $\sigma_{18}$ (short-dashed line), and $\sigma_{30}$ (dotted line).

The population statistics are discussed in the following sections and are referred to by their names in *italics* whenever they are brought up for discussion.

### 4.4.2 Low and High Evaporation Factors

In the absence of noise, the *quality of results* shows that PSO-LE and PSO-HE find solutions of similar quality, which is supported by the similarities in the remaining population statistics. The absence of noise will never cause particles from either algorithm to suffer from *blindness* because the objective values of their personal best solutions will always be estimated correctly. However, due to the evaporation mechanism, particles that do not
Figure 4.3: Population statistics on benchmark function $F_{01}$. The presentation details of each statistic are described in Section 4.4.1 (page 114).
find better solutions at any given iteration will overestimate the objective values of their personal best solutions by a factor of $\rho$. This overestimation will eventually cause particles to suffer from disorientation, whose presence is in about 10% of the iterations on average. Furthermore, since the objective values of the current solutions will always be estimated correctly and those of the personal best solutions will be overestimated, the only cause of disorientation will always be disorientation by memory. Despite that the population statistics are similar for both algorithms, we expect that, in the absence of noise, the quality of the results of PSO-LE will generally be better than PSO-HE because PSO-LE has better exploitation capabilities of the solution search space.

In the presence of noise, the quality of results shows that PSO-LE finds significantly better solutions than PSO-HE regardless of the level of noise. The main reason for such a quality of results is that the evaporation factor of PSO-HE is too high and hence provides too much exploration by having its particles replace their personal best solutions too often. However, such an operation of PSO-HE shows some improvements upon its own results at high levels of noise, but the quality of its results is still not better than that of PSO-LE. The differences between PSO-LE and PSO-HE in terms of exploitation and exploration are evidenced in their respective proportions of blindness and disorientation. Particles in PSO-LE suffer more from blindness because its low evaporation factor worsens the estimated objective values of their personal best solutions at a slower rate, hence retaining the solutions for more unsuccessful iterations. Conversely, particles in PSO-HE suffer more from disorientation because its particles worsen the objective values of their personal best solutions much faster, hence relaxing the threshold required to replace them and therefore making easier to take worse solutions after a few unsuccessful iterations. Moreover, such a quality of results can also be expected since the deterioration caused by disorientation in PSO-HE is larger than any hypothetical improvement missed by blindness (except at a very high level of noise), but more importantly, the
deterioration is much larger for PSO-HE than it is for PSO-LE.

The proportions of *blindness by memory* in PSO-LE are larger than *blindness by the environment* because its low evaporation factor will retain the personal best solutions in memory for more unsuccessful iterations, and hence particles will remain blinded to a slowly reducing range of solutions. Moreover, *blindness by memory* increases with higher levels of noise because the objective values of the personal best solutions will suffer larger underestimations which will require even more unsuccessful iterations to eventually find a replacement. Furthermore, since the direction of the optimization problem is backwards, the better solutions to which particles are blinded will be less corrupted by noise and hence will be better estimated than their personal best solutions, thereby increasing blindness by memory. Regarding the causes of blindness in PSO-HE, the proportions of *blindness by memory* and *blindness by the environment* are similar (around 50% each) at levels of noise ranging from low to very high, thus further supporting that $\rho = 0.149$ is already too large of a value given that particles are more or less arbitrarily replacing their personal best solutions after a few unsuccessful iterations.

The proportions of *disorientation by memory* in PSO-HE are significantly larger than those in PSO-LE because a high evaporation factor will cause larger differences between the true and the (eventually) overestimated objective values of the personal best solutions. Still, *disorientation by the environment* is the leading cause of disorientation for both evaporation factors, thus showing that the evaporation mechanism indeed manages to mitigate the effect of noise on the objective values of the personal best solutions up to a certain extent. Furthermore, since the direction of the optimization problem is backwards, the worse (current) solutions to which particles become disoriented will be more corrupted by noise and hence their objective values will be less accurately estimated than their previous personal best solutions, thus causing disorientation by the environment to increase.

The proportions of *regular operations* between PSO-LE and PSO-HE are
similar, but are composed differently as PSO-HE has larger proportions of regular updates. We attribute such larger proportions of regular updates to the large number of setbacks caused by disorientation throughout the search process, which makes it easier for particles to correctly improve upon their personal best solutions.

4.4.3 Low and High Probabilities to Update

The quality of results shows no significant differences between PSO-LP and PSO-HP in the absence of noise or at very low levels, but PSO-LP finds significantly better solutions at higher levels of noise. However, we expect that PSO-LP will generally find better solutions also in the absence of noise and at very low levels too because its small probability to update will provide more exploitation than PSO-HP will, not to mention its smaller proportions of disorientation. Nonetheless, we expect the similarities to also depend on the characteristics of the optimization problem. Overall, the differences between the population statistics for PSO-LP and PSO-HP are analogous to those for PSO-LE and PSO-HE given their similar approaches to address the effect of noise, that is, small values for update probabilities favour exploitation whereas large update probabilities favour exploration. Hence, instead of comparing PSO-LP and PSO-HP, we focus on contrasting them against their evaporation counterparts.

In general, the quality of results shows that PSO-LP finds solutions of similar quality to PSO-LE, while PSO-HP finds significantly better solutions than PSO-HE. Despite that PSO-LP and PSO-LE present similarities in quality of results, an important difference between the algorithms is found in their optimization curves, where PSO-LE in the presence of noise tends to diverge over the iterations. Such a divergence is further emphasized in PSO-HE and it is certainly an important reason for concern when choosing the evaporation mechanism over probabilistic updates.

The proportions of regular operations, blindness and disorientation dif-
4.4. RESULTS AND DISCUSSIONS

fer between PSO-LP and PSO-LE only by about 10%, just like the difference between PSO-HP and PSO-HE, hence suggesting an approximate equivalence between the methods utilized to compute their respective update probabilities and evaporation factors. However, the most important difference between these statistics is that blindness and disorientation in PSO-PU remain constant regardless of the level of noise (except at $\sigma = 0.00$), whereas blindness increases in PSO-E with the level of noise while disorientation decreases. We attribute the stable operation of PSO-PU to the independence of the probabilistic updates from the optimization problem, whereas the performance of PSO-E largely relies on the objective space of the problem because the evaporation factor worsens the estimated objective values of the personal best solutions by an amount proportional to their initially estimated objective values. Furthermore, the stable operation of PSO-PU also extends to the causes of blindness and disorientation, where the proportions of blindness by memory and disorientation by memory are bound to increase in PSO-LP and PSO-HP because the estimated objective values of the their personal best solutions remain unchanged during the unsuccessful iterations, and hence will also be larger than those in PSO-LE and PSO-HE, respectively.

Lastly, the proportions of regular updates differ only by about 10% between PSO-LP and PSO-LE, just like PSO-HP and PSO-LP, thereby further supporting the approximate equivalence between the methods utilized to compute their respective update probabilities and evaporation factors. However, one of the most important differences between the population statistics is that the deterioration caused by disorientation in PSO-LP and PSO-HP is significantly smaller than that in their respective evaporation counterparts, namely PSO-LE and PSO-HE. Such a difference remarks another advantage of utilizing probabilistic updates over the evaporation mechanism, especially when considering that PSO-PU and PSO-E have similar proportions of blindness and disorientation.
4.4.4 Average Neighborhoods and Regular PSO

The quality of results shows that the regular PSO finds better solutions than PSO-AN in the absence of noise, similar solutions at a very low level of noise, and worse solutions at higher levels of noise. In the absence of noise, PSO provides more exploitation because its particles are partially attracted towards the true neighborhood best solution, whereas particles in PSO-AN are partially attracted towards average solutions whose ranking is mostly worse than those of the true best solutions. However, as the levels of noise increase, particles in PSO no longer select the true neighborhood best solutions, but instead the estimated best solutions whose true objective values will be somewhat arbitrary, hence these solutions partially attract the swarm towards sub-optimal regions of the search space. Differently, particles in PSO-AN are each partially attracted towards an average solution that will be better (on average) than any of the arbitrary solutions selected by particles in PSO (see Figure 4.4). Furthermore, thanks to averaging the neighborhood solutions, PSO-AN shows a rather stable quality of results regardless of the level of noise, whereas the quality of results of PSO shows more deterioration with increasing levels of noise.

All of the remaining population statistics are similar between PSO and PSO-AN given that the only difference between the algorithms is the selection of the neighborhood best solutions. However, we still find exceptions in the regular updates and (hence) regular discards in the absence of noise, where PSO-AN has a proportion of regular updates significantly larger than PSO. We attribute such a difference to the exploration capabilities of PSO-AN, in which taking an average neighborhood best solution slows down the speed of convergence, but increases the frequency of regular updates.

The proportions of blindness in both algorithms are quite large because none of their particles address this condition in any way. Consequently, blindness by memory is by far and almost in its entirety the major cause of blindness. Moreover, such large proportions of blindness reduce disori-
4.4. RESULTS AND DISCUSSIONS

Figure 4.4: Ranked Deception. The barplots represent the average percentile rank (left axis) based on the true objective values of the selected neighborhood best solutions with respect to the personal best solutions in the swarm for each algorithm (bottom axis) on $F_{01}$, $F_{03}$ and $F_{20}$ subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). In the absence of noise, particles from PSO do not suffer from ranked deception, for which the first bar is 0.0. The algorithms are abbreviated as (a) PSO and (n) PSO-AN. Lower percentile ranks are better.

...entation to minimal proportions given their inverse correlation (see Section 3.4.2, page 75). Furthermore, given the backward direction of the optimization problem, the objective values of the worse solutions to which particles become disoriented will be more corrupted by noise than those of their previous personal best solutions, hence making disorientation by the environment the major cause of disorientation.

Most importantly, the quality of results shows that both regular PSO and PSO-AN find solutions of similar or even better quality than any of the PSO-PU and PSO-E algorithms specifically designed to address the effect of noise. The proportions of blindness, regular updates and disorientation in the following algorithms are worthy of attention. First, PSO and PSO-AN have the largest proportions of blindness amongst all the algorithms, and yet their quality of results is generally better than that obtained with PSO-PU and PSO-E. Second, the proportions of regular updates in PSO and PSO-AN in the presence of noise are under 10%, whereas those in the other algorithms are over 10%. Lastly, PSO and PSO-AN have the
smallest proportions of *disorientation* amongst all the algorithms mostly due to their large proportions of *blindness*. From these statistics, we find that, while it is important to reduce blindness in the swarms, it is more important to prevent disorientation from increasing as its effect is far more deteriorating than any hypothetical improvement missed by blindness.

### 4.4.5 Further Discussions

The overall quality of results obtained with each of the algorithms is presented in Table 4.3, from which we can rank the algorithms from best to worst utilizing the concept of dominance over the total number of functions on which an algorithm is better, similar, or worse. Thus, the ranking of the algorithms is as follows: (1) PSO-AN, (2) PSO, (3) PSO-LP, (4) PSO-LE, (5) PSO-HP, and (6) PSO-HE, thereby supporting the discussions presented thus far. However, the following results from Table 4.3 seem contradictory: if PSO-LE finds better solutions than PSO in 28 cases and PSO-LP finds better solutions than PSO in only 3 cases, it is expected that PSO-LE would find better solutions than PSO-LP in more cases, but such is not the case as PSO-LP finds better solutions than PSO-LE in 54 cases and worse solutions in 37. Such an apparent contradiction is caused by the similar quality of solutions found by PSO and PSO-LP in a number of functions at which PSO-LE finds better solutions. These cases are shown in Table 4.4, where the functions are either primarily composed by the schwefel function \((F_{19}, F_{07}, F_{12}, F_{17})\) or belong to set \(B\) \((F_{04}, F_{05}, F_{07}, F_{08})\), thus suggesting that these problems have certain characteristics that favour the operation of PSO-LE.

The average population statistics for the algorithms across all benchmark functions are presented in Table 4.5, presenting the same trends that have been discussed in the previous sections. Some additional discussions regarding the lifetime and (binary and ranked) deception are presented as follows. The *lifetime* of the swarms in PSO-PU and PSO-E is the longest
Table 4.3: Summary of statistical significance tests on the quality of results. The tables count the number of functions in which the algorithm in parentheses produces a similar (‘=’) or a significantly better (‘−’) or worse (‘+’) quality of results than its counterpart at each level of noise σ. The statistical tests are performed utilizing the pairwise Wilcoxon test at $\alpha = 0.05$ with Holm [67] correction.

<table>
<thead>
<tr>
<th>Test</th>
<th>PSO-LE vs. PSO-HE</th>
<th>PSO-LP vs. PSO-HE</th>
<th>PSO vs. PSO-LE</th>
<th>PSO vs. PSO-LP</th>
<th>PSO-AN vs. PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>σ₀₀</td>
<td>8</td>
<td>12</td>
<td>16</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>σ₀₆</td>
<td>15</td>
<td>9</td>
<td>9</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>σ₁₂</td>
<td>19</td>
<td>12</td>
<td>10</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>σ₁₈</td>
<td>17</td>
<td>14</td>
<td>11</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>σ₂₄</td>
<td>16</td>
<td>13</td>
<td>8</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>σ₃₀</td>
<td>14</td>
<td>13</td>
<td>0</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>89</td>
<td>73</td>
<td>54</td>
<td>44</td>
<td>37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test</th>
<th>PSO-LE vs. PSO-HE</th>
<th>PSO-LP vs. PSO-HE</th>
<th>PSO vs. PSO-LE</th>
<th>PSO vs. PSO-LP</th>
<th>PSO-AN vs. PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>σ₀₀</td>
<td>8</td>
<td>15</td>
<td>16</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>σ₀₆</td>
<td>15</td>
<td>12</td>
<td>9</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>σ₁₂</td>
<td>19</td>
<td>14</td>
<td>10</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>σ₁₈</td>
<td>17</td>
<td>15</td>
<td>11</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>σ₂₄</td>
<td>16</td>
<td>13</td>
<td>8</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>σ₃₀</td>
<td>14</td>
<td>14</td>
<td>0</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>89</td>
<td>73</td>
<td>54</td>
<td>44</td>
<td>37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test</th>
<th>PSO-LE vs. PSO-HE</th>
<th>PSO-LP vs. PSO-HE</th>
<th>PSO vs. PSO-LE</th>
<th>PSO vs. PSO-LP</th>
<th>PSO-AN vs. PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>σ₀₀</td>
<td>2</td>
<td>15</td>
<td>16</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>σ₀₆</td>
<td>4</td>
<td>12</td>
<td>9</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>σ₁₂</td>
<td>6</td>
<td>14</td>
<td>10</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>σ₁₈</td>
<td>7</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>σ₂₄</td>
<td>7</td>
<td>13</td>
<td>8</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>σ₃₀</td>
<td>4</td>
<td>14</td>
<td>0</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>28</td>
<td>34</td>
<td>29</td>
<td>37</td>
<td>37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test</th>
<th>PSO-LE vs. PSO-HE</th>
<th>PSO-LP vs. PSO-HE</th>
<th>PSO vs. PSO-LE</th>
<th>PSO vs. PSO-LP</th>
<th>PSO-AN vs. PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>σ₀₀</td>
<td>2</td>
<td>0</td>
<td>16</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>σ₀₆</td>
<td>7</td>
<td>10</td>
<td>9</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>σ₁₂</td>
<td>6</td>
<td>14</td>
<td>10</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>σ₁₈</td>
<td>7</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>σ₂₄</td>
<td>7</td>
<td>13</td>
<td>8</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>σ₃₀</td>
<td>4</td>
<td>14</td>
<td>0</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>28</td>
<td>34</td>
<td>29</td>
<td>37</td>
<td>37</td>
</tr>
</tbody>
</table>
Table 4.4: Statistical significance tests on the quality of results that cause a contradictory summary. Each table presents the results of statistical significance tests between relevant pairs of algorithms. The algorithms are abbreviated as (a) PSO, (e) PSO-LE, and (p) PSO-LP. A blank cell indicates the algorithms are not statistically significant at the specified level of noise, whereas ‘−’ and ‘+’ indicate that the first algorithm is significantly better or worse than the second one (respectively). Results are obtained utilizing the pairwise Wilcoxon test at $\alpha = 0.05$ with Holm [67] correction. $F_07$ is composed of schwefel and also belongs to set B.

<table>
<thead>
<tr>
<th></th>
<th>schwefel</th>
<th>Set B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$F_{04}$</td>
<td></td>
</tr>
<tr>
<td>p vs. e</td>
<td>+ + + + + +</td>
<td>p vs. e</td>
</tr>
<tr>
<td>a vs. e</td>
<td>+ + + + + +</td>
<td>a vs. e</td>
</tr>
<tr>
<td>a vs. p</td>
<td>− − − − − −</td>
<td>a vs. p</td>
</tr>
<tr>
<td></td>
<td>$F_{05}$</td>
<td></td>
</tr>
<tr>
<td>p vs. e</td>
<td>+ + + + + +</td>
<td>p vs. e</td>
</tr>
<tr>
<td>a vs. e</td>
<td>+ + + + + +</td>
<td>a vs. e</td>
</tr>
<tr>
<td>a vs. p</td>
<td>− − − − − −</td>
<td>a vs. p</td>
</tr>
<tr>
<td></td>
<td>$F_{07}$</td>
<td></td>
</tr>
<tr>
<td>p vs. e</td>
<td>+ + + + + +</td>
<td>p vs. e</td>
</tr>
<tr>
<td>a vs. e</td>
<td>+ + + + + +</td>
<td>a vs. e</td>
</tr>
<tr>
<td>a vs. p</td>
<td>− − − − − −</td>
<td>a vs. p</td>
</tr>
<tr>
<td></td>
<td>$F_{08}$</td>
<td></td>
</tr>
<tr>
<td>p vs. e</td>
<td>+ + + + + +</td>
<td>p vs. e</td>
</tr>
<tr>
<td>a vs. e</td>
<td>+ + + + + +</td>
<td>a vs. e</td>
</tr>
<tr>
<td>a vs. p</td>
<td>− − − − − −</td>
<td>a vs. p</td>
</tr>
</tbody>
</table>
because their approach prevents them from converging up to a certain extent, whereas PSO and PSO-AN converge on average after performing 58.58% and 54.53% of the iterations (respectively). The binary deception is very high (over 96%) amongst all the algorithms because none of them actually aims to select the true neighborhood best solution. The ranked deception is similar between PSO and PSO-AN because, besides some exceptions, the neighborhood best solutions in PSO have better rankings at lower levels of noise, whereas PSO-AN has better rankings at higher levels of noise. Differently, the ranked deception for PSO-E and PSO-PU is much worse than that of the others because their large proportions of disorientation cause the objective values of the personal best solutions to be largely underestimated, thereby affecting the ranks of their selected neighborhood best solutions.

### 4.5 Summary

Particle Swarm Optimization is a metaheuristic whose quality of results significantly deteriorate on optimization problems subject to noise. In this type of problem, the effect of noise hinders the ability of particles to distinguish good from bad solutions, thus causing them to suffer from blindness, disorientation, and deception throughout the search. These three conditions are responsible for the deterioration of the quality of the results, and therefore noise mitigation mechanisms need to address them explicitly. Single-evaluation methods comprise those PSO algorithms whose noise mitigation mechanisms do not utilize the computational budget of function evaluations. Different from resampling methods, these PSO algorithms have more chance of finding better solutions given that the swarms perform more iterations, but such a higher chance comes at the cost of sacrificing the accuracy to which the objective values of the solutions are estimated.

The evaporation mechanism is a single-evaluation method upon which
Table 4.5: Average population statistics for the PSO algorithms over all benchmark functions, levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>LE</th>
<th>HE</th>
<th>LP</th>
<th>HP</th>
<th>PSO</th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifetime</td>
<td>98.99</td>
<td><strong>99.64</strong></td>
<td>97.85</td>
<td>99.50</td>
<td>58.58</td>
<td>54.53</td>
</tr>
<tr>
<td>Binary Deception</td>
<td>97.81</td>
<td>97.42</td>
<td>99.03</td>
<td>98.39</td>
<td><strong>97.03</strong></td>
<td>97.83</td>
</tr>
<tr>
<td>Ranked Deception</td>
<td>45.11</td>
<td>47.11</td>
<td>44.15</td>
<td>43.47</td>
<td><strong>29.61</strong></td>
<td>31.70</td>
</tr>
<tr>
<td><strong>Regular Operation</strong></td>
<td><strong>58.70</strong></td>
<td>56.11</td>
<td>52.16</td>
<td>52.42</td>
<td>46.51</td>
<td>42.12</td>
</tr>
<tr>
<td>Updates</td>
<td>16.38</td>
<td>35.62</td>
<td>14.81</td>
<td><strong>38.80</strong></td>
<td>3.69</td>
<td>3.58</td>
</tr>
<tr>
<td>Discards</td>
<td>83.62</td>
<td>64.38</td>
<td>85.19</td>
<td><strong>61.20</strong></td>
<td>96.31</td>
<td>96.42</td>
</tr>
<tr>
<td><strong>Blindness</strong></td>
<td>31.72</td>
<td><strong>20.91</strong></td>
<td>41.10</td>
<td>27.18</td>
<td>53.00</td>
<td>57.51</td>
</tr>
<tr>
<td>Memory</td>
<td>70.92</td>
<td>58.82</td>
<td>90.54</td>
<td>75.47</td>
<td>98.96</td>
<td>99.00</td>
</tr>
<tr>
<td>Environment</td>
<td>29.08</td>
<td>41.18</td>
<td>9.46</td>
<td>24.53</td>
<td>1.04</td>
<td>1.00</td>
</tr>
<tr>
<td><strong>Disorientation</strong></td>
<td><strong>9.57</strong></td>
<td>22.98</td>
<td>6.74</td>
<td>20.40</td>
<td>0.49</td>
<td><strong>0.37</strong></td>
</tr>
<tr>
<td>Memory</td>
<td>10.29</td>
<td>44.57</td>
<td>36.06</td>
<td>45.50</td>
<td>4.48</td>
<td>4.96</td>
</tr>
<tr>
<td>Environment</td>
<td>89.71</td>
<td>55.43</td>
<td>63.94</td>
<td>54.50</td>
<td>95.52</td>
<td>95.04</td>
</tr>
</tbody>
</table>

Larger $^+$ or smaller $^-$ is better. **Bold** value is best. Units in percentages (%).
other approaches have been based. PSO-E consists of an evaporation factor that worsens the estimated objective values of the personal best solutions after each unsuccessful iteration in order to reduce the presence of blindness. However, the operation of PSO-E largely depends on an evaporation factor which previous works have selected empirically after multiple experiments, but here we have analyzed its operation and derived a method to estimate the evaporation factors with a better idea of their effect over a number of unsuccessful iterations instead of its effect on the immediately next iteration. Specifically, our method starts from the assumption that doubling the estimated objective values would encourage particles to replace their personal best solutions, and based on the doubling time it is possible to choose the evaporation factor based on the number of unsuccessful iterations. We utilized our method to experiment with different evaporation factors, and we have reported the population statistics for PSO-E with a low and a high evaporation factor. While previous works have reported that PSO-E succeeds at finding better solutions than the regular PSO in different stochastic and dynamic optimization problems, we have shown that such is not generally the case for optimization problems subject to noise. In fact, the quality of its results turned out to be mostly worse than those obtained with the regular PSO lacking a noise mitigation mechanism. The main reason for such a worse quality of results is that its particles suffer too much disorientation throughout the search due to the evaporation mechanism, whereas those particles in regular PSO are rarely disoriented mostly due to the large proportions of blindness. Moreover, the quality of the results obtained with PSO-E deteriorate further with larger evaporation factors because the proportions of disorientation further increase. Besides the quality of results, the dissipative effect of the evaporation mechanism on the objective values has two other disadvantages on PSO-E, namely the divergent behaviour that swarms are prone to exhibit throughout the search process, and the large dependency not only to the value of the evaporation factor, but also to the objective space of the
optimization problem. Nonetheless, in spite of these disadvantages, the evaporation mechanism actively reduces the presence of blindness in the swarm, and therefore it is still a relevant noise mitigation mechanism for future works.

A better alternative to PSO-E is the new PSO algorithm that we proposed which utilizes probabilistic updates. Particles in PSO-PU probabilistically decide to replace their personal best solutions at each unsuccessful iteration, and thereby its operation does not rely on the objective space of the optimization problem like PSO-E does. The quality of its results was mostly better than that of PSO-E because the lack of evaporation allows the swarm to provide more exploitation and thereby reduce the chances of exhibiting divergent behaviour. While providing more exploitation implies that blindness is bound to increase, the probabilistic updates managed to reduce blindness to proportions similar to PSO-E. Moreover, the operation of PSO-PU was quite stable across the levels of noise as suggested by the quality of the results and the similar proportions of blindness, disorientation and regular operations obtained. Differently, the quality of the results obtained with PSO-E deteriorates with higher levels of noise, while the proportions of blindness further increased. Additionally, another advantage of PSO-PU is that setting the probabilities to update according to the number of unsuccessful iterations is more straightforward than setting the evaporation factors (even with the method that we derived), and its operation is more accurate as particles will replace their personal best solutions randomly according to a known uniform distribution without relying on the objective space. Nonetheless, the major problem of dissipative approaches acting on the particles’ memories (e.g. evaporation and probabilistic updates) is that the swarms no longer have a guaranteed convergence, that is, more iterations will not necessarily lead to better (or at least not worse) solutions. Instead, dissipative approaches could even lead to worse solutions over time, thereby describing a divergent behaviour and rendering the quality of the results highly dependent
on the number of iterations performed. Therefore, in these cases, it is necessary to report the optimization curves to observe the quality of results over time and detect potential divergent behaviours.

In spite of the improvements achieved with PSO-PU over PSO-E, the regular PSO algorithm still managed to find better solutions, the main reason being its minimal proportions of disorientation. While we expected disorientation to increase as a result of utilizing evaporation or probabilistic updates, we underestimated its effect on the quality of the results. Hence, while in Chapter 3 we suggested to focus on reducing blindness due to its presence in much larger proportions than disorientation, we strongly suggest that any approach designed to reduce blindness should as well prevent disorientation from increasing.

Lastly, PSO-AN presented a different approach which produced significantly better results than any of the aforementioned algorithms in the presence of noise. PSO-AN consists of particles creating their own neighborhood best solutions by averaging the personal best solutions of the particles within their neighborhoods. In doing so, particles are more often partially attracted towards better neighborhood solutions than particles in the regular PSO are, and the evidence shows a positive correlation between better ranked deception and better quality of results in more than half of the benchmark functions at medium to high levels of noise (refer to Appendix 6.D). Moreover, PSO-AN belongs to the family of single-evaluation methods because creating the centroid solutions does not require to be evaluated, although we have evaluated such solutions just for computing the population statistics of deception. Therefore, in terms of simplicity and quality of results, PSO-AN is the best single-evaluation PSO algorithm to address optimization problems subject to noise.
Next Chapter

The next chapter will present a study on the population statistics for resampling-based PSO algorithms on optimization problems subject to noise.

4.A Population Statistics

The following figures correspond to the population statistics on each benchmark function for the algorithms utilized in the design of experiments from this chapter (Section 4.3, page 111).
Figure 4.5: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, (n) PSO-AN. The boxplots are coloured from dark to light grey to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
4.A. POPULATION STATISTICS

Figure 4.6: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, (n) PSO-AN. The boxplots are coloured from dark to light gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
Figure 4.7: Binary Deception. The barplots represent the average proportions of iterations (left axis) at which a particle is deceived by its neighbors for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, (n) PSO-AN. Particles in PSO-ERGC do not suffer from deception. Smaller proportions are better.
Figure 4.8: Ranked Deception. The barplots represent the average percentile rank (left axis) based on the true objective values of the selected neighborhood best solutions with respect to the true objective values of the personal best solutions in the swarm for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, (n) PSO-AN. Particles from PSO-ERGC do not suffer from deception and hence their neighborhood best solutions are always ranked best. Lower percentile ranks are better.
Figure 4.9: Regular Operation, Blindness and Disorientation. The stacked barplots represent the average proportions (left axis) of regular operation (dark gray), blindness (medium gray) and disorientation (light gray) experienced by a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, (n) PSO-AN. Larger proportions of regular operation and smaller proportions of blindness and disorientation are better.
4.A. POPULATION STATISTICS

<table>
<thead>
<tr>
<th>Sets A and E</th>
<th>Set B</th>
<th>Set C</th>
<th>Set D</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Barplot" /></td>
<td><img src="image2.png" alt="Barplot" /></td>
<td><img src="image3.png" alt="Barplot" /></td>
<td><img src="image4.png" alt="Barplot" /></td>
</tr>
</tbody>
</table>

Figure 4.10: Regular Updates and Discards. The stacked barplots represent the average proportions (left axis) of regular updates (dark gray) and discards (light gray) experienced by a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise \( \sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\} \) (bars from left to right). The algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, (n) PSO-AN. Larger proportions of regular updates and smaller proportions of regular discards are better.
Figure 4.11: Causes of Blindness. The stacked barplots represent the average proportions (left axis) of blindness caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, (n) PSO-AN.
Figure 4.12: Causes of Disorientation. The stacked barplots represent the average proportions (left axis) of disorientation caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, (n) PSO-AN.
Figure 4.13: Lifetime. The barplots represent the proportions of average lifetime (left axis) of a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, (n) PSO-AN. A longer lifetime is better when the swarm does not converge to the global optimum.
Figure 4.14: Effect of Disorientation and Blindness. The stacked barplots represent the normalized average magnitudes (left axis) of deterioration caused by disorientation (dark gray) and hypothetical improvement missed by blindness (light gray) on the benchmark functions subject to the levels of noise $\sigma \in \{0.00, 0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (e) PSO-LE, (E) PSO-HE, (p) PSO-LP, (P) PSO-HP, (a) PSO, (n) PSO-AN. Particles from the regular PSO and PSO-AN do not suffer from blindness or disorientation in the absence of noise.
Figure 4.15: Optimization Curves. The plots represent the average objective values (left axis) of the personal best solutions over all the independent runs at each iteration (bottom axis) of PSO-LE (marked with ‘•’) and PSO-LP (no marks) on the benchmark functions subject to levels of noise $\sigma_{00}$ (solid line), $\sigma_{06}$ (long-dashed line), $\sigma_{18}$ (short-dashed line), and $\sigma_{30}$ (dotted line). The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
Chapter 5

Resampling Methods

This chapter studies the population statistics for resampling-based PSO algorithms. The set of algorithms under study consists, on the one hand, of the PSO-ER and PSO-OCBA algorithms already proposed in the literature, and on the other hand, of two new algorithms proposed in this chapter that allocate the computational budget differently.

This chapter is structured as follows. Section 5.1 introduces this chapter. Section 5.2 presents the new resampling-based PSO algorithms. Section 5.3 describes the design of experiments. Section 5.4 presents the results and discussions. Finally, Section 5.5 ends this chapter with a summary.

5.1 Introduction

One type of noise mitigation mechanism in PSO comprises resampling methods, which serve to better estimate the true objective values of the solutions by taking a sample mean over multiple re-evaluations. As such, the more evaluations performed to a solution, the more accurate its objective value will be. However, since these evaluations are taken from a fixed and limited computational budget of evaluations available to the PSO algorithm, improving the accuracy of the solutions sacrifices the number of itera-
tions to perform. In spite of such a tradeoff, the most basic resampling-based PSO algorithm, namely PSO-ER, finds better solutions than the regular PSO thanks to better estimating the objective values of the current solutions in the swarm (see Chapter 3). However, the quality of its results is still generally worse than the more advanced resampling-based PSO-OCBA [7, 106, 154]. While the underlying reasons to such a superior performance have been implicitly attributed to the reduction of deception, blindness and disorientation [7], we are interested in quantifying such a reduction in order to have a baseline to improve upon. The PSO-ER and PSO-OCBA are described in Sections 2.6.2 and 2.6.3 (pages 35 and 39, respectively).

The population statistics proposed in Chapter 3 showed that PSO-ER finds better solutions than the regular PSO mostly because its particles suffer less often from deception, blindness and disorientation, and yet these conditions are still present in 92.74%, 36.13% and 2.20% of the iterations, respectively (see Table 3.3, page 89). To reduce the extents to which these conditions affect particles in PSO-ER, we propose a new PSO with Extended Equal Resampling (PSO-EER) in which the personal best solutions are also re-evaluated, thereby improving the accuracy of their objective values and hence directly reducing blindness and deception which are the most common causes of deterioration in PSO-ER. In doing so, we expect PSO-EER to find better solutions than PSO-ER, but we are not certain how the quality of its results will compare against PSO-OCBA because its population statistics have never been computed. Besides PSO-EER, we also propose another PSO with Equal Resampling and Allocations to the Top-N Solutions (PSO-ERN), in which the additional computational budget is allocated amongst the estimated best current solutions and particles attempt to reduce the proportions of blindness by prefering accuracy over quality.
5.2. RESAMPLING-BASED PSO ALGORITHMS

Chapter Goals

The overall goal of this chapter is to study the population statistics for resampling-based PSO algorithms on optimization problems subject to different levels of multiplicative Gaussian noise. Specifically, we will focus on the following objectives.

- Characterize the resampling-based PSO algorithms according to their population statistics.
- Compare the population statistics of PSO-OCBA, PSO-ERN, PSO-EER and PSO-ER.
- Evaluate PSO-EER utilizing different allocations of the computational budget.

5.2  Resampling-Based PSO Algorithms

Resampling methods in PSO are noise mitigation mechanisms that estimate the true objective values of the solutions by performing multiple evaluations to the solutions and then taking a sample mean over the evaluations. As such, the more re-evaluations performed to a solution, the more accurate its objective value will be because the standard error will reduce proportional to \(1/\sqrt{n}\) after \(n\) evaluations. However, since the additional evaluations performed are extracted from a fixed and limited computational budget \(B\) available to the algorithm, incorporating a resampling method into PSO creates a tradeoff between the number of re-evaluations and the number of iterations. Consequently, resampling-based PSO algorithms need to decide upon a balance between the accuracy of the existing solutions and the exploration of new solutions. Once this tradeoff has been decided upon, the problem is then to allocate the computational budget \(B^t\) amongst the solutions at iteration \(t\).
CHAPTER 5. RESAMPLING METHODS

Resampling-based PSO algorithms divide the computational budget $B$ in budgets $B^\alpha_t$ and $B^\beta_t$, one allocated after the other. The first, $B^\alpha_t$, is allocated equally between the current solutions in the swarm, and the second budget, $B^\beta_t$, is allocated according to the criteria of the algorithm. The following sections present the two algorithms we proposed for a minimization problem subject to noise, where the computational budget $B$ is the same at every iteration.

5.2.1 PSO with Extended Equal Resampling

PSO with Extended Equal Resampling (PSO-EER) is an extension to PSO-ER that we propose to reduce the deception and blindness in the swarm and hence find better solutions. The PSO-EER consists of equally allocating the computational budget $B^\beta_t$ between the personal best solutions instead of between the current solutions, the latter of which are previously estimated with $B^\alpha_t$. The PSO-EER algorithm is presented in Figure 5.1 where $b^\alpha$ and $b^\beta$ are the individual computational budgets allocated to each particle from $B^\alpha_t$ and $B^\beta_t$ (respectively), $\hat{f}(x^t_i)$ is a noisy evaluation of the current solution $x^t_i$, $X^t_i$ is the set of evaluations performed on solution $x^t_i$, $\hat{f}(y^t_i)$ is a noisy evaluation of personal best solution $y^t_i$, and $Y^t_i$ is the set of evaluations performed on the personal best solution $y^t_i$.

This extension is motivated by the population statistics computed in Chapter 3 where particles suffer from deception on 92.74% of the iterations and from blindness on 36.13% of the iterations, from which 91.85% correspond to blindness by memory (see Table 3.3, page 89). While we expect to improve upon the quality of the results of PSO-ER by reducing blindness and deception, we also expect that disorientation will increase as a consequence of less accurately estimating the objective values of the current solutions. Certainly, the increase of disorientation will deteriorate the quality of the results as suggested in Chapter 4 but considering that the objective values of the solutions in PSO-EER are better estimated, we
foreach iteration $t$ do
  foreach particle $i$ in swarm $S$ do
    $X^t_i \leftarrow \bigcup_{b} \alpha \hat{f}(x^t_i)$
    $Y^{t-1}_i \leftarrow Y^{t-1}_i \cup \left( \bigcup_{b} \beta \hat{f}(y^{t-1}_i) \right)$
    if $\text{mean}(X^t_i) < \text{mean}(Y^{t-1}_i)$ then
      $y^t_i \leftarrow x^t_i$, $Y^t_i \leftarrow X^t_i$
    else
      $y^t_i \leftarrow y^{t-1}_i$, $Y^t_i \leftarrow Y^{t-1}_i$
    foreach particle $i$ in swarm $S$ do
      $\hat{y}^t_i \leftarrow y^t_\omega$ $| \omega = \arg \min_{j \in N_i} \text{mean}(Y^t_j)$
  foreach particle $i$ in swarm $S$ do
    update $v^{t+1}_i$ and $x^{t+1}_i$ using (2.3) and (2.4)

Figure 5.1: PSO with Extended Equal Resampling (PSO-EER).

expect such an increase not to deteriorate the quality of the results much in this case, especially considering that particles in PSO-ER suffer from disorientation only in 2.20% of the iterations (see Table 3.3, page 89).

5.2.2 PSO with ER and Allocations to the Top-$N$ Solutions

PSO with Equal Resampling and Allocations to the Top-$N$ Solutions (PSO-ERN) is an algorithm that we propose as a simpler and computationally cheaper alternative to PSO-OCBA. First, PSO-ERN allocates the $B^t_\alpha$ evaluations equally between the current solutions (like PSO-OCBA), and then allocates the $B^t_\beta$ evaluations between the estimated best $N$ current solutions. The budget $B^t_\beta$ is allocated at once (unlike the sequential approach in PSO-OCBA), thereby having a cheaper computational cost. Consequently, PSO-ERN does not strive to minimize deception, but attempts to reduce it by selecting the neighborhood best solution from those solutions with the most evaluations. The particles in PSO-ERN update their personal best solutions not only when their current solutions are better, but also when they
have more evaluations because their objective values will be potentially better estimated, and hence blindness will be potentially reduced.

The PSO-ERN algorithm is presented in Figure 5.2, where $b_\alpha$ is the individual computational budget allocated to each particle from $B_\alpha^t$, $b_i$ is the individual computational budget allocated from $B_\beta^t$ to particle $i$, $N$ is the number of current solutions whose objective values will be more accurately estimated, $\hat{f}(x^t_i)$ is a noisy evaluation of the current solution $x^t_i$, $X^t_i$ is the set of evaluations performed on the current solution $x^t_i$, $Y^t_i$ is the set of evaluations performed on the personal best solution $y^t_i$, and $A_i$ is the set of particles whose personal best solutions have the most evaluations in neighborhood $N_i$.

The computational budget $B_\beta^t$ is allocated according to Equation (5.1), where $w$ is a vector of weights with $w_i \in (0, 1]$ such that $\sum w_i = 1$, and the parentheses in $b_{(i)}$ indicate that the individual computational budget $b_i$ belongs to the particle at index $i$ in the ordered set of $N$ particles sorted from better to worse according to the estimated objective values of their current solutions. As such, the parameters $N$ and $w$ provide a direct control over the allocation of $B_\beta^t$ balancing the tradeoff between significantly improving the accuracy of a few solutions to slightly improving the accuracy of all solutions. Furthermore, notice that PSO-ER is a setting of PSO-ERN when $N$ equals the number of particles in the swarm and $b_{(1)} = 1$.

$$w B_\beta^t = [w_1 B_\beta^t, w_2 B_\beta^t, \ldots, w_N B_\beta^t]$$

$$= [b_{(1)}, b_{(2)}, \ldots, b_{(N)}]$$

### 5.3 Design of Experiments

The algorithms for which we want to study the population statistics on optimization problems subject to noise are PSO-ER, PSO-ERGC (from Chapter 3), PSO-OCBA, PSO-ERN, and PSO-EER. Specifically, we want to pro-
foreach iteration $t$ do
  foreach particle $i$ in swarm $S$ do
    $X_t^i \leftarrow \cup_b \hat{f}(x_t^i)$
  endforeach
  foreach particle $i$ in $top(N, S)$ do
    $X_t^i \leftarrow X_t^i \cup \left( \cup_b \hat{f}(x_t^i) \right)$
  endforeach
  foreach particle $i$ in swarm $S$ do
    if $\text{mean}(X_t^i) < \text{mean}(Y_{t-1}^i)$ or $|X_t^i| > |Y_{t-1}^i|$ then
      $y_t^i \leftarrow x_t^i$, $Y_t^i \leftarrow X_t^i$
    else
      $y_t^i \leftarrow y_{t-1}^i$, $Y_t^i \leftarrow Y_{t-1}^i$
    endif
  endforeach
  foreach particle $i$ in swarm $S$ do
    $A_i \leftarrow \arg\max_{j \in N_i} |Y_j^t|$
    $\hat{y}_t^i \leftarrow y_t^j | \omega = \arg\min_{j \in A_i} \text{mean}(Y_j^t)$
  endforeach
  foreach particle $i$ in swarm $S$ do
    update $v_t^{i+1}$ and $x_t^{i+1}$ using (2.3) and (2.4)
  endforeach

Figure 5.2: PSO with Equal Resampling and Allocations to the Top-$N$ Solutions (PSO-ERN).

provide insights about the effect of noise on these algorithms beyond the expected deterioration of the quality of their results. The swarms consist of 50 particles and a computational budget of $B = 30\,000$ function evaluations distributed over 100 iterations, resulting in $B_t = 300$ function evaluations for each iteration. The budget $B_t$ is divided in $B_t^\alpha = 250$ and $B_t^\beta = 50$, thereby allocating five evaluations from $B_t^\alpha$ to each of the 50 current solutions, and leaving $B_t^\beta$ to be allocated according to the algorithm. The computational budget is set to match that in Chapters 3 and 4, and the distribution of $B_t$ is set taking into consideration the guidelines for OCBA presented in [25], which suggest to utilize at least five evaluations for each solution to estimate its mean and variance.

Two settings of the PSO-EER algorithm are evaluated. On the one hand, PSO-EER1 has each particle allocate one evaluation to its personal
best solution, for which the computational budget $B^t_\beta = 50$ is equally allocated between the personal best solutions in the swarm. On the other hand, PSO-EER2 has each particle allocate two evaluations to its personal best solution, for which the computational budget $B^t$ needs to be divided into $B^t_\alpha = 200$ and $B^t_\beta = 100$. Consequently, particles in PSO-EER1 and PSO-EER2 balance differently the accuracy tradeoff between their current and personal best solutions.

The particles in all swarms utilize the inertia $w$ and acceleration coefficients $c_1, c_2$ from Equation (2.3) according to the guidelines in [145], and their neighborhoods are defined by the star topology. Additionally, particles limit their velocities utilizing the hyperbolic tangent function to reduce the sensitivity of setting a maximum velocity [48], which is determined according to the limits $x_{\min}$ and $x_{\max}$ of the optimization problem. In general, the experiments are performed under the same settings as Chapters 3 and 4, but utilizing a different set of algorithms.

The population statistics are computed for each algorithm based on the information collected from all 50 particles across 100 iterations at each independent run from a total of 50. Thus, the statistics at each level of noise are made up from $50 \times 100 \times 50 = 250,000$ observations. The complete list of parameter settings is presented in Table 5.1.

### 5.4 Results and Discussions

Our discussions are based on the population statistics for the algorithms on benchmark function $F_{13}$ because these reflect most of the trends found in the other benchmark functions, all of which are presented in Appendix 5.A (page 167). The results and discussions are structured as follows. Section 5.4.1 presents and describes the population statistics for the algorithms on benchmark function $F_{13}$. Sections 5.4.2, 5.4.3 and 5.4.4 present our discussions based on pairwise comparisons of the population statistics for PSO-OCBA and PSO-ERN, PSO-EER1 and PSO-EER2, and PSO-ERGC.
### 5.4. RESULTS AND DISCUSSIONS

Table 5.1: Parameter settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent runs</td>
<td>50 with 30,000 function evaluations</td>
</tr>
<tr>
<td>Number of particles</td>
<td>$n = 50$ in $\mathbb{R}^{1000}$ with star topology</td>
</tr>
<tr>
<td>Acceleration</td>
<td>Static with $c_1 = c_2 = 1.49618$</td>
</tr>
<tr>
<td>Inertia</td>
<td>Static with $w = 0.729844$</td>
</tr>
<tr>
<td>Maximum velocity</td>
<td>$v_{\text{max}} = 0.25 \cdot</td>
</tr>
<tr>
<td>Velocity clamping</td>
<td>$\dot{v}<em>{ij}^t = v</em>{\text{max}} \cdot \tanh \left( \frac{v_{ij}^t}{v_{\text{max}}} \right)$</td>
</tr>
<tr>
<td>Severity of noise</td>
<td>$\sigma \in {0.06, 0.12, 0.18, 0.24, 0.30}$</td>
</tr>
<tr>
<td>Computational budget</td>
<td>$B = 30,000$, $B_\alpha' = 250$, $B_\beta' = 50$</td>
</tr>
<tr>
<td>Iterations</td>
<td>$B / (B_\alpha' + B_\beta') = 100$</td>
</tr>
<tr>
<td>PSO-ER, PSO-ERGC</td>
<td>$b_\alpha = 5$, $b_\beta = 1$</td>
</tr>
<tr>
<td>PSO-OCBA</td>
<td>$b_\alpha = 5$, $b_\Delta = 5$</td>
</tr>
<tr>
<td>PSO-ERN</td>
<td>$N = 2$, $b_\alpha = 5$, $b_{(1)} = b_{(2)} = 25$</td>
</tr>
<tr>
<td>PSO-EER1</td>
<td>$b_\alpha = 5$, $b_\beta = 1$</td>
</tr>
<tr>
<td>PSO-EER2</td>
<td>$B_\alpha' = 200$, $B_\beta' = 100$, $b_\alpha = 4$, $b_\beta = 2$</td>
</tr>
</tbody>
</table>
and PSO-ER, respectively. Finally, Section 5.4.5 presents a ranking of the algorithms based on the overall quality of the results obtained and a summary of the population statistics on all benchmark functions.

5.4.1 Population Statistics

The population statistics on $F_{13}$ are presented in Figure 5.3, where the algorithms are abbreviated as (g) PSO-ERGC, (o) PSO-OCBA, (n) PSO-ERN, (e1) PSO-EER1, (e2) PSO-EER2, and (e) PSO-ER. The statistics are presented as follows.

*Quality of Results.* The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) on $F_{13}$ subject to the different levels of noise (top axis). In this case only, PSO-EER1 and PSO-EER2 are abbreviated as (1) and (2), respectively. The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems and therefore lower objective values indicate better solutions.

*Binary Deception.* The barplots represent the average proportion of iterations (left axis) in which a particle for each algorithm (bottom axis) is deceived by its neighbors on $F_{13}$ subject to the different levels of noise (bars coloured from light to dark gray). Smaller proportions are better.

*Ranked Deception.* The barplots represent the average percentile rank (left axis) of the neighborhood best solution chosen by a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars coloured from light to dark gray). Smaller ranks are better.

*Regular Operation, Blindness and Disorientation.* The stacked barplots represent the average proportions (left axis) of regular operation (dark gray), blindness (medium gray) and disorientation (light gray) experienced by a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars from left to right). Larger proportions of regular operations and smaller proportions of blindness and disorientation
5.4. RESULTS AND DISCUSSIONS

are better.

Regular Updates and Discards. The stacked barplots represent the average proportions (left axis) of regular updates (dark gray) and discards (light gray) experienced by a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars from left to right). Larger proportions of regular updates and smaller proportions of regular discards are better.

Causes of Blindness. The stacked barplots represent the average proportions of blindness (left axis) caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars from left to right).

Causes of Disorientation. The stacked barplots represent the average proportions of disorientation (left axis) caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars from left to right).

Lifetime. The barplots represent the normalized average lifetime (left axis) of a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars coloured from light to dark gray). A longer lifetime is better when the swarm does not converge to the global optimum solution.

The population statistics in Figure 5.3 are discussed in the following sections and are referred to by their names in italics the first time that are brought up for discussion.

5.4.2 PSO-OCBA and PSO-ERN

The quality of results shows that PSO-OCBA finds better solutions than PSO-ERN regardless of the level of noise, although the quality of both deteriorate as the level of noise increases. These results show significant differences between PSO-OCBA and PSO-ERN unlike the mostly not significant differences found in [123]. The underlying reason for these results
Figure 5.3: Population statistics on benchmark function $F_{13}$. The presentation details of each statistic are described in Section 5.4.1 (page 154).
to differ from those in [123] is that the PSO-OCBA utilized here (and in [7, 106, 154]) allocates the additional computational budget $B_\beta$ amongst the current and personal best solutions, whereas that in [123] allocates it only between the current solutions, hence not necessarily reducing deception as originally intended. The differences between PSO-OCBA and PSO-ERN with regards to the remaining population statistics are discussed next.

The proportions of binary deception are smaller for PSO-OCBA (between 60–90%) than for PSO-ERN (between 70–95%) by about 10% at each level of noise, that is, particles in PSO-OCBA are 10% more likely to correctly select their neighborhood best solutions. Moreover, the ranked deception shows that, even if particles in PSO-OCBA fail to select their true neighborhood best solutions, the quality of whatever solution they select will still be better than any neighborhood best solution selected by the particles in PSO-ERN. The better proportions of binary and ranked deception in PSO-OCBA are due to its sequential allocation of the additional computational budget, whereby apparently good solutions have more evaluations allocated that will either reaffirm their quality or discourage their selection as neighborhood best solutions. Differently, particles in PSO-ERN select their neighborhood best solutions only from those which have the most evaluations, that is, those solutions which have been the best current solutions at previous iterations and still remain in memory. The better proportions of binary and ranked deception in PSO-OCBA significantly contribute to its superior quality of results over PSO-ERN, which confirms that reducing deception in a swarm leads to better results, as it was suggested in Chapter 3.

The proportions of regular operation are larger in PSO-OCBA (over 60%) than in PSO-ERN (over 50%), while the proportions of blindness are smaller (20–30% compared to 25–40%). On the one hand, the larger proportions of regular operations and smaller proportions of blindness in PSO-OCBA are attributed to the sequential allocation being performed between both current and personal best solutions. The sequential allocation corrects
the most importantly underestimated objective values of the solutions, thereby reducing the proportions of blindness and disorientation and indirectly increasing the proportions of regular operation. Specifically, correcting the underestimated objective values of the personal best solutions reduces the probability of blindness because it reduces the range of better solutions to which particles are blinded, and correcting the underestimated objective values of the current solutions reduces the probability of disorientation. Consequently, by reducing blindness and disorientation, the proportions of regular operations are expected to increase. On the other hand, particles in PSO-ERN suffer more often from blindness, hence reducing the proportions of regular operations. Blindness is larger in PSO-ERN due to the small number of evaluations $b_{\alpha} = 5$ upon which the objective values of most personal best solutions are estimated, only two of which will be based on $b_{\alpha} + b_{(i)} = 30$ evaluations and perhaps a few others from previous iterations. Consequently, not only the proportions of regular operations will be reduced due to blindness, but also the lifetime of the swarm will be shortened in spite of its update mechanism by which more accurate solutions will also update. Regarding disorientation, the proportions are rather similar between PSO-OCBA and PSO-ERN, although we expected the latter to suffer more from disorientation given the precedence of accuracy over quality in its update mechanism of the personal best solutions. However, considering that there will be at most two particles of PSO-ERN at each iteration updating their personal best solutions on the grounds of accuracy, it explains why disorientation does not increase much (even if it does).

The proportions of blindness by memory are larger than blindness by the environment in both algorithms because the ignored current solutions which are truly better than the personal best solutions will generally be more accurate given the backward direction of the optimization problem (refer to Section 3.4.2, page 75). PSO-OCBA has smaller proportions of blindness by memory than PSO-ERN (60–75% compared to 70–85%) because the
accuracy of its personal best solutions will potentially be improved over the iterations, whereas such is not the case in PSO-ERN. The proportions of disorientation by the environment are larger than disorientation by memory in both algorithms because, in a backward optimization problem, the worse solutions to which particles update will have larger standard deviations than their once personal best solutions (refer to Section 3.4.2 page 75). PSO-ERN has larger proportions of disorientation by memory than PSO-OCBA (20–25% compared to 5–20%), which we attribute partly to the precedence of accuracy over quality in its position update mechanism and partly to the less accurate objective values of most of its current solutions. Lastly, the proportions of regular updates between PSO-OCBA and PSO-ERN are rather similar (20–45%), although the former has a larger number of both because its proportions of regular operations are slightly larger.

In general, PSO-ERN finds worse solutions than PSO-OCBA as expected from the population statistics. Nonetheless, its computationally cheaper model may prove convenient for time-limited environments where PSO-ERN would be able to perform more iterations than PSO-OCBA. Furthermore, we expect that allocating $B_j$ in PSO-ERN between a larger number of solutions than two might improve the quality of its results.

5.4.3 PSO-EER1 and PSO-EER2

The quality of results shows that PSO-EER1 finds solutions that are slightly better than PSO-EER2, and both of them are better than PSO-ER. These results show the importance of continuing to improve the accuracy to which the objective values of the personal best solutions are estimated. Additionally, these results confirm that reducing blindness allows the swarm to find better solutions, although the quality of results is also influenced by other factors that are discussed next.

The proportions of binary deception between PSO-EER1 and PSO-EER2
are rather similar (75–95%), thus suggesting that allocating one additional evaluation to estimate the objective values of the personal best solutions does not make an important difference. This is further confirmed with the ranked deception showing that the quality of the selected neighborhood best solutions is also similar (top 10–40%). However, the ranked deception of PSO-EER1 and PSO-EER2 is better by about 10% with respect to PSO-ER at low levels of noise and rather similar at high levels. In this case, allocating an additional evaluation to the personal best solutions (PSO-EER1) does make a difference because it allows particles to recover from blindness up to a certain extent, whereas particles in PSO-ER will remain blinded until their personal best solutions are replaced with new ones.

The proportions of blindness are slightly smaller in PSO-EER2 (10–20%) than in PSO-EER1 (10–25%), while the proportions of disorientation are slightly larger (5–15% compared to 5–10%). The proportions of blindness in PSO-EER2 are smaller because the objective values of its personal best solutions are estimated more accurately thanks to allocating two additional evaluations at every iteration. However, such an increased accuracy comes at the cost of estimating less accurately the objective values of the current solutions, thereby increasing the chance of disorientation which contributes to the deterioration of its results. Compared to PSO-ER, which shows 25–50% of blindness and 1–5% of disorientation, the differences are noteworthy because particles in PSO-ER have the objective values of their current solutions estimated more accurately and therefore the proportions of disorientation are reduced. However, since the objective values of the personal best solutions are less accurately estimated, PSO-ER shows larger proportions of blindness. The superior quality of results obtained with PSO-EER1 and PSO-EER2 with respect to PSO-ER is not only because they have smaller proportions of blindness, but also because they have larger proportions of regular operation, and especially larger proportions of regular updates which are crucial to finding better solutions. These proportions are also reflected in the longer lifetime of the swarms from PSO-EER1 and
PSO-EER2.

The causes of blindness show important differences between PSO-EER1 and PSO-EER2 that further support our analysis. The proportions of blindness by memory are larger in PSO-EER1 (60–75%) than in PSO-EER2 (50–60%) because the objective values of its personal best solutions are estimated less accurately. Conversely, the proportions of blindness by the environment are larger in PSO-EER2 because the objective values of its current solutions are estimated less accurately. These proportions compared to PSO-ER (85–95%) are significantly smaller because the accuracy of the objective values of their personal best solutions is continuously improving, whereas such is not the case in PSO-ER. Regarding the causes of disorientation, we expected to find the smallest proportions of disorientation by memory in PSO-EER2 because its personal best solutions are estimated more accurately, but instead the statistic shows similar proportions to those in PSO-EER1 and PSO-ER, and even slightly larger at times. Still, the differences in disorientation by memory are rather small, especially considering that these are made up from the already small proportions of disorientation.

5.4.4 PSO-ERGC and PSO-ER

The quality of results shows that PSO-ERGC and PSO-ER find the best and worst solutions (respectively) amongst the algorithms. The only difference between PSO-ERGC and PSO-ER is that particles in PSO-ERGC do not suffer from deception whereas those in PSO-ER not only have the highest proportions of binary deception amongst the algorithms, but also the worst proportions of ranked deception. Despite that PSO-ERGC has the unrealistic assumption of global certainty and PSO-ER has a poor quality of results, both algorithms are still important references to compare against.

The proportions of regular operation and blindness, however, show counterintuitive results as PSO-ERGC has smaller proportions of regular oper-
ations than PSO-ER (30–70% compared to 50–70%) and larger proportions of blindness (25–70% compared to 25–50%). These results are counterintuitive because both algorithms are under the same initial conditions, the current and personal best solutions are estimated from the same number of evaluations, and the only difference is the lack of deception in PSO-ERGC. The larger proportions of blindness could be due to a fast convergence of the swarm thanks to the attraction exerted by the true neighborhood best solution, but a more detailed study is needed to confirm that as the underlying reason to such high proportions of blindness. The effect of blindness is noticeable in the shorter lifetime of both swarms with respect to the other algorithms. Regarding the remaining statistics, the proportions of regular updates are larger for PSO-ERGC (30–50% compared to 10–30%) as particles are more likely to find better solutions when partially attracted to better regions of the search space, and the causes of blindness and disorientation are similar because the algorithms do not address blindness or disorientation differently.

5.4.5 Further Discussions

The algorithms can be ranked according to the transitive relation noticeable in the overall quality of their results shown in Table 5.2, thus ranking from best to worst as follows: PSO-ERGC, PSO-OCBA, PSO-ERN, PSO-EER1, PSO-EER2, and PSO-ER. This ranking not only shows the importance of reducing deception in PSO, but also the significant improvements that can be achieved by allocating \( B^\beta \) evaluations differently from straight equal resampling. The underlying reasons to the quality of these results can be found in Table 5.3 where the average population statistics are presented for each algorithm across all benchmark functions. PSO-ER finds the worst solutions because its particles suffer from the largest proportions of deception and from the second largest proportions of blindness, which in turn leads to the shortest lifetime amongst the algorithms,
all these in spite of having minimal proportions of disorientation. PSO-EER2 finds better solutions than PSO-ER because it reduces blindness and deception at the cost of increasing disorientation. PSO-EER1 finds better solutions than PSO-EER2 because its particles suffer less often from disorientation despite a small increase in blindness and deception. PSO-ERN finds better solutions than PSO-EER1 mostly thanks to its better ranked deception, but also to its reduced proportions of disorientation. PSO-OCBA finds better solutions than the previous algorithms because its sequential allocation method reduces blindness and disorientation of the most important particles in order to minimize the proportions of deception in the swarm. PSO-ERGC finds the best solutions because its particles never suffer from deception and have the second highest proportions of regular updates despite having the highest proportions of blindness and the smallest proportions of regular operations. While PSO-EER2 has the best statistics of lifetime, regular operations, regular updates and discards, and blindness, its main drawback is having the worst proportions of disorientation which further confirm the important setbacks disorientation cause throughout the search.

5.5 Summary

Resampling methods in PSO are noise mitigation mechanisms that help reduce deception, blindness and disorientation by better estimating the objective values of the solutions after multiple re-evaluations. The most basic resampling-based PSO algorithm is PSO-ER, which re-evaluates all the current solutions in the swarm the same number of times. Since the objective values of the personal best solutions are better estimated, its particles suffer less from blindness and deception than the regular PSO (refer to Section 3.4.2, page 75), and hence the quality of its results is significantly better. However, compared to other resampling-based PSO algorithms, PSO-ER finds worse solutions mainly due to the large proportions of de-
Table 5.2: Summary of statistical significance tests on the quality of results. The tables count the number of functions in which the quality of results from a PSO algorithm is similar (‘=’) or significantly better (‘−’) or worse (‘+’) than its PSO counterpart at each level of noise $\sigma$ according to the pairwise Wilcoxon test at $\alpha = 0.05$ with Holm [67] correction.

<table>
<thead>
<tr>
<th>ERGC vs. OCBA</th>
<th>OCBA vs. ERN</th>
<th>ERN vs. EER1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{06}$</td>
<td>13 7 0</td>
<td>9 11 0</td>
</tr>
<tr>
<td>$\sigma_{12}$</td>
<td>15 5 0</td>
<td>10 10 0</td>
</tr>
<tr>
<td>$\sigma_{18}$</td>
<td>13 7 0</td>
<td>11 9 0</td>
</tr>
<tr>
<td>$\sigma_{24}$</td>
<td>13 7 0</td>
<td>10 10 0</td>
</tr>
<tr>
<td>$\sigma_{30}$</td>
<td>14 6 0</td>
<td>11 9 0</td>
</tr>
<tr>
<td>Total</td>
<td>68 32 0</td>
<td>51 49 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EER1 vs. EER2</th>
<th>EER2 vs. ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{06}$</td>
<td>6 14 0</td>
</tr>
<tr>
<td>$\sigma_{12}$</td>
<td>6 14 0</td>
</tr>
<tr>
<td>$\sigma_{18}$</td>
<td>8 12 0</td>
</tr>
<tr>
<td>$\sigma_{24}$</td>
<td>9 11 0</td>
</tr>
<tr>
<td>$\sigma_{30}$</td>
<td>10 10 0</td>
</tr>
<tr>
<td>Total</td>
<td>39 61 0</td>
</tr>
</tbody>
</table>
Table 5.3: Average population statistics for the PSO algorithms over all the benchmark functions, levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>ERGC</th>
<th>OCBA</th>
<th>ERN</th>
<th>EER1</th>
<th>EER2</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lifetime</strong>+</td>
<td>72.47</td>
<td>90.47</td>
<td>88.89</td>
<td>93.58</td>
<td><strong>94.63</strong></td>
<td>63.97</td>
</tr>
<tr>
<td><strong>Binary Deception</strong>−</td>
<td>78.53</td>
<td></td>
<td>86.22</td>
<td>88.50</td>
<td>88.05</td>
<td>92.74</td>
</tr>
<tr>
<td><strong>Ranked Deception</strong>−</td>
<td>20.90</td>
<td></td>
<td>23.46</td>
<td>30.78</td>
<td>32.40</td>
<td>36.34</td>
</tr>
<tr>
<td><strong>Regular Operation</strong>+</td>
<td>53.04</td>
<td>67.60</td>
<td>60.87</td>
<td>69.78</td>
<td><strong>70.28</strong></td>
<td>61.67</td>
</tr>
<tr>
<td><strong>Updates</strong>+</td>
<td>27.31</td>
<td>23.12</td>
<td>25.10</td>
<td>24.36</td>
<td><strong>27.33</strong></td>
<td>16.06</td>
</tr>
<tr>
<td><strong>Discards</strong>−</td>
<td>72.69</td>
<td>76.88</td>
<td>74.90</td>
<td>75.64</td>
<td><strong>72.67</strong></td>
<td>83.94</td>
</tr>
<tr>
<td><strong>Blindness</strong>−</td>
<td>44.75</td>
<td>27.33</td>
<td>33.53</td>
<td>22.36</td>
<td><strong>19.01</strong></td>
<td>36.13</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>89.17</td>
<td>71.75</td>
<td>81.19</td>
<td>68.37</td>
<td>55.60</td>
<td>91.85</td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td>10.83</td>
<td>28.25</td>
<td>18.81</td>
<td>31.63</td>
<td>44.40</td>
<td>8.15</td>
</tr>
<tr>
<td><strong>Disorientation</strong>−</td>
<td>2.21</td>
<td>5.07</td>
<td>5.60</td>
<td>7.86</td>
<td>10.71</td>
<td><strong>2.20</strong></td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>12.42</td>
<td>12.00</td>
<td>24.61</td>
<td>15.47</td>
<td>17.77</td>
<td>10.53</td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td>87.58</td>
<td>88.00</td>
<td>75.39</td>
<td>84.53</td>
<td>82.23</td>
<td>89.47</td>
</tr>
</tbody>
</table>

Larger+ or smaller− is better. **Bold** value is best. Units in percentages (%).
ception (92.74%) and blindness (36.13%) from which particles still suffer throughout the search.

The extension we designed for PSO-ER, namely PSO-EER, aims to further reduce the proportions of deception and blindness in PSO-ER by also allocating evaluations to the personal best solutions. Specifically, we evaluated two settings of PSO-EER that we refer to as PSO-EER1 and PSO-EER2, in which one and two evaluations (respectively) are allocated to the personal best solutions and the remaining evaluations are allocated between the current solutions. Both settings reduced the proportions of blindness and deception, thereby finding better solutions than PSO-ER. However, PSO-EER1 found significantly better solutions than PSO-EER2 because the better estimated objective values of its current solutions made disorientation less frequent than in PSO-EER2 in spite of an increase of blindness. These results show that the quality of the results can deteriorate significantly due to disorientation, as shown in PSO-EER2 where disorientation is present on average in only 10.71% of the iterations.

PSO-OCBA is the resampling-based PSO algorithm that produces the best quality of results. PSO-OCBA allocates a computational budget between the most relevant solutions in the swarm, either current or personal best solutions, based on their z-scores to favour the allocation to those solutions whose objective values have the lower means and higher variances. Moreover, since the allocation is performed sequentially, it strives to better estimate the objective values of those solutions which are more likely to be the best solutions. The design of PSO-OCBA was originally intended to maximize the probability of particles to correctly select their neighborhood best solutions, that is, to minimize the proportions of binary deception in the swarm. However, its design not only reduces the binary deception in the swarm, but also reduces the proportions of blindness and disorientation along with it. Furthermore, the particles of PSO-OCBA select neighborhood best solutions that rank higher than those of the other algorithms even if the true best solutions are not selected.
The closest alternative to PSO-OCBA is PSO-ERN, which is a simpler and computationally cheaper algorithm that manages to produce similar results in about half of the benchmark functions and worse results in the remaining benchmarks. PSO-ERN allocates the computational budget of additional evaluations between the estimated top-$N$ solutions at once and not sequentially, thereby saving some computational time. Moreover, its particles update their personal best solutions with new solutions if these are better or more accurate, thus encouraging frequent updates. While PSO-ERN does not outperform PSO-OCBA, the quality of its results is better than PSO-ER and better than both settings of PSO-EER, thus remarking the importance of having accurate objective values on at least a few good solutions.

The population statistics have shown again to be useful to understand the underlying reasons for the quality of the results of resampling-based PSO algorithms, and also helped to design the PSO-EER algorithm that finds better solutions than PSO-ER by reducing blindness and deception.

Next Chapter

The next chapter will present a study on the population statistics for hybrid PSO algorithms that combine both single-evaluation and resampling methods to address optimization problems subject to noise. The population statistics for single-evaluation and resampling-based PSO algorithms will also be compared to determine which approach (if any) produces better results.

5.A Population Statistics

The following figures correspond to the population statistics on each benchmark function for the algorithms utilized in the design of experiments from this chapter (Section 5.3, page 150).
Figure 5.4: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The algorithms are abbreviated as (g) PSO-ERGC, (e) PSO-ER, (1) PSO-EER1, (2) PSO-EER2, (n) PSO-ERN, and (o) PSO-OCBA. The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions. The boxplots in $F_{06}$ corresponding to (o) and (1) at $\sigma_{06}$ have been upscaled as well as those corresponding to (g) in order to improve their presentation without changing their relative ordering.
Figure 5.5: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The algorithms are abbreviated as (g) PSO-ERGC, (e) PSO-ER, (1) PSO-EER1, (2) PSO-EER2, (n) PSO-ERN, and (o) PSO-OCBA. The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
Figure 5.6: Binary Deception. The barplots represent the average proportions of iterations (left axis) at which a particle is deceived by its neighbors for each algorithm (bottom axis) on the benchmark functions subject to levels of noise \( \sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\} \) (bars coloured from light to dark gray). The algorithms are abbreviated as (g) PSO-ERGC, (o) PSO-OCBA, (n) PSO-ERN, (e1) PSO-EER1, (e2) PSO-EER2, and (e) PSO-ER. Particles in PSO-ERGC do not suffer from deception. Smaller proportions are better.
### 5.A. POPULATION STATISTICS

#### Figure 5.7: Ranked Deception

The barplots represent the average percentile rank (left axis) based on the true objective values of the selected neighborhood best solutions with respect to the true objective values of the personal best solutions in the swarm for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (g) PSO-ERGC, (o) PSO-OCBA, (n) PSO-ERN, (e1) PSO-EER1, (e2) PSO-EER2, and (e) PSO-ER. Particles from PSO-ERGC do not suffer from deception and hence their neighborhood best solutions are always ranked best. Lower percentile ranks are better.
Figure 5.8: Regular Operation, Blindness and Disorientation. The stacked barplots represent the average proportions (left axis) of regular operation (dark gray), blindness (medium gray) and disorientation (light gray) experienced by a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (g) PSO-ERGC, (o) PSO-OCBA, (n) PSO-ERN, (e1) PSO-EER1, (e2) PSO-EER2, and (e) PSO-ER. Larger proportions of regular operation and smaller proportions of blindness and disorientation are better.
Figure 5.9: Regular Updates and Discards. The stacked barplots represent the average proportions (left axis) of regular updates (dark gray) and discards (light gray) experienced by a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (g) PSO-ERGC, (o) PSO-OCBA, (n) PSO-ERN, (e1) PSO-EER1, (e2) PSO-EER2, and (e) PSO-ER. Larger proportions of regular updates and smaller proportions of regular discards are better.
Figure 5.10: Causes of Blindness. The stacked barplots represent the average proportions (left axis) of blindness caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (g) PSO-ERGC, (o) PSO-OCBA, (n) PSO-ERN, (e1) PSO-EER1, (e2) PSO-EER2, and (e) PSO-ER.
Figure 5.11: Causes of Disorientation. The stacked barplots represent the average proportions (left axis) of disorientation caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (g) PSO-ERGC, (o) PSO-OCBA, (n) PSO-ERN, (e1) PSO-EER1, (e2) PSO-EER2, and (e) PSO-ER.
Figure 5.12: Lifetime. The barplots represent the proportions of average lifetime (left axis) of a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (g) PSO-ERGC, (o) PSO-OCBA, (n) PSO-ERN, (e1) PSO-EER1, (e2) PSO-EER2, and (e) PSO-ER. A longer lifetime is better when the swarm does not converge to the global optimum.
Chapter 6

Hybrid Methods

This chapter develops the first hybrid PSO algorithms based on the best single-evaluation method found in Chapter 4, namely PSO-AN, and the different resampling methods studied in Chapter 5. The population statistics for the new algorithms are studied on the set of benchmark functions subject to noise, and compared against the population statistics for the purely single-evaluation and resampling-based PSO algorithms. Furthermore, the effect of different computational budget allocations is explored on the population statistics for PSO-AN and other hybrid variants based on PSO-AN and PSO-EER.

This chapter is structured as follows. Section 6.1 introduces this chapter. Section 6.2 presents the design of new hybrid PSO algorithms. Section 6.3 describes the design of experiments. Section 6.4 presents the results and discussions. Finally, Section 6.5 ends this chapter with a summary.

6.1 Introduction

Noise mitigation mechanisms for PSO can be classified into two different approaches according to their allocation of the computational budget of function evaluations. On the one hand, there is single-evaluation methods
that do not re-evaluate any solution, but instead settle with their objective values being estimated by a single evaluation as the PSO algorithm dictates. On the other hand, there is resampling methods which make use of the computational budget of function evaluations to perform multiple evaluations to the solutions in order to better estimate their true objective values. Hence, both approaches provide opposite tradeoffs regarding the exploration of new solutions and improving the accuracy of existing solutions.

The best single-evaluation method for PSO, amongst the ones we have studied, is PSO-AN. The goal of PSO-AN is to blur the effect of noise on the selection of the neighborhood best solution in order to reduce deception. Each particle in PSO-AN computes the centroid from all the personal best solutions in the neighborhood, and the centroid is selected as the neighborhood best solution. In doing so, the centroid solution will generally have a better quality than any other solution a particle would have selected based on the very inaccurately estimated objective values. Consequently, such a better neighborhood best solution will have particles partially attracted towards better regions of the search space that will improve the ranked deception and, ultimately, the quality of the results.

The best resampling method for PSO, amongst the ones we have studied, is PSO-OCBA. The goal of PSO-OCBA is to asymptotically minimize the binary deception in the swarm. First, PSO-OCBA equally allocates the computational budget between the particles in order to better estimate their current solutions. Then, an additional computational budget is sequentially allocated to the most important (current or personal best) solutions, that is, those solutions whose estimated objective values have the lower means and larger variances. After each allocation of a group of evaluations, the means and variances of the objective values of the recipient solutions will be re-estimated and taken into consideration before allocating the next group of evaluations. Hence, after each allocation, the estimated best solutions will be reaffirmed or discarded to be selected as
neighborhood best solutions. At the end, when the computational budget is exhausted, the solution with the estimated best objective value will potentially be the true neighborhood best solution.

While single-evaluation and resampling-based PSO algorithms are completely different from each other, this does not prevent us from merging them and deriving a hybrid PSO algorithm. On the contrary, such a hybrid algorithm could potentially deliver better results than both their algorithms independently from each other. The most compelling reason to support our argument is that PSO-OCBA still fails most of the time to correctly select the true best solution and, even when it has a superior quality of results amongst resampling-based algorithms, selecting the centroid of the estimated best solutions will likely have a better objective value than that of the estimated best solution. Therefore, we expect that a hybrid between PSO-AN and PSO-OCBA will reduce the proportions of (at least ranked) deception and thereby improve the quality of the solutions. More importantly, since this is not an exclusive case for PSO-OCBA, we can also hybridize PSO-EER with PSO-AN and possibly improve the quality of the results.

Chapter Goals

The overall goal of this chapter is to create hybrid algorithms between PSO-AN and different resampling-based methods as an attempt to further improve the quality of the results found with PSO on optimization problems subject to noise. Specifically, the following objectives will be addressed:

- Contrast the population statistics for hybrid PSO algorithms with their respective single-evaluation and resampling-based PSO algorithms to determine whether hybrid methods find better solutions.

- Contrast the population statistics for hybrid PSO algorithms with
PSO-ERGC in order to determine the extent to which efforts should be made to correctly select the neighborhood best solutions.

- Contrast the population statistics for single-evaluation and resampling-based PSO algorithms in order to determine which approach finds better solutions.

6.2 Hybrid PSO Algorithms

Thus far, we have studied the population statistics for PSO utilizing single-evaluation and resampling methods, both of which balance the accuracy tradeoff differently. On the one hand, PSO with single-evaluation methods perform more iterations at the cost of having the objective values of the solutions very inaccurately estimated. On the other hand, PSO with resampling methods sacrifice the number of iterations over improving the accuracy of the solutions. In each method, the algorithms that have shown the best quality of results are PSO-AN and PSO-OCBA. Hence, in order to further improve the performance of PSO, we incorporate into resampling-based PSO algorithms the idea of using, as neighborhood best solutions, the centroid computed from a number of estimated best solutions.

The centroid solution is utilized in the single-evaluation PSO-AN, where particles compute the centroid from the personal best solutions in the neighborhood, and use that to create a new solution that serves as the neighborhood best. As such, particles mitigate the effect of noise that causes deception by creating a potentially better solution that does not need to be evaluated for the algorithm to work. However, have in mind that we do evaluate the solution for the purpose of estimating its quality with respect to the swarm and thus compute the population statistics of deception.

The centroid solution was proposed in Chapter 4 as the average neighborhood best solution in PSO-AN, and the quality of results obtained was significantly better than that of any other single-evaluation algorithm con-
6.2. HYBRID PSO ALGORITHMS

Considered. The main reason for such an improvement at medium and high levels of noise is that particles were more often attracted towards better neighborhood solutions as was represented by the better ranked deception. However, considering a hybrid method, the objective values of the solutions will be better estimated thanks to the resampling method. Therefore, a better centroid solution can be created from a small number of the estimated best solutions in the swarm instead of all the solutions therein. Figure 6.1 describes the creation and selection of the centroid solution at iteration $t$ for any given PSO algorithm, where $\mathcal{C}_i$ is a subset of $\mathcal{N}_i$ that contains the particles whose personal best solutions are utilized to compute the centroid solution.

The hybrid PSO algorithms that we propose consist of merging the best resampling-based PSO algorithms, namely PSO-EER1, PSO-ERN and PSO-OCBA, with the single-evaluation PSO-AN. As such, the hybrid PSO algorithms differ from their respective resampling-based counterparts only on the selection of the neighborhood best solution, and from PSO-AN on the better estimated objective values of the solutions and the fewer iterations performed due to resampling. The hybrid PSO-EER is presented in Figure 6.2, the hybrid PSO-ERN is presented in Figure 6.3, and the hybrid PSO-OCBA is presented in Figure 6.4, where $b_\alpha$ is the individual computational budget allocated from $B^t_\alpha$ to each particle, $\hat{f}(x^t_i)$ is a noisy evaluation of the current solution $x^t_i$, $\mathcal{X}^t_i$ is the set of evaluations performed on the current solution $x^t_i$, and $\mathcal{Y}^t_i$ is the set of evaluations performed on the personal best solution $y^t_i$. In hybrid PSO-EER, $b_\beta$ is the individual computational budget allocated from $B^t_\beta$ to each particle. In hybrid PSO-ERN, $b_i$ is the

```plaintext
foreach particle $i$ in swarm $S$ do
  $\hat{y}^t_i \leftarrow \frac{1}{|\mathcal{C}_i|} \sum_{j \in \mathcal{C}_i} y^t_j$ where $\mathcal{C}_i \subseteq \mathcal{N}_i$
```

Figure 6.1: The Centroid Solution.
foreach iteration $t$ do
    foreach particle $i$ in swarm $S$ do
        $\mathcal{X}^t_i \leftarrow \bigcup_{\alpha} b \hat{f}(x^t_i)$
        $\mathcal{Y}^{t-1}_i \leftarrow \mathcal{Y}^{t-1}_i \cup \left( \bigcup_{\beta} b \hat{f}(y^{t-1}_i) \right)$
        if mean($\mathcal{X}^t_i$) $<$ mean($\mathcal{Y}^{t-1}_i$) then
            $y^t_i \leftarrow x^t_i$, $\mathcal{Y}^t_i \leftarrow \mathcal{X}^t_i$
        else $y^t_i \leftarrow y^{t-1}_i$, $\mathcal{Y}^t_i \leftarrow \mathcal{Y}^{t-1}_i$
    foreach particle $i$ in swarm $S$ do
        $\hat{y}^t_i \leftarrow \frac{1}{|C_i|} \sum_{j \in C_i} y^t_j$ where $C_i \subseteq N_i$
    foreach particle $i$ in swarm $S$ do
        update $v^{t+1}_i$ and $x^{t+1}_i$ using (2.3) and (2.4)

Figure 6.2: Hybrid PSO-EER.

individual computational budget allocated from $B^t_\beta$ to particle $i$, and $N$ is the number of current solutions whose objective values will be more accurately estimated. In hybrid PSO-OCBA, the computational budget $B^t_\beta$ is allocated sequentially in groups of $b_\Delta$ evaluations.

6.3 Design of Experiments

The algorithms for which we want to compute the population statistics are the hybrid PSO algorithms made up from different resampling methods and using the centroid solution as neighborhood best solution. Specifically, considering the results from Chapter 5, we will focus on the hybrids of PSO-EER1, PSO-ERN and PSO-OCBA, all of which compute the centroid solution from the estimated best five (personal best) solutions in the swarm. Hereinafter, we refer to these hybrid algorithms as PSO-EER1$^5$, PSO-ERN$^5$ and PSO-OCBA$^5$, where $^5$ indicates the number of solutions from which the centroid solution is computed. In addition, we want to
6.3. DESIGN OF EXPERIMENTS

foreach iteration $t$ do
    foreach particle $i$ in swarm $S$ do
        $\mathcal{X}_i^t \leftarrow \bigcup_{j=1}^{b_\alpha} \hat{f}(x_i^t)$
    endforeach
    foreach particle $i$ in $\text{top}(N, S)$ do
        $\mathcal{X}_i^t \leftarrow \mathcal{X}_i^t \cup \left( \bigcup_{j=1}^{b_\alpha} \hat{f}(x_i^t) \right)$
    endforeach
    foreach particle $i$ in swarm $S$ do
        if mean($\mathcal{X}_i^t$) < mean($\mathcal{Y}_i^{t-1}$) or $|\mathcal{X}_i^t| > |\mathcal{Y}_i^{t-1}|$
            then $y_i^t \leftarrow x_i^t$, $\mathcal{Y}_i^t \leftarrow \mathcal{X}_i^t$
        else $y_i^t \leftarrow y_i^{t-1}$, $\mathcal{Y}_i^t \leftarrow \mathcal{Y}_i^{t-1}$
    endforeach
    foreach particle $i$ in swarm $S$ do
        $\hat{y}_i^t \leftarrow \frac{1}{|C_i|} \sum_{j \in C_i} y_j^t$ where $C_i \subseteq N_i$
    endforeach
    foreach particle $i$ in swarm $S$ do
        update $v_i^{t+1}$ and $x_i^{t+1}$ using (2.3) and (2.4)
    endforeach

Figure 6.3: Hybrid PSO-ERN

foreach iteration $t$ do
    foreach particle $i$ in swarm $S$ do
        $\mathcal{X}_i^t \leftarrow \bigcup_{j=1}^{b_\alpha} \hat{f}(x_i^t)$
    endforeach
    ocba($S, B_\beta, b_{\Delta}$)
    foreach particle $i$ in swarm $S$ do
        if mean($\mathcal{X}_i^t$) < mean($\mathcal{Y}_i^{t-1}$)
            then $y_i^t \leftarrow x_i^t$, $\mathcal{Y}_i^t \leftarrow \mathcal{X}_i^t$
        else $y_i^t \leftarrow y_i^{t-1}$, $\mathcal{Y}_i^t \leftarrow \mathcal{Y}_i^{t-1}$
    endforeach
    foreach particle $i$ in swarm $S$ do
        $\hat{y}_i^t \leftarrow \frac{1}{|C_i|} \sum_{j \in C_i} y_j^t$ where $C_i \subseteq N_i$
    endforeach
    foreach particle $i$ in swarm $S$ do
        update $v_i^{t+1}$ and $x_i^{t+1}$ using (2.3) and (2.4)
    endforeach

Figure 6.4: Hybrid PSO-OCBA.
compare the population statistics for hybrid PSO algorithms against those for single-evaluation and resampling-based algorithms to assess their respective differences. Beside these comparisons, the population statistics between resampling-based and single-evaluation PSO algorithms also need to be contrasted to determine empirically which approach finds better results on optimization problems subject to noise. Thus, the population statistics for the following algorithms will be compared: PSO-EER1, PSO-ERN, PSO-OCBA, PSO-EER1\textsuperscript{5}, PSO-ERN\textsuperscript{5}, PSO-OCBA\textsuperscript{5}, PSO-AN and PSO-ERGC.

The swarms from each algorithm are made up of 50 particles whose inertia $w$ and acceleration coefficients $c_1$ and $c_2$ are set according to the guidelines in [145], and their neighborhoods are defined by the star topology. Particles limit their velocities utilizing the hyperbolic tangent function to reduce the sensitivity of setting a maximum velocity [48], which is set according to the limits $x_{\text{min}}$ and $x_{\text{max}}$ of the optimization problem. The computational budget available to each swarm is set to 30 000 function evaluations, thereby allowing the swarms of PSO-AN to perform 600 iterations, whereas resampling-based and hybrid PSO algorithms perform 100 iterations. As such, the population statistics for each swarm of PSO-AN are made up from $50 \times 600 \times 50 = 1 500 000$ observations and those for the remaining algorithms are made up from $50 \times 100 \times 50 = 250 000$ observations. In general, the settings are the same as those in Chapters 3, 4 and 5, but with a different set of algorithms. The complete list of parameter settings is presented in Table 6.1.

## 6.4 Results and Discussions

The results and discussions are structured as follows. Section 6.4.1 presents and describes the population statistics for the algorithms. Section 6.4.2 compares resampling-based and hybrid PSO algorithms. Section 6.4.3 compares the hybrid PSO algorithms. Sections 6.4.4 and 6.4.5 compare the hybrid PSO algorithms against the resampling-based PSO-OCBA and PSO-ERGC,
Table 6.1: Parameter settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent runs</td>
<td>50 with 30,000 function evaluations</td>
</tr>
<tr>
<td>Number of particles</td>
<td>50 in $\mathbb{R}^{1000}$ with star topology</td>
</tr>
<tr>
<td>Acceleration</td>
<td>Static with $c_1 = c_2 = 1.49618$</td>
</tr>
<tr>
<td>Inertia</td>
<td>Static with $w = 0.729844$</td>
</tr>
<tr>
<td>Maximum velocity</td>
<td>$0.25 \cdot</td>
</tr>
<tr>
<td>Velocity clamping</td>
<td>$v_{ij}^t = v_{\text{max}} \cdot \tanh \left( \frac{v_{ij}^t}{v_{\text{max}}} \right)$</td>
</tr>
<tr>
<td>Severity of noise</td>
<td>$\sigma \in {0.06, 0.12, 0.18, 0.24, 0.30}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO-EER1, PSO-EER1$^5$</td>
<td>$b_\alpha = 5$, $b_\beta = 1$</td>
</tr>
<tr>
<td>PSO-OCBA, PSO-OCBA$^5$</td>
<td>$b_\alpha = 5$, $b_\Delta = 5$</td>
</tr>
<tr>
<td>PSO-ERN, PSO-ERN$^5$</td>
<td>$b_\alpha = 5$, $N = 2$, $b_{(1)} = b_{(2)} = 25$</td>
</tr>
</tbody>
</table>

respectively. Section 6.4.6 compares the single-evaluation PSO-AN against resampling-based and hybrid PSO algorithms. Lastly, Section 6.4.7 presents four new hybrid PSO algorithms based on PSO-AN and PSO-EER that allocate differently the computational budget of evaluations, and the discussions of their respective population statistics are presented therein after another design of experiments.

### 6.4.1 Population Statistics

Our discussions are based mostly on the average population statistics presented in Table 6.2, whose trends are also reflected on the population statistics on $F_{13}$ presented in Figure 6.5. The population statistics on the remaining benchmark functions are presented in Appendix 6.A (page 207). The population statistics on $F_{13}$ were selected because the trends therein can be more easily distinguished than in the other benchmark functions. The algorithms in Figure 6.5 are abbreviated as (e) PSO-EER1, (n) PSO-ERN,
(o) PSO-OCBA, (E) PSO-EER$^5$, (N) PSO-ERN$^5$, (O) PSO-OCBA$^5$, (a) PSO-AN and (g) PSO-ERGC, and the population statistics are presented as follows.

Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) on $F_{13}$ subject to the different levels of noise (top axis). The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems and therefore lower objective values indicate better solutions.

Binary Deception. The barplots represent the average proportion of iterations (left axis) in which a particle for each algorithm (bottom axis) is deceived by its neighbors on $F_{13}$ subject to the different levels of noise (bars coloured from light to dark gray). Smaller proportions are better. Particles from PSO-ERGC do not suffer from binary deception as they correctly select their true neighborhood best solutions.

Ranked Deception. The barplots represent the average percentile rank (left axis) of the neighborhood best solution chosen by a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars coloured from light to dark gray). Smaller ranks are better. Particles from PSO-ERGC do not suffer from ranked deception as they correctly select the neighborhood best solutions.

Regular Operation, Blindness and Disorientation. The stacked barplots represent the average proportions (left axis) of regular operation (dark gray), blindness (medium gray) and disorientation (light gray) experienced by a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars from left to right). Larger proportions of regular operations and smaller proportions of blindness and disorientation are better.

Regular Updates and Discards. The stacked barplots represent the average proportions (left axis) of regular updates (dark gray) and discards (light gray) experienced by a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars from left to right). Larger
proportions of regular updates and smaller proportions of discards are better.

*Causes of Blindness.* The stacked barplots represent the average proportions of blindness (left axis) caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars from left to right).

*Causes of Disorientation.* The stacked barplots represent the average proportions of disorientation (left axis) caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars from left to right).

*Effect of Disorientation and Blindness.* The stacked barplots represent the normalized average magnitudes (left axis) of deterioration caused by disorientation (dark gray) and hypothetical improvement missed by blindness (light gray) on $F_{13}$ subject to the different levels of noise (bars from left to right).

*Lifetime.* The barplots represent the normalized average lifetime (left axis) of a particle for each algorithm (bottom axis) on $F_{13}$ subject to the different levels of noise (bars coloured from light to dark gray). A longer lifetime is better when the swarm does not converge to the global optimum solution.

The population statistics in Table 6.2 and Figure 6.5 are discussed in the following sections and are referred to by their names in *italics* the first time that they are brought up for discussion.

### 6.4.2 Resampling-Based and Hybrid Algorithms

The quality of results obtained with the hybrid PSO algorithms is significantly better than that of their purely resampling-based counterparts, and that is the general case as shown in Table 6.3. The underlying reasons to such an improved quality of results are explained utilizing the remaining population statistics as follows.

In PSO-OCBA, the *binary deception* increases by about 5% with respect
Table 6.2: Average population statistics for the PSO algorithms over all benchmark functions, levels of noise, independent runs, iterations and particles. Larger or smaller is better.

<table>
<thead>
<tr>
<th>Benchmark Function</th>
<th>Environment</th>
<th>Memory</th>
<th>Hybrid</th>
<th>Resampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>LifeTime</td>
<td>84.3</td>
<td>81.17</td>
<td>74.50</td>
<td>98.50</td>
</tr>
<tr>
<td>Binary Deception</td>
<td>74.67</td>
<td>75.39</td>
<td>88.39</td>
<td>84.36</td>
</tr>
<tr>
<td>Ranked Deception</td>
<td>74.67</td>
<td>75.39</td>
<td>88.39</td>
<td>84.36</td>
</tr>
<tr>
<td>Regular Operation</td>
<td>74.67</td>
<td>75.39</td>
<td>88.39</td>
<td>84.36</td>
</tr>
<tr>
<td>Larger</td>
<td>99.00</td>
<td>89.17</td>
<td>87.58</td>
<td>87.58</td>
</tr>
</tbody>
</table>

Larger or smaller is better. Bold value is best in group. Bold value is best overall. Units in percentages (%).
Figure 6.5: Population statistics on benchmark function $F_{13}$. The presentation details of each statistic are described in Section 6.4.1 (page 185).
Table 6.3: Summary of statistical tests on the quality of the results between resampling-based and hybrid PSO algorithms.

<table>
<thead>
<tr>
<th></th>
<th>EER1&lt;sup&gt;5&lt;/sup&gt; vs. EER1</th>
<th>ERN&lt;sup&gt;5&lt;/sup&gt; vs. ERN</th>
<th>OCBA&lt;sup&gt;5&lt;/sup&gt; vs. OCBA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>− = +</td>
<td>− = +</td>
<td>− = +</td>
</tr>
<tr>
<td>σ&lt;sub&gt;06&lt;/sub&gt;</td>
<td>17 3 0</td>
<td>15 4 1</td>
<td>17 2 1</td>
</tr>
<tr>
<td>σ&lt;sub&gt;12&lt;/sub&gt;</td>
<td>16 4 0</td>
<td>15 5 0</td>
<td>14 4 2</td>
</tr>
<tr>
<td>σ&lt;sub&gt;18&lt;/sub&gt;</td>
<td>16 4 0</td>
<td>13 7 0</td>
<td>15 4 1</td>
</tr>
<tr>
<td>σ&lt;sub&gt;24&lt;/sub&gt;</td>
<td>17 3 0</td>
<td>15 5 0</td>
<td>15 5 0</td>
</tr>
<tr>
<td>σ&lt;sub&gt;30&lt;/sub&gt;</td>
<td>17 3 0</td>
<td>13 7 0</td>
<td>14 6 0</td>
</tr>
<tr>
<td>Total</td>
<td>83 17 0</td>
<td>71 28 1</td>
<td>75 21 4</td>
</tr>
</tbody>
</table>

To PSO-OCBA, thus showing that the centroid solution is generally worse than the true neighborhood best solution. However, the ranked deception reduces by about 5\% in PSO-OCBA<sup>5</sup>, which is a more important achievement in the long term because the centroid solution will be better more often than the solutions selected by the particles in PSO-OCBA whenever they fail to correctly select the true neighborhood best solution.

In PSO-EER1<sup>5</sup>, the binary deception decreases by about 1\% with respect to PSO-EER1, which is a negligible improvement over just selecting the estimated neighborhood best solution. However, the ranked deception significantly decreases by about 10\% in PSO-EER1<sup>5</sup>, thus showing that the centroid solution is better more often than the estimated neighborhood best solution. Even when PSO-EER1<sup>5</sup> decreased the ranked deception by a larger proportion than PSO-OCBA<sup>5</sup> did (both with respect to their resampling-based counterparts), PSO-OCBA<sup>5</sup> still has a smaller ranked deception (15.99\% vs. 21.21\%) because its centroid solution is made up from better solutions.

In PSO-ERN<sup>5</sup>, differently, the binary and ranked deception each increases by more than 3\% and 4\% with respect to PSO-ERN. Such a difference is expected because PSO-ERN allocates its additional computational budget
between the estimated best two solutions, leaving many other solutions without additional evaluations. Hence, the centroid solution will be made up utilizing less accurately estimated solutions which may not even include the two solutions whose objective values are the most accurate. A workaround to this issue would be to compute the centroid solution from the solutions with the most evaluations, which will not necessarily be the solutions with estimated best objective values. Nonetheless, the quality of the results of PSO-ERN is still much better than that of PSO-ERN, even with such a deterioration in binary and ranked deception.

The proportions of blindness significantly increase by 12% in both PSO-EER1 and PSO-OCBA, and by 4% in PSO-ERN with respect to their purely resampling-based counterparts, even when the only difference between them is the selection of the centroid solution. Nonetheless, we have seen before that selecting better neighborhood best solutions increases the presence of blindness in the swarm. Specifically, the most compelling case is that of PSO-ERGC and PSO-ER in Section (page 161), where blindness in PSO-ERGC increased by over 8% on average with respect to PSO-ER (see Table 5.3 page 165). However, we acknowledge that further research is required to find the underlying reasons to such an increment of blindness. On the other hand, the proportions of disorientation have negligible reductions in the hybrid algorithms with respect to their purely resampling-based counterparts, a reduction that we attribute to the larger proportions of blindness given their inverse correlation described in Section 3.4.2 (page 75). Also, as a result from larger proportions of blindness, the proportions of regular operations are reduced in the hybrid algorithms.

The remaining population statistics show that the quality of the centroid solution is generally better, which helps to increase the proportions of regular updates in PSO-EER1 and PSO-OCBA by more than 8% and 11%, and just by 2% in PSO-ERN with respect to their purely resampling-based counterparts. The causes of blindness and disorientation have mild variations of around 3% on both conditions caused by memory. Also, the
Table 6.4: Summary of statistical tests on the quality of the results between hybrid PSO algorithms

<table>
<thead>
<tr>
<th></th>
<th>OCBA(^5) vs. EER1(^5)</th>
<th>EER1(^5) vs. ERN(^5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>−   =   +</td>
<td>−   =   +</td>
</tr>
<tr>
<td>(\sigma_{06})</td>
<td>5   15  0</td>
<td>11  9  0</td>
</tr>
<tr>
<td>(\sigma_{12})</td>
<td>8   11  1</td>
<td>13  7  0</td>
</tr>
<tr>
<td>(\sigma_{18})</td>
<td>6   14  0</td>
<td>13  7  0</td>
</tr>
<tr>
<td>(\sigma_{24})</td>
<td>7   13  0</td>
<td>12  8  0</td>
</tr>
<tr>
<td>(\sigma_{30})</td>
<td>10  10  0</td>
<td>13  7  0</td>
</tr>
<tr>
<td>Total</td>
<td>36  63  1</td>
<td>62  38  0</td>
</tr>
</tbody>
</table>

The lifetime of the swarms shows mild variations of up to 2%.

### 6.4.3 Hybrid Algorithms

The quality of results shows that PSO-OCBA\(^5\) finds better solutions than PSO-EER1\(^5\), and PSO-EER1\(^5\) finds better solutions than PSO-ERN\(^5\). This ranking is further supported by the transitive relation found in the summary of statistical tests in Table 6.4. The underlying reason to such a ranking lies on the computation of the centroid solution, which reflects on the binary and ranked deception statistics as we have explained in the previous section. Additionally, considering the negligible differences between the proportions of regular operations in the hybrid algorithms, the proportions of regular updates and regular discards alone indicate the superior performance of PSO-EER1\(^5\) and PSO-OCBA\(^5\) over PSO-ERN\(^5\). As for the causes of blindness and disorientation, the larger proportions of blindness and disorientation by memory in PSO-ERN\(^5\) show that its personal best solutions are significantly less accurately estimated than the personal best solutions of PSO-EER1\(^5\) and PSO-OCBA\(^5\), which is another detrimental characteristic that supports the worse quality of results from PSO-ERN\(^5\).
6.4. RESULTS AND DISCUSSIONS

Table 6.5: Summary of statistical tests on the quality of the results between hybrid PSO algorithms and PSO-OCBA

<table>
<thead>
<tr>
<th></th>
<th>EER\textsuperscript{15} vs. OCBA</th>
<th>ERN\textsuperscript{5} vs. OCBA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{06}$</td>
<td>16 1 3</td>
<td>15 1 4</td>
</tr>
<tr>
<td>$\sigma_{12}$</td>
<td>14 4 2</td>
<td>10 7 3</td>
</tr>
<tr>
<td>$\sigma_{18}$</td>
<td>13 4 3</td>
<td>10 6 4</td>
</tr>
<tr>
<td>$\sigma_{24}$</td>
<td>13 5 2</td>
<td>11 5 4</td>
</tr>
<tr>
<td>$\sigma_{30}$</td>
<td>12 6 2</td>
<td>10 2 8</td>
</tr>
<tr>
<td>Total</td>
<td>68 20 12</td>
<td>56 21 23</td>
</tr>
</tbody>
</table>

6.4.4 Hybrid Algorithms and PSO-OCBA

PSO-EER\textsuperscript{15} and PSO-ERN\textsuperscript{5} are hybrid algorithms whose operation is much simpler and less computationally expensive than that of the purely resampling-based PSO-OCBA, and yet both hybrids still find significantly better solutions in the majority of the benchmark functions as shown in Table 6.5. PSO-EER\textsuperscript{15} and PSO-ERN\textsuperscript{5} are the first algorithms in the literature to outperform the resampling-based PSO-OCBA in so many benchmark functions, therefore questioning the importance of the large efforts put towards correctly selecting the neighborhood best solutions in the swarms.

The comparison hereinafter is made between the population statistics for PSO-EER\textsuperscript{15} and PSO-OCBA. We focus on this comparison only because we have already compared PSO-OCBA\textsuperscript{5} and PSO-OCBA in Section 6.4.2 and PSO-ERN\textsuperscript{5} yields the worst quality of results amongst the hybrid algorithms. Different from the previous comparisons explaining the quality of the results based on the population statistics, the goal of this comparison will be to find out whether we can predict the superior quality of the results obtained with PSO-EER\textsuperscript{15} by just looking at the population statistics.

The lifetime of PSO-EER\textsuperscript{15} is about 5% longer than that of PSO-OCBA.
(95.19% vs. 90.47%), which is generally a positive indicator for a superior quality of results. However, the lifetime needs to be considered together with the proportions of regular operations, and especially the proportions of regular updates. In regular operations, the proportions of PSO-OCBA are larger by about 10% than those of PSO-EER1\(^5\) (67.60% vs. 57.74%). In regular updates, however, the proportions of PSO-OCBA are smaller by about 12% than those of PSO-EER1\(^5\) (23.12% vs. 35.71%). Given the conflicting statistics of regular operations and regular updates between PSO-EER1\(^5\) and PSO-OCBA, and considering that the regular updates are computed as proportions of the regular operations, we can solve the conflict by computing the proportions of regular updates with respect to the number of iterations as:

\[
0.3571 \times 57.74 = 20.62\% \text{ for PSO-EER1}^5, \quad \text{and} \quad 0.2312 \times 67.60 = 15.63\% \text{ for PSO-OCBA.}
\]

Therefore, on average, PSO-EER1\(^5\) not only has a longer lifetime, but it also has about 5% more iterations with regular updates, which supports the superior quality of the results of PSO-EER1\(^5\) over PSO-OCBA.

The binary deception for PSO-OCBA is smaller by about 9% than PSO-EER1\(^5\) (78.53% vs. 87.20%). However, the proportions of binary deception in both algorithms are still very large, which indicates that the algorithms seldom find the true neighborhood best solutions. Thus, the ranked deception provides more information about the swarm given that it measures the average ranking of whatever neighborhood best solution is selected. In ranked deception, the difference between PSO-OCBA and PSO-EER1\(^5\) is just about 1% favouring PSO-OCBA. Therefore, given such a negligible difference, the superior quality of results expected for PSO-EER1\(^5\) still holds thanks to its longer lifetime and larger proportions of global regular updates.

The proportions of blindness and disorientation, however, are about 7% and 3% smaller for PSO-OCBA than they are for PSO-EER1\(^5\) (27.33% vs. 34.65%, and 5.07% vs. 7.62%), thereby contradictorily suggesting that the quality of the results obtained with PSO-OCBA should be superior. Cer-
tainly, we cannot attribute the larger proportions of blindness to the selection of better neighborhood best solutions (as we have done before) because both algorithms have rather similar proportions of ranked deception, and we cannot dismiss the small differences in the proportions of disorientation because we have seen before the detrimental effects of disorientation on the quality of the results. Therefore, we can only advise to take the population statistics as very useful guidelines of performance having in mind that contradictions are still possible. These findings, far from discrediting the importance of the population statistics designed thus far to analyze the performance of PSO algorithms, encourage instead the creation of other population statistics to represent other variables throughout the search process that could be helpful to better estimate the quality of the results.

6.4.5 Hybrid Algorithms and PSO-ERGC

The previous comparison between hybrid algorithms and the resampling-based PSO-OCBA questioned the relevance of making costly efforts towards the correct selection of the neighborhood best solutions, especially considering that the efforts made by PSO-OCBA to such an end still fail in 78.53% of the iterations on average (see Table 6.2). More importantly, even if PSO-OCBA successfully managed to correctly select the true best solution at every iteration, that is, even if PSO-OCBA were as successful as PSO-ERGC, the hybrid algorithms would still find better solutions in most cases as shown in Table 6.6.

The population statistics for the hybrid algorithms, compared to those for PSO-ERGC, suggest their superior quality of results. Specifically, the hybrid algorithms have a longer lifetime, more regular operations divided into more regular updates (and hence fewer regular discards), and smaller proportions of blindness. These statistics compensate for the absence of binary and ranked deception in PSO-ERGC as well as for its smaller propor-
Table 6.6: Summary of statistical tests on the quality of the results between hybrid PSO algorithms and PSO-ERGC

<table>
<thead>
<tr>
<th>ERGC vs. OCBA</th>
<th>EER1 vs. ERGC</th>
<th>ERN vs. ERGC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{06}$</td>
<td>13 7 0</td>
<td>$\sigma_{06}$</td>
</tr>
<tr>
<td>$\sigma_{12}$</td>
<td>15 5 0</td>
<td>$\sigma_{12}$</td>
</tr>
<tr>
<td>$\sigma_{18}$</td>
<td>13 7 0</td>
<td>$\sigma_{18}$</td>
</tr>
<tr>
<td>$\sigma_{24}$</td>
<td>13 7 0</td>
<td>$\sigma_{24}$</td>
</tr>
<tr>
<td>$\sigma_{30}$</td>
<td>14 6 0</td>
<td>$\sigma_{30}$</td>
</tr>
<tr>
<td>Total</td>
<td>68 32 0</td>
<td>Total</td>
</tr>
</tbody>
</table>

6.4.6 Single-Evaluation against Resampling-Based and Hybrid Algorithms

In general, the population statistics for the single-evaluation PSO-AN are significantly worse than those for any other resampling-based and hybrid algorithm. Still, we find one exception in the population statistic of disorientation, where PSO-AN has the smallest proportions due to the presence of very large proportions of blindness; and other exceptions in the quality of the results across different benchmark functions (see Table 6.7). On the exceptions found with respect to the quality of the results, PSO-AN is better than the resampling-based algorithms in about half of the benchmark functions regardless of the level of noise, but it is worse in the remaining half with a few cases not being significantly different. Similarly, PSO-AN is better than the hybrid algorithms in about a half of the benchmark functions, but only at medium-to-high levels of noise. The benchmark functions on which PSO-AN has a better quality of results are those composed of the ackley and rastrigin functions, but also two of them are composed of rosenbrock (see Table 6.8). The remarkable differences
in the quality of the results can be found in Figures 6.7 and 6.8 located in Appendix 6.A (page 207), and these differences suggest that the most appropriate accuracy tradeoff is problem-dependent. Therefore, in the next section, we will explore the effect of utilizing different allocations of the computational budget on the population statistics and the quality of the results.

6.4.7 Budget Allocation: from PSO-AN to PSO-EER1

Besides the remarkable differences found in the quality of results between PSO-AN and the hybrid algorithms, the population statistics for PSO-AN in Table 6.2 show that the average lifetime of PSO-AN is 54.43%, which suggests that about half of the iterations performed are a waste of function evaluations which could have been spent better on re-evaluating every (current or personal best) solution at least once. In doing so, not only the objective values would have been better estimated, but also the proportions of blindness could have been significantly reduced and the regular operations increased, alongside other potential positive effects such as a longer lifetime and more regular updates in spite of a likely increase of disorientation. Furthermore, once the (current or personal best) solutions are better estimated, the centroid solution can be created utilizing only a few of the estimated best solutions instead of all of them, thereby reducing the binary and (mostly) the ranked deception in the swarm. With these expectations in mind, we performed an additional set of experiments utilizing PSO-AN and hybrid variants of PSO-AN that spend the computational budget differently.

Specifically, we designed the following four hybrid PSO-AN algorithms: PSO-AN_{11}, PSO-AN_{21}, PSO-AN_{31}, and PSO-AN_{41}, where the subscripts are in the form $xy$ and refer to the number of evaluations performed to the current ($x$) and to the personal best ($y$) solutions. As such, these hybrid variants cover the range of different computational budget alloca-
Table 6.7: Summary of statistical tests on the quality of the results of PSO-AN against resampling-based and hybrid PSO algorithms. The table compares the performance of PSO-AN with other methods in terms of accuracy and reliability.

<table>
<thead>
<tr>
<th>Method Comparison</th>
<th>AN vs. EER1</th>
<th>AN vs. OCBA</th>
<th>AN vs. EERG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Statistic</td>
<td>$\bar{z}$</td>
<td>$\bar{z}$</td>
<td>$\bar{z}$</td>
</tr>
<tr>
<td>$\bar{z}$</td>
<td>0.00</td>
<td>0.04</td>
<td>0.00</td>
</tr>
<tr>
<td>$\bar{z}$</td>
<td>1.00</td>
<td>1.20</td>
<td>1.00</td>
</tr>
<tr>
<td>$\bar{z}$</td>
<td>0.00</td>
<td>0.06</td>
<td>0.00</td>
</tr>
<tr>
<td>$\bar{z}$</td>
<td>1.00</td>
<td>1.20</td>
<td>1.00</td>
</tr>
<tr>
<td>$\bar{z}$</td>
<td>0.00</td>
<td>0.06</td>
<td>0.00</td>
</tr>
<tr>
<td>$\bar{z}$</td>
<td>1.00</td>
<td>1.20</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Total | 6.00 | 6.00 | 6.00 |

**Note:** The table shows the results of statistical tests comparing the performance of PSO-AN with other methods. The test statistic $\bar{z}$ is used to determine the significance of the differences in performance. Higher values of $\bar{z}$ indicate a greater difference in performance. The total value is the sum of the test statistic values for each comparison.
Table 6.8: Benchmarks on which PSO-AN is better than PSO-ERGC.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>E</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>elliptic</td>
<td>$F_{01}$</td>
<td>$F_{04}$</td>
<td>$F_{09}$</td>
<td>$F_{14}$</td>
<td></td>
</tr>
<tr>
<td>rastrigin</td>
<td>$F_{02}$</td>
<td>$F_{05}$</td>
<td>$F_{10}$</td>
<td>$F_{15}$</td>
<td></td>
</tr>
<tr>
<td>ackley</td>
<td>$F_{03}$</td>
<td>$F_{06}$</td>
<td>$F_{11}$</td>
<td>$F_{16}$</td>
<td></td>
</tr>
<tr>
<td>schwefel</td>
<td>$F_{19}$</td>
<td>$F_{07}$</td>
<td>$F_{12}$</td>
<td>$F_{17}$</td>
<td></td>
</tr>
<tr>
<td>rosenbrock</td>
<td>$F_{20}$</td>
<td>$F_{08}$</td>
<td>$F_{13}^*$</td>
<td>$F_{18}$</td>
<td></td>
</tr>
</tbody>
</table>

$^*$ only at $\sigma_{24}, \sigma_{30}$
tions between the single-evaluation PSO-AN and the resampling-based PSO-EER1. The new hybrid algorithms balance differently the accuracy tradeoff, for which they perform the following number of iterations: 600 in PSO-AN, 300 in PSO-AN_{11}, 200 in PSO-AN_{21}, 150 in PSO-AN_{31}, 120 in PSO-AN_{41}, and 100 in PSO-EER1 (or PSO-AN_{51}, equivalently). Furthermore, the new algorithms are hybridized by utilizing centroid solutions computed from the estimated best five solutions in the swarm. The design of experiments and the parameters utilized are the same as described in Section 6.3 and in Table 6.1, but with a different set of algorithms including PSO-OCBA$^5$ to serve just as a reference.

Results and Discussions

The quality of the results obtained with PSO-AN and its hybrid variants on $F_{13}$ are shown in Figure 6.6, where the algorithms are abbreviated as follows: (a) PSO-AN, (1–5) PSO-AN$_{x1}$, and (O) PSO-OCBA$^5$. The average population statistics for the set of algorithms over all the benchmark functions are shown in Table 6.9. The population statistics on each benchmark function are attached in Appendix 6.B (page 218).

Figure 6.6 confirms our expectations of improving the quality of the results of PSO-AN by re-evaluating the current and personal best solutions. Furthermore, the quality of results remarks the importance of sacrificing iterations over improving the accuracy of the estimated objective values. PSO-EER1 (or PSO-AN$_{51}$, equivalently) shows the best accuracy tradeoff and the trends observed in Figure 6.6 are similar to those on the remaining benchmark functions. Moreover, the average population statistics confirm our expectations on the lifetime of the swarms, (binary and ranked) deception, regular operations, blindness and disorientation, all of which are discussed in the following paragraphs. However, we found again exceptions in the benchmark functions made up of ackley and rastrigin, both of which are discussed at the end of this section.

The lifetime of PSO-AN$_{11}$ is the longest amongst the algorithms, while
Figure 6.6: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by PSO-AN, its hybrid variants and PSO-OCBA\(^5\) (bottom axis) at each level of noise (top axis) in all independent runs on benchmark function \(F_{13}\). The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.

Table 6.9: Average population statistics for PSO-AN and hybrid algorithms over all benchmark functions, levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>AN(_{11})</th>
<th>AN(_{21})</th>
<th>AN(_{31})</th>
<th>AN(_{41})</th>
<th>AN(_{51})</th>
<th>OCBA(^5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifetime(^+)</td>
<td>54.43</td>
<td><strong>99.05</strong></td>
<td>98.25</td>
<td>97.32</td>
<td>96.30</td>
<td>95.19</td>
<td>91.43</td>
</tr>
<tr>
<td>Binary Deception(^-)</td>
<td>97.64</td>
<td>97.08</td>
<td>94.55</td>
<td>92.15</td>
<td>89.52</td>
<td>87.20</td>
<td><strong>83.30</strong></td>
</tr>
<tr>
<td>Ranked Deception(^-)</td>
<td>31.77</td>
<td>38.56</td>
<td>30.91</td>
<td>26.38</td>
<td>23.28</td>
<td>21.21</td>
<td><strong>15.99</strong></td>
</tr>
<tr>
<td>Regular Operation(^+)</td>
<td>41.99</td>
<td>52.34</td>
<td>53.94</td>
<td>55.33</td>
<td>56.61</td>
<td><strong>57.74</strong></td>
<td>56.18</td>
</tr>
<tr>
<td>Updates(^+)</td>
<td>3.59</td>
<td><strong>38.33</strong></td>
<td>35.33</td>
<td>34.63</td>
<td>35.02</td>
<td>35.71</td>
<td>31.34</td>
</tr>
<tr>
<td>Discards(^-)</td>
<td>96.41</td>
<td><strong>61.67</strong></td>
<td>64.67</td>
<td>65.37</td>
<td>64.98</td>
<td>64.29</td>
<td>68.66</td>
</tr>
<tr>
<td>Blindness(^-)</td>
<td>57.64</td>
<td><strong>31.71</strong></td>
<td>33.84</td>
<td>34.63</td>
<td>34.78</td>
<td>34.65</td>
<td>39.21</td>
</tr>
<tr>
<td>Memory</td>
<td>99.00</td>
<td>51.49</td>
<td>61.75</td>
<td>66.96</td>
<td>70.09</td>
<td>72.13</td>
<td>74.35</td>
</tr>
<tr>
<td>Environment</td>
<td>1.00</td>
<td>48.51</td>
<td>38.25</td>
<td>33.04</td>
<td>29.91</td>
<td>27.87</td>
<td>25.65</td>
</tr>
<tr>
<td>Disorientation(^-)</td>
<td>0.36</td>
<td>15.96</td>
<td>12.22</td>
<td>10.04</td>
<td>8.61</td>
<td>7.62</td>
<td><strong>4.61</strong></td>
</tr>
<tr>
<td>Memory</td>
<td>5.00</td>
<td>26.62</td>
<td>22.79</td>
<td>20.10</td>
<td>18.53</td>
<td>17.61</td>
<td>10.49</td>
</tr>
<tr>
<td>Environment</td>
<td>95.00</td>
<td>73.38</td>
<td>77.21</td>
<td>79.90</td>
<td>81.47</td>
<td>82.39</td>
<td>89.51</td>
</tr>
</tbody>
</table>

Larger\(^+\) or smaller\(^-\) is better. **Bold** value is the best. Units in percentages (%).
the lifetime of PSO-AN is the shortest, and in between the lifetime progressively shortens as the computational budget allocates more evaluations to the current solutions. First, the longest lifetime of PSO-AN11 remarks the importance of allocating evaluations to the personal best solutions in order to reduce blindness and thereby prevent stagnation to what may not even be a local optimum. PSO-AN11 has the longest lifetime thanks to a continuous correction of the estimated objective values of the personal best solutions and to the very inaccurately estimated objective values of the current solutions. Specifically, the continuous correction will reduce the proportions of blindness, but will also ease the replacement of the personal best solutions by the current solutions. Therefore, the personal best solutions can be replaced more easily at any point throughout the search, which leads to such a longer lifetime and also to important proportions of disorientation. Differently, the lifetime of the other hybrid PSO-AN algorithms shortens because the current solutions are better estimated, which makes finding a better solution harder once they replace the personal best solutions, especially considering that it is already hard for the regular PSO to find better solutions (in the absence of noise) as observed in Section 3.4.2 (page 75) by the large number of regular discards.

The proportions of binary and ranked deception reduce as more evaluations are allocated to the current solutions. Since the current solutions eventually become personal best solutions, having had better estimated their objective values, the swarms are able to select better neighborhood best solutions. However, notice that the binary deception in PSO-AN is only slightly larger than that of PSO-AN11, whereas its ranked deception is smaller by about 7%. We attribute the smaller ranked deception of PSO-AN to its shorter lifetime and much larger number of iterations as follows. A lifetime of 54.43% suggests that the personal best solutions do not change much in the last half of the search process, for which we also expect that the ranking of the neighborhood best solution will not change much either. Moreover, if we consider the candidates for neighborhood
best solution being sorted according to their true objective values, we expect that the ranking of the estimated neighborhood best solution will lead to a ranking biased towards the better half of the solutions. Hence, considering both lifetime and number of iterations, it is not unexpected that PSO-AN will have a better ranked deception. However, as more evaluations are allocated to the current solutions, their objective values are better estimated and hence the ranked deception is significantly reduced.

Compared to PSO-AN, the proportions of blindness significantly reduce by about 25% in PSO-AN thanks to the allocation of a single evaluation to the personal best solutions. However, blindness remains mostly unchanged as more evaluations are allocated to the current solutions, which is probably due to the smaller number of iterations that these algorithms perform. Nonetheless, blindness is still reduced, which increases the proportions of regular operations as would be expected from having solutions with better estimated objective values. Furthermore, as more evaluations are allocated to the current solutions, the proportions of disorientation also reduce thanks to the current solutions having their objective values better estimated. While the inverse correlation between blindness and disorientation is not clear in these cases, we attribute that to the different number of iterations that the algorithms perform.

The proportions of blindness by the environment decrease thanks to the current solutions having better estimated objective values and to the backward direction of the optimization problem, leading consequently to a proportional increment of blindness by memory. Likewise, the proportions of disorientation by the environment increase mostly because the direction of the optimization problem is backwards, and hence the probability density functions of the worse solutions to which particles update will generally have larger standard deviations than their previous (better) personal best solutions. As the disorientation by the environment increases, the disorientation by memory decreases proportionally.

Lastly, regarding the cases of ackley and rastrigin, the quality of
the results and their respective average population statistics are attached in Appendices 6.C and 6.D (pages 230 and 236, respectively). The quality of the results show that PSO-AN still finds better solutions than any of the new hybrid algorithms in most of the benchmark functions based on ackley and rastrigin, and yet the average population statistics follow the same general trends we have discussed before. The population statistics on these exceptions fail to provide further insights on what makes PSO-AN produce a better quality of results than the hybrid algorithms. Again, these findings do not discredit the importance of the population statistics to analyze the performance of PSO algorithms on optimization problems subject to noise, but instead encourage further studies on the creation of other statistics that could be more helpful to better estimate the quality of the results. Moreover, these findings provide useful information for future studies to address the relative importance of the different population statistics with respect to the quality of the results.

6.5 Summary

Hybrid methods in PSO are noise mitigation mechanisms that combine resampling and single-evaluation methods to reduce deception, blindness and disorientation in the swarm. While it is possible to create several different combinations between resampling and single-evaluation methods, we focused on combining the resampling methods from Chapter 5 with the best performing single-evaluation method from Chapter 4. As such, we have developed three new hybrid algorithms based on PSO-EER, PSO-ERN and PSO-OCBA, each combined with PSO-AN. However, since the new algorithms better estimate the solutions in the swarm thanks to their respective resampling methods, the centroid solutions are computed instead from the estimated best five solutions in the swarm and not from all the solutions as PSO-AN does. The goal of restricting the centroid to be computed from only the best five solutions is to prevent worse solutions
from blurring the effect of better solutions, whereas such is not the case in PSO-AN because the objective values of all its solutions are very inaccurately estimated *per se* and hence it is meaningless to select only a group of them.

The hybrid algorithms found significantly better solutions than their purely resampling-based counterparts in most of the cases, and in the remaining cases the solutions were of similar quality with only a few exceptions in which the quality was worse. The main reason for such improvements was the reduced (binary and ranked) deception in the swarms thanks to the centroid solutions. However, as a consequence of selecting better neighborhood best solutions, we observed again an important increase of the proportions of blindness. This precise issue had been observed before in Sections 3.4.2 and 5.4.4 (pages 75 and 161, respectively) when comparing PSO-ER against PSO-ERGC, but the underlying reasons to such an increment of blindness still remain uncertain.

Amongst the hybrid algorithms, PSO-OCBA⁵ yields the overall best quality of results, although PSO-EER1⁵ is still a strong competitor as its differences in quality of results with respect to PSO-OCBA⁵ are not statistically significant in most cases. The superior quality of results obtained with PSO-OCBA⁵ is mostly thanks to the better estimated objective values of the most important solutions, which leads to a better selection of solutions from which the centroid solutions are computed. However, we consider PSO-EER1⁵ to be a more promising algorithm due to its much simpler approach (conceptually and computationally) and the quality of its results. Furthermore, given the simplicity of PSO-EER1⁵, the quality of its results could even be more finely tuned according to its population statistics than PSO-OCBA⁵ could due to its more complex allocation method. Regarding PSO-ERN⁵, not only is the quality of its results worse than that of its hybrid counterparts, but the tractability of its results is also harder to analyze due to its two-criteria replacement of the personal best solutions. While we find no compelling reason to further investigate the
performance of PSO-ERN\textsuperscript{5}, we find that the two-criteria replacement of the personal best solutions is a cheap and alternative strategy to increase the diversity of the solutions in the swarm.

The vastly superior quality of results found by hybrid PSO algorithms does not fully extend to the case of the single-evaluation PSO-AN. Even when the hybrid PSO algorithms find significantly better solutions in most cases, PSO-AN still manages to outperform the hybrid algorithms in about half of the benchmark functions at higher levels of noise, and most of these cases are made up of \texttt{ackley} and \texttt{rastrigin} functions. These results suggest that we have reached a point in which the most appropriate balance of the accuracy tradeoff is problem-dependent.

Considering the comparison between single-evaluation and hybrid algorithms, it was not surprising to see the better performance of the single-evaluation PSO-AN when compared to the purely resampling-based algorithms. In fact, the improvements were such that the quality of the results obtained with PSO-AN was significantly better regardless of the level of noise in about half of the benchmark functions, and significantly worse in the other half. Again, most of the benchmarks in which PSO-AN found a better quality of results were made up mostly of \texttt{ackley} and \texttt{rastrigin}, but also of \texttt{rosenbrock} in two cases. Such a clear segmentation on the quality of results obtained with single-evaluation and resampling-based algorithms further supports that the most appropriate balance of the accuracy tradeoff is problem-dependent.

The accuracy tradeoff is determined by the allocation of the computational budget of function evaluations. On the one hand, single-evaluation methods evaluate each solution only once, leading to very inaccurately estimated objective values, but also to performing a much larger number of iterations that potentially increases the chances of finding better solutions. On the other hand, resampling-based methods re-evaluate the solutions multiple times, leading to more accurately estimated objective values at the cost of performing a smaller number of iterations. Thus, in
order to explore different accuracy tradeoffs, we designed four new hybrid PSO-AN algorithms that allocate the computational budget differently. Each of the four new algorithms gradually moved from a PSO-AN algorithm that equally allocates the additional computational budget between the current solutions, to a PSO-EER1 algorithm that equally allocates the computational budget between the personal best solutions. The quality of the results obtained with these algorithms gradually improved as they approached PSO-EER1, but still, none of them was able to improve upon the quality of the results obtained with PSO-AN on the cases made up of ackley and rastrigin. Even when the population statistics of the new algorithms also matched our expectations, the quality of the results was not improved as we had expected. Nonetheless, we consider these findings important to encourage further studies that could better estimate the quality of the results. Specifically, we expect further research to focus on new population statistics and on their respective relevance towards the quality of the results obtained with PSO algorithms on optimization problems subject to noise.

Next Chapter

The next chapter will present the conclusions of this thesis, together with general discussions and suggestions for future research on PSO for optimization problems subject to noise.

6.A Population Statistics

The following figures correspond to the population statistics on each benchmark function for the algorithms utilized in the design of experiments from this chapter (Section 6.3, page 182).
Sets A and E

<table>
<thead>
<tr>
<th>Function</th>
<th>$\sigma_{06}$</th>
<th>$\sigma_{12}$</th>
<th>$\sigma_{18}$</th>
<th>$\sigma_{24}$</th>
<th>$\sigma_{30}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{01}$</td>
<td>$2.0e+11$</td>
<td>$1.5e+08$</td>
<td>$5.0e+07$</td>
<td>$1.0e+07$</td>
<td>$3e+16$</td>
</tr>
<tr>
<td>$F_{02}$</td>
<td>$0e+00$</td>
<td>$3e+16$</td>
<td>$6e+16$</td>
<td>$0e+00$</td>
<td>$4e+12$</td>
</tr>
<tr>
<td>$F_{03}$</td>
<td>$1.0e+11$</td>
<td>$2.0e+07$</td>
<td>$1.0e+07$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
</tr>
<tr>
<td>$F_{04}$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
<td>$6e+08$</td>
<td>$0e+00$</td>
<td>$0e+00$</td>
</tr>
<tr>
<td>$F_{05}$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
<td>$6e+08$</td>
<td>$0e+00$</td>
<td>$0e+00$</td>
</tr>
<tr>
<td>$F_{06}$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
<td>$6e+08$</td>
<td>$0e+00$</td>
<td>$0e+00$</td>
</tr>
<tr>
<td>$F_{07}$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
<td>$6e+08$</td>
<td>$0e+00$</td>
<td>$0e+00$</td>
</tr>
<tr>
<td>$F_{08}$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
<td>$6e+08$</td>
<td>$0e+00$</td>
<td>$0e+00$</td>
</tr>
</tbody>
</table>

Set B

<table>
<thead>
<tr>
<th>Function</th>
<th>$\sigma_{06}$</th>
<th>$\sigma_{12}$</th>
<th>$\sigma_{18}$</th>
<th>$\sigma_{24}$</th>
<th>$\sigma_{30}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{04}$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
<td>$6e+08$</td>
<td>$0e+00$</td>
<td>$0e+00$</td>
</tr>
<tr>
<td>$F_{05}$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
<td>$6e+08$</td>
<td>$0e+00$</td>
<td>$0e+00$</td>
</tr>
<tr>
<td>$F_{06}$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
<td>$6e+08$</td>
<td>$0e+00$</td>
<td>$0e+00$</td>
</tr>
<tr>
<td>$F_{07}$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
<td>$6e+08$</td>
<td>$0e+00$</td>
<td>$0e+00$</td>
</tr>
<tr>
<td>$F_{08}$</td>
<td>$0e+00$</td>
<td>$3e+15$</td>
<td>$6e+08$</td>
<td>$0e+00$</td>
<td>$0e+00$</td>
</tr>
</tbody>
</table>

Figure 6.7: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The algorithms are abbreviated as (e) PSO-EER1, (n) PSO-ERN, (o) PSO-OCBA, (E) PSO-EER1\textsuperscript{5}, (N) PSO-ERN\textsuperscript{5}, (O) PSO-OCBA\textsuperscript{5}, (a) PSO-AN and (g) PSO-ERGC. The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
6.A. POPULATION STATISTICS

Figure 6.8: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The algorithms are abbreviated as (e) PSO-EER1, (n) PSO-ERN, (o) PSO-OCBA, (E) PSO-EER1\(^5\), (N) PSO-ERN\(^5\), (O) PSO-OCBA\(^5\), (a) PSO-AN and (g) PSO-ERGC. The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
CHAPTER 6. HYBRID METHODS

Sets A and E | Set B | Set C | Set D

Figure 6.9: Binary Deception. The barplots represent the average proportions of iterations (left axis) at which a particle is deceived by its neighbors for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (e) PSO-EER1, (n) PSO-ERN, (o) PSO-OCBA, (E) PSO-EER1$^5$, (N) PSO-ERN$^5$, (O) PSO-OCBA$^5$, (a) PSO-AN and (g) PSO-ERGC. Particles in PSO-ERGC do not suffer from deception. Smaller proportions are better.
6.A. POPULATION STATISTICS

Figure 6.10: Ranked Deception. The barplots represent the average percentile rank (left axis) based on the true objective values of the selected neighborhood best solutions with respect to the true objective values of the personal best solutions in the swarm for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (e) PSO-EER1, (n) PSO-ERN, (o) PSO-OCBA, (E) PSO-EER1$^5$, (N) PSO-ERN$^5$, (O) PSO-OCBA$^5$, (a) PSO-AN and (g) PSO-ERGC. Particles from PSO-ERGC do not suffer from deception and hence their neighborhood best solutions are always ranked best. Lower percentile ranks are better.
Figure 6.11: Regular Operation, Blindness and Disorientation. The stacked barplots represent the average proportions (left axis) of regular operation (dark gray), blindness (medium gray) and disorientation (light gray) experienced by a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (e) PSO-EER1, (n) PSO-ERN, (o) PSO-OCBA, (E) PSO-EER1$^5$, (N) PSO-ERN$^5$, (O) PSO-OCBA$^5$, (a) PSO-AN and (g) PSO-ERGC. Larger proportions of regular operation and smaller proportions of blindness and disorientation are better.
Figure 6.12: Regular Updates and Discards. The stacked barplots represent the average proportions (left axis) of regular updates (dark gray) and discards (light gray) experienced by a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (e) PSO-EER1, (n) PSO-ERN, (o) PSO-OCBA, (E) PSO-EER1$^5$, (N) PSO-ERN$^5$, (O) PSO-OCBA$^5$, (a) PSO-AN and (g) PSO-ERGC. Larger proportions of regular updates and smaller proportions of regular discards are better.
Figure 6.13: Causes of Blindness. The stacked barplots represent the average proportions (left axis) of blindness caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (e) PSO-EER1, (n) PSO-ERN, (o) PSO-OCBA, (E) PSO-EER1$^5$, (N) PSO-ERN$^5$, (O) PSO-OCBA$^5$, (a) PSO-AN and (g) PSO-ERGC.
Figure 6.14: Causes of Disorientation. The stacked barplots represent the average proportions (left axis) of disorientation caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (e) PSO-EER1, (n) PSO-ERN, (o) PSO-OCBA, (E) PSO-EER1$^5$, (N) PSO-ERN$^5$, (O) PSO-OCBA$^5$, (a) PSO-AN and (g) PSO-ERGC.
Figure 6.15: Lifetime. The barplots represent the proportions of average lifetime (left axis) of a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (e) PSO-EER1, (n) PSO-ERN, (o) PSO-OCBA, (E) PSO-EER1$^5$, (N) PSO-ERN$^5$, (O) PSO-OCBA$^5$, (a) PSO-AN and (g) PSO-ERGC. A longer lifetime is better.
Figure 6.16: Effect of Disorientation and Blindness. The stacked barplots represent the normalized average magnitudes (left axis) of deterioration caused by disorientation (dark gray) and hypothetical improvement missed by blindness (light gray) on the benchmark functions subject to the levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (e) PSO-EER1, (n) PSO-ERN, (o) PSO-OCBA, (E) PSO-EER1$^5$, (N) PSO-ERN$^5$, (O) PSO-OCBA$^5$, (a) PSO-AN and (g) PSO-ERGC.
6.B Population Statistics for PSO-AN and Hybrid Variants

The following figures represent the population statistics for PSO-AN and its hybrid variants from the experiments in Section 6.4.7 (page 197). The hybrid variants allocate differently the computational budget of function evaluations. Each variant is abbreviated as a number between 1 and 5, each of which indicates the number of evaluations that the algorithm performs to the current solutions. The population statistics are presented for the following algorithms: (a) PSO-AN, (1) PSO-AN\textsubscript{11}, (2) PSO-AN\textsubscript{21}, (3) PSO-AN\textsubscript{31}, (4) PSO-AN\textsubscript{41}, (5) PSO-AN\textsubscript{51}, and (O) PSO-OCBA\textsuperscript{5}, performing $t = \{600, 300, 200, 150, 120, 100, 100\}$ iterations, respectively.
CHAPTER 6. HYBRID METHODS

Figure 6.17: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The algorithms are abbreviated as (a) PSO-AN, (1-5) PSO-AN_{1-5}, and (O) PSO-OCBA^5. The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
Figure 6.18: Quality of Results. The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The algorithms are abbreviated as (a) PSO-AN, (1-5) PSO-AN\(_x\), and (O) PSO-OCBA\(^5\). The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
Figure 6.19: Binary Deception. The barplots represent the average proportions of iterations (left axis) in which a particle is deceived by its neighbors for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (a) PSO-AN, (1-5) PSO-AN$_x$, and (O) PSO-OCBA$^5$. Smaller proportions are better.
6.B. POPULATION STATISTICS FOR PSO-AN AND HYBRID VARIANTS

Figure 6.20: Ranked Deception. The barplots represent the average percentile rank (left axis) based on the true objective values of the selected neighborhood best solutions with respect to the true objective values of the personal best solutions in the swarm for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (a) PSO-AN, (1-5) PSO-AN$_{x1}$, and (O) PSO-OCBA$^5$. Lower percentile ranks are better.
Figure 6.21: Regular Operation, Blindness and Disorientation. The stacked barplots represent the average proportions (left axis) of regular operation (dark gray), blindness (medium gray) and disorientation (light gray) experienced by a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (a) PSO-AN, (1-5) PSO-AN$_x$, and (O) PSO-OCBA$^5$. Larger proportions of regular operation and smaller proportions of blindness and disorientation are better.
6.B. POPULATION STATISTICS FOR PSO-AN AND HYBRID VARIANTS

Sets A and E | Set B | Set C | Set D

Figure 6.22: Regular Updates and Discards. The stacked barplots represent the average proportions (left axis) of regular updates (dark gray) and discards (light gray) experienced by a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (a) PSO-AN, (1-5) PSO-AN$_{x1}$, and (O) PSO-OCBA$^5$. Larger proportions of regular updates and smaller proportions of regular discards are better.
Figure 6.23: Causes of Blindness. The stacked barplots represent the average proportions (left axis) of blindness caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (a) PSO-AN, (1-5) PSO-AN$_{x1}$, and (O) PSO-OCBA$^5$. 
### 6.B. POPULATION STATISTICS FOR PSO-AN AND HYBRID VARIANTS

<table>
<thead>
<tr>
<th>Sets A and E</th>
<th>Set B</th>
<th>Set C</th>
<th>Set D</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Stacked Barplot" /></td>
<td><img src="image2.png" alt="Stacked Barplot" /></td>
<td><img src="image3.png" alt="Stacked Barplot" /></td>
<td><img src="image4.png" alt="Stacked Barplot" /></td>
</tr>
</tbody>
</table>

Figure 6.24: Causes of Disorientation. The stacked barplots represent the average proportions (left axis) of disorientation caused by memory (dark gray) and by the environment (light gray) in a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars from left to right). The algorithms are abbreviated as (a) PSO-AN, (1-5) PSO-AN$_{x1}$, and (O) PSO-OCBA$^5$. 
Figure 6.25: Lifetime. The barplots represent the proportions of average lifetime (left axis) of a particle for each algorithm (bottom axis) on the benchmark functions subject to levels of noise $\sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\}$ (bars coloured from light to dark gray). The algorithms are abbreviated as (a) PSO-AN, (1-5) PSO-AN$_{21}$, and (O) PSO-OCBA$^5$. A longer lifetime is better when the swarm does not converge to the global optimum.
Figure 6.26: Effect of Disorientation and Blindness. The stacked barplots represent the normalized average magnitudes (left axis) of deterioration caused by disorientation (dark gray) and hypothetical improvement missed by blindness (light gray) on the benchmark functions subject to the levels of noise \( \sigma \in \{0.06, 0.12, 0.18, 0.24, 0.30\} \) (bars from left to right). The algorithms are abbreviated as (a) PSO-AN, (1-5) PSO-AN\(_{\sigma 1}\), and (O) PSO-OCBA\(^5\).
6.C Ackley Benchmark Functions

The following figures represent the population statistics for PSO-AN and its hybrid variants on the benchmark functions primarily composed of ackley. The population statistics are computed from the experiments in Section 6.4.7 (page 197). The hybrid variants allocate differently the computational budget of function evaluations. Each variant is abbreviated as a number between 1 and 5, each of which indicates the number of evaluations that the algorithm performs to the current solutions. The population statistics are presented for the following algorithms: (a) PSO-AN, (1) PSO-AN$_{11}$, (2) PSO-AN$_{21}$, (3) PSO-AN$_{31}$, (4) PSO-AN$_{41}$, (5) PSO-AN$_{51}$, and (O) PSO-OCBA$^5$. The number of iterations performed by the algorithms are $t = \{600, 300, 200, 150, 120, 100, 100\}$, respectively.

The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
(This page is intentionally left blank)
CHAPTER 6. HYBRID METHODS

$F_{03}$: shifted ackley’s function

![Figure 6.27: Quality of Results on $F_{03}$.](image)

Table 6.10: Average population statistics on $F_{03}$ over all levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>AN$_{11}$</th>
<th>AN$_{21}$</th>
<th>AN$_{31}$</th>
<th>AN$_{41}$</th>
<th>AN$_{51}$</th>
<th>OCBA$^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifetime$^+$</td>
<td>49.76</td>
<td><strong>99.06</strong></td>
<td>98.23</td>
<td>97.36</td>
<td>96.30</td>
<td>95.02</td>
<td>88.65</td>
</tr>
<tr>
<td>Binary Deception$^-$</td>
<td>92.72</td>
<td>96.99</td>
<td>95.21</td>
<td>94.15</td>
<td>92.27</td>
<td>90.95</td>
<td><strong>87.26</strong></td>
</tr>
<tr>
<td>Ranked Deception$^-$</td>
<td>26.20</td>
<td>46.61</td>
<td>43.68</td>
<td>41.03</td>
<td>37.40</td>
<td>34.71</td>
<td><strong>24.09</strong></td>
</tr>
<tr>
<td><strong>Regular Operation$^+$</strong></td>
<td>35.23</td>
<td><strong>50.23</strong></td>
<td>50.17</td>
<td>50.08</td>
<td>49.56</td>
<td>49.23</td>
<td>45.72</td>
</tr>
<tr>
<td>Updates$^+$</td>
<td>2.77</td>
<td><strong>36.13</strong></td>
<td>31.07</td>
<td>28.28</td>
<td>27.08</td>
<td>26.51</td>
<td>18.42</td>
</tr>
<tr>
<td>Discards$^-$</td>
<td>97.23</td>
<td><strong>63.87</strong></td>
<td>68.93</td>
<td>71.72</td>
<td>72.92</td>
<td>73.49</td>
<td>81.58</td>
</tr>
<tr>
<td><strong>Blindness$^-$</strong></td>
<td>64.55</td>
<td><strong>32.78</strong></td>
<td>36.15</td>
<td>38.29</td>
<td>40.28</td>
<td>41.68</td>
<td>49.95</td>
</tr>
<tr>
<td>Memory</td>
<td>98.90</td>
<td>51.96</td>
<td>62.61</td>
<td>68.15</td>
<td>71.76</td>
<td>74.23</td>
<td>77.22</td>
</tr>
<tr>
<td>Environment</td>
<td>1.10</td>
<td>48.04</td>
<td>37.39</td>
<td>31.85</td>
<td>28.24</td>
<td>25.77</td>
<td>22.78</td>
</tr>
<tr>
<td><strong>Disorientation$^-$</strong></td>
<td><strong>0.22</strong></td>
<td>16.99</td>
<td>13.68</td>
<td>11.63</td>
<td>10.16</td>
<td>9.08</td>
<td>4.33</td>
</tr>
<tr>
<td>Memory</td>
<td>0.74</td>
<td>27.93</td>
<td>25.38</td>
<td>23.53</td>
<td>21.88</td>
<td>20.59</td>
<td>8.65</td>
</tr>
<tr>
<td>Environment</td>
<td>99.26</td>
<td>72.07</td>
<td>74.62</td>
<td>76.47</td>
<td>78.12</td>
<td>79.41</td>
<td>91.35</td>
</tr>
</tbody>
</table>

Larger$^+$ or smaller$^-$ is better. **Bold** value is the best. Units in percentages (%).
6.C. ACKLEY BENCHMARK FUNCTIONS

\( F_{06} \): single-group shifted and \( m \)-rotated ackley’s function

![Figure 6.28: Quality of Results on \( F_{06} \).](image)

Table 6.11: Average population statistics on \( F_{06} \) over all levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>AN_{11}</th>
<th>AN_{21}</th>
<th>AN_{31}</th>
<th>AN_{41}</th>
<th>AN_{51}</th>
<th>OCBA^5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lifetime^+</strong></td>
<td>49.59</td>
<td><strong>99.03</strong></td>
<td>98.28</td>
<td>97.29</td>
<td>96.21</td>
<td>95.07</td>
<td>88.89</td>
</tr>
<tr>
<td>Binary Deception^-</td>
<td>97.95</td>
<td>98.32</td>
<td>97.75</td>
<td>97.79</td>
<td>97.54</td>
<td>97.56</td>
<td><strong>97.04</strong></td>
</tr>
<tr>
<td>Ranked Deception^-</td>
<td>41.21</td>
<td>47.43</td>
<td>49.71</td>
<td>48.64</td>
<td>47.78</td>
<td>46.95</td>
<td><strong>41.15</strong></td>
</tr>
<tr>
<td><strong>Regular Operation^+</strong></td>
<td>47.84</td>
<td>51.36</td>
<td>52.32</td>
<td>52.75</td>
<td>53.00</td>
<td><strong>53.23</strong></td>
<td>51.47</td>
</tr>
<tr>
<td>Updates^+</td>
<td>1.80</td>
<td><strong>35.26</strong></td>
<td>29.45</td>
<td>26.53</td>
<td>24.85</td>
<td>23.85</td>
<td>16.08</td>
</tr>
<tr>
<td>Discards^-</td>
<td>98.20</td>
<td><strong>64.74</strong></td>
<td>70.55</td>
<td>73.47</td>
<td>75.15</td>
<td>76.15</td>
<td>83.92</td>
</tr>
<tr>
<td><strong>Blindness^-</strong></td>
<td>51.81</td>
<td><strong>31.62</strong></td>
<td>33.87</td>
<td>35.44</td>
<td>36.53</td>
<td>37.39</td>
<td>43.69</td>
</tr>
<tr>
<td>Memory</td>
<td>98.91</td>
<td>51.85</td>
<td>62.33</td>
<td>68.06</td>
<td>72.02</td>
<td>74.41</td>
<td>78.09</td>
</tr>
<tr>
<td>Environment</td>
<td>1.09</td>
<td>48.15</td>
<td>37.67</td>
<td>31.94</td>
<td>27.98</td>
<td>25.59</td>
<td>21.91</td>
</tr>
<tr>
<td><strong>Disorientation^-</strong></td>
<td><strong>0.34</strong></td>
<td>17.02</td>
<td>13.81</td>
<td>11.82</td>
<td>10.47</td>
<td>9.39</td>
<td>4.84</td>
</tr>
<tr>
<td>Memory</td>
<td>1.70</td>
<td>26.38</td>
<td>22.70</td>
<td>20.17</td>
<td>18.29</td>
<td>16.90</td>
<td>7.26</td>
</tr>
<tr>
<td>Environment</td>
<td>98.30</td>
<td>73.62</td>
<td>77.30</td>
<td>79.83</td>
<td>81.71</td>
<td>83.10</td>
<td>92.74</td>
</tr>
</tbody>
</table>

Larger^+ or smaller^- is better. **Bold** value is the best. Units in percentages (%).
$F_{11}$: $\frac{d}{2m}$-group shifted and $m$-rotated ackley’s function

Figure 6.29: Quality of Results on $F_{11}$.

Table 6.12: Average population statistics on $F_{11}$ over all levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>AN$_{11}$</th>
<th>AN$_{21}$</th>
<th>AN$_{31}$</th>
<th>AN$_{41}$</th>
<th>AN$_{51}$</th>
<th>OCBA$^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifetime$^+$</td>
<td>49.67</td>
<td><strong>99.05</strong></td>
<td>98.23</td>
<td>97.33</td>
<td>96.30</td>
<td>95.15</td>
<td>88.71</td>
</tr>
<tr>
<td>Binary Deception$^-$</td>
<td>94.26</td>
<td>97.26</td>
<td>95.79</td>
<td>95.20</td>
<td>94.05</td>
<td>92.90</td>
<td><strong>90.97</strong></td>
</tr>
<tr>
<td>Ranked Deception$^-$</td>
<td>29.51</td>
<td>47.06</td>
<td>45.24</td>
<td>42.99</td>
<td>40.08</td>
<td>37.33</td>
<td><strong>28.30</strong></td>
</tr>
<tr>
<td><strong>Regular Operation$^+$</strong></td>
<td>38.67</td>
<td>50.33</td>
<td><strong>50.45</strong></td>
<td>50.29</td>
<td>50.10</td>
<td>49.49</td>
<td>46.68</td>
</tr>
<tr>
<td>Updates$^+$</td>
<td>2.48</td>
<td><strong>35.96</strong></td>
<td>30.80</td>
<td>28.20</td>
<td>26.70</td>
<td>26.19</td>
<td>17.87</td>
</tr>
<tr>
<td>Discards$^-$</td>
<td>97.52</td>
<td><strong>64.04</strong></td>
<td>69.20</td>
<td>71.80</td>
<td>73.30</td>
<td>73.81</td>
<td>82.13</td>
</tr>
<tr>
<td><strong>Blindness$^-$</strong></td>
<td>61.08</td>
<td><strong>32.74</strong></td>
<td>35.80</td>
<td>37.92</td>
<td>39.68</td>
<td>41.40</td>
<td>48.86</td>
</tr>
<tr>
<td>Memory</td>
<td>98.88</td>
<td>52.04</td>
<td>62.59</td>
<td>68.05</td>
<td>71.80</td>
<td>74.34</td>
<td>77.35</td>
</tr>
<tr>
<td>Environment</td>
<td>1.12</td>
<td>47.96</td>
<td>37.41</td>
<td>31.95</td>
<td>28.20</td>
<td>25.66</td>
<td>22.65</td>
</tr>
<tr>
<td><strong>Disorientation$^-$</strong></td>
<td><strong>0.25</strong></td>
<td>16.93</td>
<td>13.75</td>
<td>11.79</td>
<td>10.22</td>
<td>9.11</td>
<td>4.46</td>
</tr>
<tr>
<td>Memory</td>
<td>0.87</td>
<td>27.58</td>
<td>25.04</td>
<td>23.03</td>
<td>21.63</td>
<td>20.28</td>
<td>8.44</td>
</tr>
<tr>
<td>Environment</td>
<td>99.13</td>
<td>72.42</td>
<td>74.96</td>
<td>76.97</td>
<td>78.37</td>
<td>79.72</td>
<td>91.56</td>
</tr>
</tbody>
</table>

Larger$^+$ or smaller$^-$ is better. **Bold** value is the best. Units in percentages (%).
$F_{16}$: $\frac{d}{m}$-group shifted and $m$-rotated ackley’s function

![Quality of Results on $F_{16}$](chart.png)

Figure 6.30: Quality of Results on $F_{16}$.

Table 6.13: Average population statistics on $F_{16}$ over all levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>AN_{11}</th>
<th>AN_{21}</th>
<th>AN_{31}</th>
<th>AN_{41}</th>
<th>AN_{51}</th>
<th>OCBA^5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lifetime</strong>+</td>
<td>50.22</td>
<td>99.03</td>
<td>98.29</td>
<td>97.39</td>
<td>96.22</td>
<td>94.93</td>
<td>88.87</td>
</tr>
<tr>
<td><strong>Binary Deception</strong>−</td>
<td>92.56</td>
<td>96.86</td>
<td>94.99</td>
<td>93.92</td>
<td>92.80</td>
<td>91.88</td>
<td>87.43</td>
</tr>
<tr>
<td><strong>Ranked Deception</strong>−</td>
<td>26.34</td>
<td>46.89</td>
<td>44.33</td>
<td>41.55</td>
<td>38.59</td>
<td>35.98</td>
<td>24.43</td>
</tr>
<tr>
<td><strong>Regular Operation</strong>+</td>
<td>36.76</td>
<td><strong>50.30</strong></td>
<td>50.25</td>
<td>50.16</td>
<td>49.84</td>
<td>49.31</td>
<td>45.67</td>
</tr>
<tr>
<td>Updates+</td>
<td>2.66</td>
<td><strong>36.14</strong></td>
<td>30.88</td>
<td>28.23</td>
<td>26.94</td>
<td>26.52</td>
<td>18.46</td>
</tr>
<tr>
<td>Discards−</td>
<td>97.34</td>
<td><strong>63.86</strong></td>
<td>69.12</td>
<td>71.77</td>
<td>73.06</td>
<td>73.48</td>
<td>81.54</td>
</tr>
<tr>
<td><strong>Blindness</strong>−</td>
<td>63.01</td>
<td><strong>32.75</strong></td>
<td>36.13</td>
<td>38.23</td>
<td>39.93</td>
<td>41.57</td>
<td>49.96</td>
</tr>
<tr>
<td>Memory</td>
<td>98.91</td>
<td>52.05</td>
<td>62.62</td>
<td>68.23</td>
<td>71.67</td>
<td>74.09</td>
<td>77.37</td>
</tr>
<tr>
<td>Environment</td>
<td>1.09</td>
<td>47.95</td>
<td>37.38</td>
<td>31.77</td>
<td>28.33</td>
<td>25.91</td>
<td>22.63</td>
</tr>
<tr>
<td><strong>Disorientation</strong>−</td>
<td><strong>0.23</strong></td>
<td>16.95</td>
<td>13.63</td>
<td>11.61</td>
<td>10.23</td>
<td>9.12</td>
<td>4.38</td>
</tr>
<tr>
<td>Memory</td>
<td>0.92</td>
<td>27.65</td>
<td>25.10</td>
<td>23.17</td>
<td>22.12</td>
<td>20.71</td>
<td>8.61</td>
</tr>
<tr>
<td>Environment</td>
<td>99.08</td>
<td>72.35</td>
<td>74.90</td>
<td>76.83</td>
<td>77.88</td>
<td>79.29</td>
<td>91.39</td>
</tr>
</tbody>
</table>

Larger+ or smaller− is better. **Bold** value is the best. Units in percentages (%).
6.D Rastrigin Benchmark Functions

The following figures represent the population statistics for PSO-AN and its hybrid variants on the benchmark functions primarily composed of rastrigin. The population statistics are computed from the experiments in Section 6.4.7 (page 197). The hybrid variants allocate differently the computational budget of function evaluations. Each variant is abbreviated as a number between 1 and 5, each of which indicates the number of evaluations that the algorithm performs to the current solutions. The population statistics are presented for the following algorithms: (a) PSO-AN, (1) PSO-AN$_{11}$, (2) PSO-AN$_{21}$, (3) PSO-AN$_{31}$, (4) PSO-AN$_{41}$, (5) PSO-AN$_{51}$, and (O) PSO-OCBA$_{5}$. The number of iterations performed by the algorithms are $t = \{600, 300, 200, 150, 120, 100, 100\}$, respectively.

The boxplots represent the true objective values (left axis) of the best solutions found by the algorithms (bottom axis) at each level of noise (top axis) in all independent runs. The boxplots are coloured from light to dark gray to ease the comparison. The benchmark functions are minimization problems, therefore lower objective values indicate better solutions.
**F₀₂: shifted rastrigin’s function**

![Quality of Results on F₀₂](image)

**Figure 6.31:** Quality of Results on F₀₂.

**Table 6.14:** Average population statistics on F₀₂ over all levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>AN₁₁</th>
<th>AN₂₁</th>
<th>AN₃₁</th>
<th>AN₄₁</th>
<th>AN₅₁</th>
<th>OCBA⁵</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifetime⁺</td>
<td>51.02</td>
<td><strong>99.06</strong></td>
<td>98.20</td>
<td>97.33</td>
<td>96.31</td>
<td>95.14</td>
<td>90.29</td>
</tr>
<tr>
<td>Binary Deception⁻</td>
<td>98.97</td>
<td>97.97</td>
<td>96.25</td>
<td>95.07</td>
<td>93.17</td>
<td>92.40</td>
<td><strong>88.76</strong></td>
</tr>
<tr>
<td>Ranked Deception⁻</td>
<td>35.54</td>
<td>41.24</td>
<td>34.32</td>
<td>31.21</td>
<td>28.65</td>
<td>27.67</td>
<td><strong>22.63</strong></td>
</tr>
<tr>
<td><strong>Regular Operation⁺</strong></td>
<td>43.48</td>
<td>51.47</td>
<td>52.69</td>
<td>53.76</td>
<td>54.62</td>
<td><strong>55.76</strong></td>
<td>54.95</td>
</tr>
<tr>
<td>Updates⁺</td>
<td>2.60</td>
<td><strong>36.41</strong></td>
<td>31.55</td>
<td>29.11</td>
<td>27.78</td>
<td>26.81</td>
<td>19.21</td>
</tr>
<tr>
<td>Discards⁻</td>
<td>97.40</td>
<td><strong>63.59</strong></td>
<td>68.45</td>
<td>70.89</td>
<td>72.22</td>
<td>73.19</td>
<td>80.79</td>
</tr>
<tr>
<td><strong>Blindness⁻</strong></td>
<td>56.18</td>
<td><strong>31.83</strong></td>
<td>34.04</td>
<td>35.03</td>
<td>35.48</td>
<td>35.32</td>
<td>39.85</td>
</tr>
<tr>
<td>Memory</td>
<td>99.05</td>
<td>51.72</td>
<td>62.39</td>
<td>67.87</td>
<td>71.65</td>
<td>74.10</td>
<td>78.54</td>
</tr>
<tr>
<td>Environment</td>
<td>0.95</td>
<td>48.28</td>
<td>37.61</td>
<td>32.13</td>
<td>28.35</td>
<td>25.90</td>
<td>21.46</td>
</tr>
<tr>
<td><strong>Disorientation⁻</strong></td>
<td><strong>0.34</strong></td>
<td>16.70</td>
<td>13.27</td>
<td>11.21</td>
<td>9.90</td>
<td>8.92</td>
<td>5.20</td>
</tr>
<tr>
<td>Memory</td>
<td>4.10</td>
<td>26.85</td>
<td>22.73</td>
<td>19.76</td>
<td>17.22</td>
<td>15.44</td>
<td>7.87</td>
</tr>
<tr>
<td>Environment</td>
<td>95.90</td>
<td>73.15</td>
<td>77.27</td>
<td>80.24</td>
<td>82.78</td>
<td>84.56</td>
<td>92.13</td>
</tr>
</tbody>
</table>

Larger⁺ or smaller⁻ is better. **Bold** value is the best. Units in percentages (%).
6.D. RASTRIGIN BENCHMARK FUNCTIONS

$F_{05}$: single-group shifted and $m$-rotated rastrigin’s function

![Graph showing quality of results on $F_{05}$](image)

**Figure 6.32:** Quality of Results on $F_{05}$.

**Table 6.15:** Average population statistics on $F_{05}$ over all levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>AN$_{11}$</th>
<th>AN$_{31}$</th>
<th>AN$_{31}$</th>
<th>AN$_{51}$</th>
<th>OCBA$^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lifetime</strong></td>
<td>56.03</td>
<td>99.05</td>
<td>98.19</td>
<td>97.06</td>
<td>96.01</td>
<td>94.50</td>
</tr>
<tr>
<td><strong>Binary Deception</strong></td>
<td>99.72</td>
<td>96.15</td>
<td>91.67</td>
<td>87.28</td>
<td>83.63</td>
<td>80.71</td>
</tr>
<tr>
<td><strong>Regular Operation</strong></td>
<td>62.46</td>
<td>54.97</td>
<td>59.45</td>
<td>64.18</td>
<td>68.31</td>
<td>72.26</td>
</tr>
<tr>
<td><strong>Updates</strong></td>
<td>1.80</td>
<td>36.31</td>
<td>31.70</td>
<td>29.01</td>
<td>27.75</td>
<td>26.01</td>
</tr>
<tr>
<td><strong>Discards</strong></td>
<td>98.20</td>
<td>63.69</td>
<td>68.30</td>
<td>70.99</td>
<td>72.25</td>
<td>73.99</td>
</tr>
<tr>
<td><strong>Blindness</strong></td>
<td>37.09</td>
<td>29.60</td>
<td>29.18</td>
<td>26.85</td>
<td>24.04</td>
<td><strong>21.02</strong></td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>98.96</td>
<td>51.25</td>
<td>61.35</td>
<td>65.90</td>
<td>67.82</td>
<td>68.81</td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td>1.04</td>
<td>48.75</td>
<td>38.65</td>
<td>34.10</td>
<td>32.18</td>
<td>31.19</td>
</tr>
<tr>
<td><strong>Disorientation</strong></td>
<td><strong>0.45</strong></td>
<td>15.43</td>
<td>11.37</td>
<td>8.97</td>
<td>7.65</td>
<td>6.72</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>3.26</td>
<td>24.54</td>
<td>19.22</td>
<td>15.75</td>
<td>14.39</td>
<td>14.03</td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td>96.74</td>
<td>75.46</td>
<td>80.78</td>
<td>84.25</td>
<td>85.61</td>
<td>85.97</td>
</tr>
</tbody>
</table>

Larger $^+$ or smaller $^-$ is better. **Bold** value is the best. Units in percentages (%).
$F_{10}$: $\frac{d}{2^m}$-group shifted and $m$-rotated rastrigin’s function

Figure 6.33: Quality of Results on $F_{10}$.

Table 6.16: Average population statistics on $F_{10}$ over all levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>AN$_{11}$</th>
<th>AN$_{21}$</th>
<th>AN$_{31}$</th>
<th>AN$_{41}$</th>
<th>AN$_{51}$</th>
<th>OCBA$^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifetime$^+$</td>
<td>50.55</td>
<td><strong>99.04</strong></td>
<td>98.25</td>
<td>97.30</td>
<td>96.09</td>
<td>95.17</td>
<td>90.14</td>
</tr>
<tr>
<td>Binary Deception$^-$</td>
<td>98.61</td>
<td>97.93</td>
<td>95.97</td>
<td>94.18</td>
<td>91.89</td>
<td>90.70</td>
<td><strong>87.03</strong></td>
</tr>
<tr>
<td><strong>Regular Operation$^+$</strong></td>
<td>45.68</td>
<td>51.56</td>
<td>52.84</td>
<td>54.04</td>
<td>55.08</td>
<td>56.17</td>
<td><strong>56.30</strong></td>
</tr>
<tr>
<td>Updates$^+$</td>
<td>2.43</td>
<td><strong>36.30</strong></td>
<td>31.48</td>
<td>28.85</td>
<td>27.48</td>
<td>26.61</td>
<td>18.38</td>
</tr>
<tr>
<td>Discards$^-$</td>
<td>97.57</td>
<td><strong>63.70</strong></td>
<td>68.52</td>
<td>71.15</td>
<td>72.52</td>
<td>73.39</td>
<td>81.62</td>
</tr>
<tr>
<td><strong>Blindness$^-$</strong></td>
<td>53.97</td>
<td><strong>31.81</strong></td>
<td>33.89</td>
<td>34.67</td>
<td>34.95</td>
<td>34.80</td>
<td>38.51</td>
</tr>
<tr>
<td>Memory</td>
<td>99.05</td>
<td>51.69</td>
<td>62.31</td>
<td>68.27</td>
<td>71.73</td>
<td>74.26</td>
<td>78.82</td>
</tr>
<tr>
<td>Environment</td>
<td>0.95</td>
<td>48.31</td>
<td>37.69</td>
<td>31.73</td>
<td>28.27</td>
<td>25.74</td>
<td>21.18</td>
</tr>
<tr>
<td><strong>Disorientation$^-$</strong></td>
<td>0.35</td>
<td>16.63</td>
<td>13.27</td>
<td>11.30</td>
<td>9.97</td>
<td>9.03</td>
<td>5.19</td>
</tr>
<tr>
<td>Memory</td>
<td>4.19</td>
<td>26.85</td>
<td>22.51</td>
<td>19.28</td>
<td>16.91</td>
<td>15.24</td>
<td>7.64</td>
</tr>
<tr>
<td>Environment</td>
<td>95.81</td>
<td>73.15</td>
<td>77.49</td>
<td>80.72</td>
<td>83.09</td>
<td>84.76</td>
<td>92.36</td>
</tr>
</tbody>
</table>

Larger$^+$ or smaller$^-$ is better. Bold value is the best. Units in percentages (%).
6.D. RASTRIGIN BENCHMARK FUNCTIONS

\[ F_{15} : \frac{d}{m} - \text{group shifted and} \ m-\text{rotated rastrigin's function} \]

![Figure 6.34: Quality of Results on F_{15}.](image)

Table 6.17: Average population statistics on \( F_{15} \) over all levels of noise, independent runs, iterations and particles.

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
<th>AN_{11}</th>
<th>AN_{21}</th>
<th>AN_{31}</th>
<th>AN_{41}</th>
<th>AN_{51}</th>
<th>OCBA$^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifetime$^+$</td>
<td>51.61</td>
<td><strong>99.04</strong></td>
<td>98.31</td>
<td>97.35</td>
<td>96.20</td>
<td>94.96</td>
<td>89.69</td>
</tr>
<tr>
<td>Binary Deception$^-$</td>
<td>98.54</td>
<td>97.90</td>
<td>95.30</td>
<td>92.55</td>
<td>91.02</td>
<td>89.09</td>
<td><strong>85.64</strong></td>
</tr>
<tr>
<td>Ranked Deception$^-$</td>
<td>38.12</td>
<td>38.37</td>
<td>29.02</td>
<td>26.09</td>
<td>25.67</td>
<td>25.45</td>
<td><strong>24.52</strong></td>
</tr>
<tr>
<td><strong>Regular Operation$^+$</strong></td>
<td>46.63</td>
<td>51.70</td>
<td>53.19</td>
<td>54.41</td>
<td>55.71</td>
<td>56.73</td>
<td><strong>57.09</strong></td>
</tr>
<tr>
<td>Updates$^+$</td>
<td>2.37</td>
<td><strong>36.25</strong></td>
<td>31.28</td>
<td>28.72</td>
<td>26.99</td>
<td>26.02</td>
<td>17.87</td>
</tr>
<tr>
<td>Discards$^-$</td>
<td>97.63</td>
<td><strong>63.75</strong></td>
<td>68.72</td>
<td>71.28</td>
<td>73.01</td>
<td>73.98</td>
<td>82.13</td>
</tr>
<tr>
<td>Blindness$^-$</td>
<td>53.02</td>
<td><strong>31.58</strong></td>
<td>33.50</td>
<td>34.32</td>
<td>34.29</td>
<td>34.22</td>
<td>37.73</td>
</tr>
<tr>
<td>Memory</td>
<td>99.03</td>
<td>51.46</td>
<td>62.28</td>
<td>67.83</td>
<td>71.51</td>
<td>74.37</td>
<td>78.93</td>
</tr>
<tr>
<td>Environment</td>
<td>0.97</td>
<td>48.54</td>
<td>37.72</td>
<td>32.17</td>
<td>28.49</td>
<td>25.63</td>
<td>21.07</td>
</tr>
<tr>
<td><strong>Disorientation$^-$</strong></td>
<td><strong>0.35</strong></td>
<td>16.72</td>
<td>13.31</td>
<td>11.27</td>
<td>10.00</td>
<td>9.05</td>
<td>5.19</td>
</tr>
<tr>
<td>Memory</td>
<td>3.78</td>
<td>26.92</td>
<td>22.68</td>
<td>19.44</td>
<td>16.78</td>
<td>15.07</td>
<td>7.71</td>
</tr>
<tr>
<td>Environment</td>
<td>96.22</td>
<td>73.08</td>
<td>77.32</td>
<td>80.56</td>
<td>83.22</td>
<td>84.93</td>
<td><strong>92.29</strong></td>
</tr>
</tbody>
</table>

Larger$^+$ or smaller$^-$ is better. **Bold** value is the best. Units in percentages (%).
Chapter 7

Conclusions

The overall goal of this thesis was to study the effect of noise on PSO beyond the known deterioration of the quality of its results in order to develop more efficient noise mitigation mechanisms that reduce such a deterioration. The goal was achieved by a) formally identifying the conditions responsible for the deterioration of the quality of its results, b) studying the extents to which such conditions affect the particles throughout the search process in PSO algorithms utilizing conceptually different noise mitigation mechanisms, and c) developing new PSO algorithms to further improve upon the quality of the results obtained with existing PSO algorithms. Experiments performed on 20 large-scale benchmark functions of varying difficulties and challenges supports the substantial reduction of deterioration achieved at different levels of multiplicative Gaussian noise.

The remainder of this chapter is structured as follows. Section 7.1 summarizes the research objectives achieved in this thesis. Section 7.2 presents the main conclusions from each of the individual contribution chapters. Section 7.3 provides further discussions on key topics covered in this thesis. Finally, Section 7.4 discusses potential areas for future research.
7.1 Achieved Objectives

The following research objectives have been achieved in this thesis.

- This thesis has identified and formally defined *deception, blindness* and *disorientation* as three conditions from which particles suffer due to the effect of noise on optimization problems. While deception is just a new name to refer to the problem known in the literature as the incorrect selection of the neighborhood best solution, blindness and disorientation are new concepts to the PSO domain. Deception, blindness and disorientation are responsible for the deterioration of the quality of the results observed in the PSO algorithms, for which we developed a set of *population statistics* to evaluate the extents to which these conditions affect particles throughout the search process. The population statistics computed for the regular PSO and PSO-ER revealed that particles suffer from large proportions of deception and blindness, confirmed the sensitivity of PSO to noise in optimization problems, remarked the importance of incorporating resampling methods into PSO, and suggested that blindness and disorientation should be reduced first before addressing deception.

- This thesis has studied the population statistics for single-evaluation PSO algorithms on optimization problems subject to noise. The study revealed that PSO-E, despite being the main reference for single-evaluation methods, often yields significantly worse results than the regular PSO lacking a noise mitigation mechanism. The underlying reason to such a worse quality of results is that the evaporation mechanism often leads to divergent behaviour due to large proportions of disorientation. To improve upon these results, we proposed the PSO-PU algorithm, where particles decide probabilistically whether to replace their personal best solutions at each unsuccessful iteration. The quality of the results obtained with PSO-PU
7.1. ACHIEVED OBJECTIVES

was generally better than that obtained with PSO-E because it provided the swarm with more exploitation capabilities, thereby preventing divergent behaviour up to a certain extent. Still, the regular PSO provides a better quality of results than both PSO-E and PSO-PU, for which we proposed the PSO-AN algorithm as a different approach where particles are partially attracted towards the centroid solution within their neighborhoods. PSO-AN successfully improved upon the quality of the results obtained with the regular PSO because particles suffered less from ranked deception thanks to the centroid solution, which is generally a better solution than any other solution particles could have selected based on their very inaccurately estimated objective values.

- This thesis has studied the population statistics for resampling-based PSO algorithms on optimization problems subject to noise. The study proposed two new resampling-based algorithms, namely PSO-EER and PSO-ERN, whose quality of results is superior to that of PSO-ER. However, the more advanced PSO-OCBA algorithm still finds significantly better solutions than the proposed algorithms thanks to the reduced proportions of deception, blindness and disorientation achieved with its sequential allocation approach. This study further consolidated the performance of PSO-OCBA, but also remarked the higher complexity of its underlying resampling method and its higher computational cost.

- This thesis has proposed the first hybrid PSO algorithms in the literature that incorporate mechanisms from both single-evaluation and resampling methods. The hybrid algorithms were designed combining each of the resampling methods studied with the centroid solution of PSO-AN computed from the estimated best five solutions. The quality of the results from the hybrid algorithms was vastly superior to that of their purely resampling-based counterparts, and
mostly better than the single-evaluation PSO-AN with some exceptions. The best quality of results was obtained with the PSO-OCBA hybrid, but the difference with respect to the PSO-EER1 hybrid was mostly not significantly different. Thus, the much simpler and computationally cheaper approach of PSO-EER1 is a promising algorithm because its simplicity and flexibility allows straightforward modifications that could provide convenient to tune its general performance based on its population statistics.

7.2 Main Conclusions

This section presents the main conclusions from the four major contribution chapters that address our research objectives, namely Chapters 3 to 6.

7.2.1 Deception, Blindness and Disorientation

In PSO, particles suffer from three conditions when optimization problems are subject to noise, namely deception, blindness and disorientation, all of which are responsible for the deterioration of the quality of its results. A particle suffers from deception when it fails to correctly select the neighborhood best solution, from blindness when it misses an opportunity to improve upon its personal best solution, and from disorientation when it mistakenly replaces its personal best solution with a worse solution. Particles will suffer from these conditions whenever PSO is utilized on optimization problems subject to noise. Therefore, it is important to study the extents to which these conditions affect the particles in order to design efficient noise mitigation mechanisms that will prevent (or at least reduce) the deterioration of the quality of its results.

The population statistics is the term by which we refer to a set of indicators that measure different characteristics of the swarm throughout the search process. Amongst these statistics, we have computed the propor-
7.2. MAIN CONCLUSIONS

tions of deception, blindness and disorientation for different PSO algorithms on 20 large-scale benchmark functions subject to different levels of multiplicative Gaussian noise. The results revealed that particles suffer from large proportions of deception and blindness, and rather small proportions of disorientation, all of which have an important impact on the quality of the results. Besides these three population statistics, the following 13 others have been designed and studied for the algorithms.

The ranked deception (1) computes the average ranking of the selected neighborhood best solutions in the swarm with respect to the personal best solutions, thereby providing information about the quality of the solutions that partially attract the swarm. The ranked deception supplements the information provided by the binary deception, and provides even more useful information regarding the overall quality of the neighborhood best solutions selected.

The blindness by memory (2) and blindness by the environment (3) attributes the cause of blindness in particles to the location of the less accurately estimated objective values. Blinded particles suffer from blindness by memory when the objective values of the personal best solutions are estimated less accurately than those of the current solutions. Otherwise, particles suffer from blindness by the environment. These indicators provide valuable information to decide which solution, either the current or personal best solution, should be better estimated. Similarly, disorientation by memory (4) and disorientation by the environment (5) provide the same information for disoriented particles.

The regular operations (6) comprise those particles which replace the personal best solutions with truly better current solutions, or just discard the current solutions because they are truly worse than the personal best solutions. As such, the regular operations are made up of two other statistics, namely regular updates (7) and regular discards (8), respectively. The proportions of regular operations are the complement of the proportions of blindness and disorientation, and together with the regular updates and
discards provide valuable information to assess the efficiency of the incorporated noise mitigation mechanism.

The lifetime (9) of the swarm refers to the average last iteration at which particles no longer find better solutions. The lifetime provides valuable information that not only reflects the convergence of the swarm, but also the proportions of unsuccessful iterations whose computational budget could have been spent differently.

The deterioration caused by disorientation (10) computes the average magnitude by which the true objective values of the personal best solutions are worsened due to disorientation. Similarly, the hypothetical improvements missed by blindness (11) computes the average magnitude by which the true objective values could have been improved had the particles correctly replaced the personal best solutions with truly better solutions. Both indicators provide additional information about the effect of blindness and disorientation on the quality of the results.

The optimization curves (12) depict the average true objective values of the personal best solutions in the swarm at each iteration, thereby providing general information about the search process of the swarm to reach the quality of its results.

Lastly, the quality of results (13) consists of the set of true objective values of the best solutions found by the swarm over all the independent runs. The quality of results, together with the other population statistics, reflect the potential of the swarm to address the optimization problem at hand.

### 7.2.2 Single-Evaluation PSO Algorithms

Single-evaluation PSO algorithms spend the computational budget of function evaluations as the regular PSO dictates, that is, evaluating each solution only once at each iteration. Therefore, in optimization problems subject to noise, single-evaluation PSO algorithms need to assume that the
7.2. MAIN CONCLUSIONS

estimated objective values are very inaccurate, for which it is necessary to take actions towards preventing the effect of such inaccuracies on the quality of the results. The leverage of these algorithms with respect to the resampling-based PSO algorithms is that particles search over more iterations, potentially increasing the opportunities for finding better solutions.

The population statistics were computed for different single-evaluation PSO algorithms, and the results revealed that PSO-E, despite being the main reference in this type of algorithm, often yields significantly worse results than the regular PSO lacking a noise mitigation mechanism. The underlying reason to such a worse quality of results is that the evaporation mechanism causes particles to suffer from large proportions of disorientation, often leading the swarm to divergent behaviour. To overcome the divergent behaviour, the newly proposed PSO-PU algorithm utilizes update probabilities instead of evaporation factors. In doing so, the decision to replace the personal best solutions is made probabilistically at each unsuccessful iteration, thereby reducing the risk of divergence up to a certain extent. Moreover, the probabilistic updates provide a more intuitive and accurate mechanism of replacement, and an additional control to the tradeoff between exploration and exploitation of the swarm.

The quality of the results obtained with PSO-PU was generally better than that obtained with PSO-E. However, the regular PSO still provided a better quality of results than both of them, in part due to its minimal proportions of disorientation, but mostly due to the large proportions of disorientation present in PSO-E and PSO-PU. These findings raised awareness towards the effect of disorientation on the quality of the results. Even when disorientation is present in the algorithms at rather small proportions, its effects are devastating on the quality of the results, for which it is important to design noise mitigation mechanisms that prevent disorientation from increasing. Considering the quality of these results, we designed the new single-evaluation PSO-AN algorithm whose particles are partially attracted towards an average neighborhood best solution.
Particles in PSO-AN compute a new solution based on the centroid from the personal best solutions within the neighborhood. The goal is to create and select a new neighborhood best solution from the best solutions found by the swarm, and hence avoid the selection of a single neighborhood best solution based on very inaccurately estimated objective values. The idea of the centroid solution is based mostly on the fact that, without it, the selection of the neighborhood best solutions is rather arbitrary due to the inaccurately estimated objective values. Hence, we expected that the new centroid solutions could have truly better objective values than any other arbitrary solution selected from the neighborhoods. As indicated by the ranked deception, our expectations were generally correct at medium and high levels of noise, which in turn led to an overall better quality of results compared to that of the regular PSO.

7.2.3 Resampling-Based PSO Algorithms

Resampling-based PSO algorithms spend the computational budget re-evaluating the solutions multiple times in order to better estimate their objective values with a sample mean over the evaluations. Compared to single-evaluation PSO algorithms, the solutions found with resampling-based PSO algorithms have more accurate objective values at the cost of performing fewer iterations.

The population statistics computed for the different resampling-based PSO algorithms, two of them proposed in this thesis, showed that the basic PSO-ER generally finds a better quality of results than the regular PSO, mostly due to an important increment of regular operations (especially regular updates) and a decrement of blindness and ranked deception. However, particles from PSO-ER still suffer from large proportions of blindness (especially by memory) and binary deception, which strongly suggests that addressing these conditions could significantly improve the quality of its results. To this end, we developed the PSO-EER as an ex-
7.2. **MAIN CONCLUSIONS**

tension to PSO-ER, where the computational budget is also allocated between the personal best solutions to better estimate their objective values, thereby reducing the proportions of blindness and (both binary and ranked) deception. Since the computational budget remained the same, we had to balance its allocation between the current and personal best solutions, that is, we had to sacrifice the accuracy of the objective values from the current solutions to improve the accuracy of the objective values of the personal best solutions. The population statistics for PSO-EER1 (allocating one evaluation to the personal best solutions) confirmed our expectations of reducing blindness and disorientation, and thereby improving the quality of its results. The constant allocation of that additional evaluation to the personal best solutions also prolonged the lifetime of the swarm which, together with the reduction of blindness and disorientation, compensated for an increase of disorientation due to the now less accurately estimated objective values of the current solutions. However, as we experimented with PSO-EER2 (two evaluations to the personal best solutions) the proportions of disorientation largely increased due to the less accurately estimated objective values of the current solutions, which caused a significant deterioration of the quality of its results.

PSO-ERN is another resampling-based algorithm that we proposed, which instead of allocating evaluations to the personal best solutions, these are allocated between the estimated best two current solutions. In addition, each particle always replaces its personal best solution with any current solution that is better estimated. The goal behind this update rule was to reduce the proportions of blindness at the cost of increasing disorientation. PSO-ERN found a better quality of results than PSO-EER, but the population statistics were not entirely as expected. The proportions of blindness did not reduce, the disorientation increased, and the ranked deception was significantly improved. The blindness was not reduced because the objective values of most personal best solutions were still estimated based on five evaluations and only a few solutions based
on 30 evaluations. The disorientation increased because the update rule would replace the personal best solutions even with worse solutions as long as these have more evaluations. The ranked deception significantly improved because particles selected their neighborhood best solutions from the solutions with the most evaluations.

In spite of the improvements obtained with PSO-ERN, the PSO-OCBA algorithm was shown to be significantly superior, thus further consolidating the findings from previous works [7, 106, 154]. The PSO-OCBA algorithm was designed to asymptotically minimize the probability of binary deception by allocating the computational budget sequentially amongst the most important (current and personal best) solutions. In doing so sequentially, PSO-OCBA manages to re-estimate the objective values of the solutions before allocating further evaluations and, more importantly, PSO-OCBA manages to reduce the proportions of blindness and disorientation simultaneously. Even when the underlying operation of PSO-OCBA is more complex than any of the algorithms we proposed, both conceptually and computationally, such a complexity is surely worth of the significantly better quality of results.

### 7.2.4 Single-Evaluation vs Resampling-Based Algorithms

Single-evaluation and resampling-based PSO algorithms provide opposite tradeoffs. On the one hand, single-evaluation algorithms perform more iterations at the cost of handling very inaccurately estimated objective values. On the other hand, resampling-based algorithms better estimate the objective values of the solutions at the cost of performing less iterations. The quality of the results obtained with both approaches suggests that the proper accuracy tradeoff is problem-dependent.

Specifically, we found that PSO-AN yields a superior quality of results mostly on the benchmark functions primarily composed of ackley and rastrigin, which are certainly the most challenging to optimize
due to the little gradient information of their objective spaces and the nu-
merous local minima (respectively). However, the resampling-based PSO
algorithms yield a superior quality of results on the remaining bench-
mark functions. Overall, PSO-AN and the resampling-based PSO algo-
rithms significantly outperformed each other in about half of the bench-
mark functions, therefore suggesting that the performance of both ap-
proaches is problem dependent. Nonetheless, we consider it important to
further investigate the underlying characteristics of \textit{ackley} and \textit{rastrigin}
and their effect on single-evaluation and resampling-based PSO algorithms.

\section*{7.2.5 Hybrid PSO Algorithms}

The single-evaluation and resampling methods studied encouraged us to
create hybrid methods to be incorporated into PSO. Since the PSO-AN
yields the best quality of results amongst the single-evaluation PSO al-
gorithms, we decided to merge its centroid solution into the different re-
sampling-based PSO algorithms. The population statistics computed for
these hybrid algorithms were compared to those computed for their purely
single-evaluation and resampling-based counterparts, leading to the fol-
lowing conclusions.

The hybrid algorithms yield a significantly better quality of results than
their respective purely resampling-based counterparts. The similar pro-
portions of binary deception between hybrid and resampling-based algo-
rithms suggested that the centroid solution is generally not better than
the true best solution within the swarm. However, the significantly bet-
ter ranked deception of the hybrid algorithms showed that the centroid
solution is certainly better than any neighborhood best solution that par-
ticles from the resampling-based algorithms select. Another positive in-
dicator to such a superior quality of results was the proportions of reg-
ular updates, which increased significantly for the hybrids of PSO-EER1
and PSO-OCBA. Still, in all hybrid algorithms, the proportions of blind-
ness significantly increased due to selecting better neighborhood best solutions, a correlation that was also found when comparing PSO-ER and PSO-ERGC, but the underlying reason for such an increment of blindness remains as an objective for future research.

The hybrid algorithms not only found a better quality of results than the resampling-based PSO algorithms, but they were also mostly better than the quality of results obtained with PSO-ERGC. The significance of these findings is that they question the importance of using complex algorithms like PSO-OCBA to address optimization problems subject to noise. Specifically, even when the hybrid PSO-OCBA yields the best quality of results due to its better centroid solutions, the quality obtained with the hybrid PSO-EER1 was mostly not significantly different. Therefore, we consider PSO-EER1 a promising algorithm because its simplicity and flexibility allows straightforward modifications that could provide convenient to tune its general performance based on its population statistics.

Regarding the comparison to single-evaluation PSO algorithms, PSO-AN still managed to outperform the hybrid algorithms on some of the benchmark functions made up of ackley and rastrigin, which account for about a third of the cases. Considering the population statistics of lifetime for PSO-AN, we found that particles do not find better solutions in about half of the iterations, for which we decided to experiment allocating the computational budgets differently. The experiments were designed to compare the different allocations from converting PSO-AN to a hybrid PSO-EER1, but the quality of the results from PSO-AN remained superior on the benchmark functions made up of ackley and rastrigin. At this point, we consider it important to further investigate the characteristics of such functions that make them so challenging for the PSO algorithms.
7.3 Discussions

The previous section presented the main conclusions from each contribution chapter. In this section, we present further discussions on the thesis as a whole, focusing on the implications of the population statistics, the benchmark functions, and the computational budget of function evaluations.

7.3.1 Population Statistics

The population statistics have presented empirical evidence to support the analyses and discussions about the underlying operation of PSO throughout its search process. In addition, the population statistics have provided valuable information to characterize the performance of the swarms, thereby locating their weaknesses and strengths to the optimization problem at hand. As such, the population statistics have been an important guidance towards the design of better PSO algorithms.

The study of the population statistics throughout this thesis was mostly focused on understanding the underlying operation of different PSO algorithms, and also on finding interactions between the population statistics. However, towards the end of this thesis, we were able to predict (up to a certain extent) the trends of the population statistics for new PSO algorithms on the optimization problems. Furthermore, the population statistics for each algorithm showed mostly similar trends on the optimization problems, even when the composition of the problems was inherently different from each other. Nevertheless, we also came across exceptions on the problems primarily composed of ackley and rastrigin, whose objective spaces proved to be more challenging than those of the other problems.

The study of the population statistics revealed interactions between different factors. For example, we found an inverse correlation between the proportions of blindness and disorientation, increments of blindness
when the particles select better neighborhood best solutions, the importance of addressing blindness and disorientation before deception, amongst others. However, it is important to remark that we found such trends after multiple experiments carefully analyzing the results, for which we expect more trends to be found by utilizing data mining techniques on the vast amounts of data produced by the experiments. Furthermore, we expect that, eventually, the analysis of this data could lead to other population statistics that will be able to better differentiate the quality of the results between the algorithms, and thereby allow to finely tune them in order to yield a better quality of results.

Simplicity has been an important goal throughout this thesis, especially on the design of the new algorithms. In keeping the design of the algorithms simple, we have been able to accurately understand and discuss the underlying operation of the swarms. More importantly, considering the population statistics for PSO-E in Chapter 4, we have shown that, even for a simple modification to the PSO algorithm, the analyses based only on the quality of the results may miss taking into account other unintended effects. Therefore, besides calling for simplicity in future PSO algorithms, we consider it important to properly justify their creation on a set of population statistics and not only on the quality of their results.

7.3.2 Benchmark Functions vs Real-World Problems

The population statistics developed in this thesis can only be computed when the true objective values of the solutions are known. Consequently, the population statistics cannot be computed for real-world optimization problems, where not only the true objective values are not known, but probably the model of noise, its severity and even its direction may not be known in advance. However, the significance of the population statistics proposed to the PSO domain is not affected by such a limitation because the goal is not to compute them on real-world optimization problems, but
rather on benchmark functions where the PSO algorithms can be carefully tuned and prepared before testing their performance on the real-world problems. Precisely, the main goal of utilizing benchmark functions is to model the real-world problems up to a certain extent such that algorithms can be prepared in advance. More importantly, deception, blindness and disorientation are conditions from which the particles of PSO will suffer on any stochastic or dynamic optimization problem, for which the study of the population statistics in both types of problems could lead to new approaches to better address them.

The model of noise utilized throughout this thesis consists of sampling values from a Gaussian distribution, and each sampled value is a factor that multiplies the true objective values of the solutions. The main characteristics of this model of noise are the following. First, the multiplicative effect on the objective values of the solutions will cause a standard error larger than that caused by an additive effect given the same standard deviation of noise. Second, the standard error will change throughout the solution search space proportionally to the objective values, thereby varying the severity of noise according to the location of the solutions. Third, the minimization of the benchmark functions, together with the multiplicative effect, will cause the direction of the optimization problem to be backwards, for which better solutions will be less corrupted by noise. Lastly, thanks to the Gaussian model, the true objective values of the solutions can be approximated by a sample mean over $n$ evaluations, thereby reducing the standard error proportionally to $1/\sqrt{n}$.

The model of noise, its severity and direction are irrelevant with respect to the significance of the designed population statistics, but are fundamental for the analyses of the results obtained, and especially for the design and operation of the resampling-based PSO algorithms. A different model of noise, including effect, severity or direction, will not change the fact that particles will still suffer from deception, blindness and disorientation on noisy optimization problems, but changing either of these char-
acteristics will have different implications, some of which are discussed as follows.

**Noise distribution.** Different noise distributions will render the sample mean over multiple evaluations inaccurate because the mean will no longer be an accurate measure of central tendency. Examples can be found in the Cauchy, Poisson, Rayleigh, and Weibull distributions, in which resampling-based PSO algorithms utilizing the sample mean over the evaluations will estimate less accurately the objective values of their solutions, ultimately leading to different population statistics and a different quality of results.

**Effect of noise.** A convenient characteristic of multiplicative noise is that its effect scales directly across problems that we have addressed, whereas the additive noise is relative to the range of the function values of the problem at hand. For example, consider the impact of additive Gaussian noise with $\sigma = 1.0$ applied to a problem whose range of function values is between $[0.0, 100.0]$ and to another whose range is between $[0.0, 1.0]$. On average, the effect of noise in the former problem will be up to 1%, whereas that in the latter will be up to 100%. As such, the severity of additive noise will not necessarily imply similar challenges across different problems. Furthermore, if the effect of noise does not change across the search space, then a simpler approach could be to estimate the model of noise only once on any given solution, and then use that information to correct the estimated objective values of the other solutions throughout the search process.

**Severity of noise.** Different standard deviations $\sigma$ of the noise distribution will lead to different population statistics. Specifically, for larger values of $\sigma$, the quality of the results will deteriorate due to the larger range of the objective values to which the particles will be blinded. Likewise, the proportions of disorientation are bound to increase early in the search process due to the larger overlapping areas of the probability density functions depicted around the true objective values (as discussed in
7.3. DISCUSSIONS

Section 3.4.2 on page 75. Furthermore, for $\sigma \geq 0.34$, the corruption of the objective values will be already over 100\% of their true objective values, therefore providing PSO algorithms with little information about the problem to optimize.

Direction due to noise. The backward direction of the optimization problems is fundamental to the discussions presented throughout this thesis. The backward direction has been associated with the large proportions of blindness by memory, disorientation by the environment, and the general occurrence of larger proportions of disorientation early in the search process, all of which have been discussed in Section 3.4.2 (page 75). Therefore, making the optimization problem forwards will lead to different challenges as well as population statistics. For example, disorientation will become more frequent as particles further improve their solutions, thereby suggesting a probability of divergent behaviour. However, just as more disorientation occurs, the range of the objective values that potentially blind a particle will become larger, thereby suggesting that the inverse correlation between blindness and disorientation (discussed in Section 3.4.2) will still hold in forward optimization problems.

7.3.3 Computational Budget

The allocation of the computational budget of function evaluations presents an accuracy tradeoff to be decided upon. On the one hand, single-evaluation PSO algorithms do not spend the computational budget on re-evaluating the solutions multiple times, but instead utilize the computational budget to perform more iterations. As such, the swarms potentially increase their chance of finding better solutions at the cost of dealing with very inaccurately estimated objective values. One advantage of these algorithms is that they make no assumptions about the underlying noise distribution, for which these algorithms are, to some extent, agnostic to the model of noise. At least, that is the case of PSO-AN, whose quality of results is sig-
significantly better than that of the other single-evaluation PSO algorithms studied, even with its much larger proportions of blindness. The other single-evaluation algorithms yield a significantly worse quality of results, but still managed to successfully reduce the proportions of blindness from which the particles suffer, for which we still consider PSO-E and PSO-PU to be very relevant algorithms for future research.

On the other hand, resampling-based PSO algorithms re-evaluate the solutions multiple times in order to better estimate their true objective values by a sample mean over the evaluations. Thus, these algorithms sacrifice the number of iterations over improving the accuracy of the objective values. The performance of these algorithms is mostly better than that of single-evaluation PSO algorithms, but each yields better results on different problems. Specifically, the single-evaluation PSO-AN was able to find better solutions on those problems made up of ackley and rastrigin, both of which have the most challenging objective spaces from the problems considered. Other than those, resampling-based PSO algorithms find significantly better solutions, especially PSO-OCBA despite its computational complexity.

The resampling-based PSO algorithms studied and designed in this thesis work under the assumption that the noise distribution is Gaussian, which is a reasonable assumption \cite{89,90,91}. However, changing the distribution of noise may cause the sample mean not to reflect an accurate measure of central tendency, in which case the true objective values will be estimated inaccurately. To overcome this disadvantage, it is necessary to define a criterion to compare two random variables, that is, a criterion to compare the estimated objective values of the current and the personal best solutions based on their respective sets of evaluations. In doing so, particles will be able to decide what solutions are better throughout the search process.

The computational budget of 30,000 function evaluations is large enough for the following swarms of 50 particles, whose average lifetime (in paren-
theses) suggests that they generally reach convergence: regular PSO (58.58%), PSO-ER (63.97%), and PSO-AN (54.53%). However, the much longer lifetime of the resampling-based PSO algorithms, namely PSO-ERN (88.89%), PSO-EER1 (93.58%), PSO-EER2 (94.63%), and PSO-OCBA (90.47%), suggests that these algorithms could benefit greatly from a larger computational budget such that they can perform more iterations and hence potentially reach convergence. Likewise, the hybrid PSO algorithms could benefit from a larger computational budget given their lifetime: PSO-EER1\(^5\) (95.19%), PSO-ERN\(^5\) (87.72%), and PSO-OCBA\(^5\) (91.43%). Regarding PSO-E and PSO-PU, even when these swarms have an even longer lifetime (both over 97%), we discourage from utilizing a larger computational budget for them due to the risk of divergent behaviour that these swarms pose.

Another option to differently allocate the computational budget, besides single-evaluation and resampling-methods, is to change the number of particles in the swarm. While such an approach has been used before in Genetic Algorithms with successful results [52, 62], increasing the size of the swarm will not change the fact that more particles will suffer from deception, blindness and disorientation. However, since more particles cover more regions of the search space, it is still possible that increasing the size of the swarm will lead to a better quality of results.

7.4 Future Work

Thus far, this thesis has presented the most complete work in the literature on the performance of PSO on optimization problems subject to noise. However, we envision the following objectives could also provide valuable contributions to the PSO domain.
7.4.1 Population Statistics

The population statistics have been useful to characterize the performance of different PSO algorithms on optimization problems subject to noise. However, the population statistics need not be exclusive to such problems, but rather could serve as general indicators of the performance of the swarms throughout the search process regardless of the type of problem. Therefore, we find it important for future research to design other population statistics such that the discussions of the underlying operation of PSO can be supported on more solid grounds other than just on the quality of the results. Furthermore, we expect that the study of the population statistics will not only provide support to justify the creation of new variants of the PSO algorithm, but will also promote simplicity in their design in order to be able to understand and properly discuss the underlying operation of the swarms. In addition, we expect the population statistics and the simplicity of the algorithms to enhance the reproducibility of the results.

Further research could explore the computational complexity of the algorithms theoretically via big $O$ notation, but an empirical approach is also possible via new population statistics measuring the computational time that particles take to operate. Moreover, the population statistics could be further studied under different computational budgets to evaluate the effect of different accuracy trade-offs. Future works could also explore the population statistics for different parameter settings of PSO such as the inertia and acceleration coefficients of the particles, velocity update rules, network topologies, and mechanisms for the selection of the neighborhood best solutions.

The population statistics provide large amounts of information that may be daunting not only to be presented in proper formats, but also to find correlations amongst the different indicators. Therefore, we expect future works to explore different visualization options and to consider techniques from data mining to extract useful information. In particular, some
potential approaches are correlograms [55] to visualize correlations, correlation coefficients between the statistics to identify their relations, and feature ranking [64] to weight the importance of the statistics with respect to the quality of the results.

The population statistics have provided important guidance and control of PSO up to a certain extent, for which we consider it useful to explore the relevance of population statistics for other population-based algorithms such as Ant Colony Optimization [39], Differential Evolution [117] and Genetic Algorithms [66].

7.4.2 Benchmark Functions

The set of benchmark functions utilized throughout this thesis presents multiple challenges with different characteristics that helped a better exploration of the performance of the PSO algorithms. However, we still find potential improvements to be made by investigating the effect of the separability of the benchmark functions, and especially the characteristics that make functions composed of ackley and rastrigin to be so challenging to the PSO algorithms. Further research could also consider experimenting on the extended set of benchmark functions recently published in [81] that includes more functions and challenges to optimization algorithms.

Another potential area of research is the study of the population statistics on forward optimization problems and the differences therein with respect to backward optimization problems. As discussed before, we expect particles to suffer more from disorientation as they improve the solutions, which could lead to divergent behaviour, but we also expect the inverse correlation between blindness and disorientation to hold. Thus, future works could validate empirically our expectations and perhaps find other characteristics and correlations between the population statistics.

Two other potential areas of research are on dynamic optimization prob-
CHAPTER 7. CONCLUSIONS

problems [16, 80, 99, 100, 101, 132, 151] and on multi-objective noisy or dynamic optimization problems [20, 59, 60, 128, 142]. In dynamic optimization problems, the particles will also suffer from deception, blindness and disorientation, and the population statistics therein could provide a new approach to address such problems. Regarding multi-objective optimization [27, 28, 34, 41, 110, 130], just the fact of dealing with multiple objectives will surely create a whole new set of different challenges to be addressed.

7.4.3 Single-Evaluation PSO Algorithms

In single-evaluation PSO algorithms, future research could be directed towards the study of the population statistics for alternative mechanisms to those proposed in Chapter 4. In addition, considering that Genetic Algorithms are noise invariant when utilizing roulette-wheel selection [90, 91], we would find it interesting to study the population statistics for a similar selection mechanism on the neighborhood best solutions. Other studies could address the population statistics for dynamic evaporation factors [51] and heterogeneous swarms with evaporation [125].

Another single-evaluation approach worth studying is to change the size of the swarm, thereby sacrificing iterations over coverage of the search space and hence allocating differently the computational budget. Such an approach has been used in Genetic Algorithms to mitigate the effect of noise [52, 62], but have in mind that the challenges therein are different from those in PSO.

7.4.4 Resampling-Based PSO Algorithms

The allocation of the computational budget provides different tradeoffs regarding the accuracy of the estimated objective values and the number of iterations performed. While our experiments in Chapter 6 show that the quality of the results can be improved by allocating more evaluations to the solutions, we expect there to be a limit where the trend will reverse
7.4. FUTURE WORK

given the fewer iterations performed. Therefore, future works could explore the limits where the quality of the results deteriorates due to performing fewer iterations as a result of better estimating the objective values of the solutions.

The PSO-OCBA is, without a question, the best resampling-based PSO algorithm, at least from those studied. Future works could explore the performance of PSO-OCBA according to different computational budgets in terms of the number of evaluations equally allocated at the first stage, and the number of evaluations allocated sequentially. Another idea worth exploring is to incorporate the extended OCBA to select an optimal set of solutions [24], where the goal is to maximize the probability of correctly selecting the best $k$ neighborhood solutions in the swarm, for which neighborhoods could then be dynamically defined around the $k$ solutions such that each neighborhood is attracted towards a potential best solution. In general, other resampling methods [24, 40, 76, 131, 138] could be worth considering for incorporation into PSO.

The computational budget of function evaluations has been equally distributed across the algorithms examined in this thesis, that is, the algorithms always have the same number of evaluations to allocate at each iteration. However, future works could explore different distributions of the computational budget across the iterations. For example, considering the general low quality of the initial solutions, perhaps multiple evaluations could be saved earlier during the search process to spend them later. Moreover, we expect the direction of optimization problem to be a factor towards deciding the distribution of the computational budget across the iterations. Specifically, in backward optimization problems, the quality of the solutions will be more corrupted by noise earlier during the search, for which more evaluations could be allocated then to better estimating these solutions and direct the swarm towards better regions of the search space. Differently, in forward optimization problems, a better strategy could be to allocate more evaluations later during the search. Some related works
in this domain are \cite{22, 111, 112, 135, 140}.

As mentioned before, the resampling-based PSO algorithms studied in this thesis work under the assumption that the distribution of noise is Gaussian, and therefore the mean over the set of evaluations performed to any given solution represents an accurate measure of central tendency towards its true objective value. However, changing the noise distribution will have a negative effect on the algorithms given that the mean may no longer be an accurate statistic when the sample size is small. To handle different noise distributions, we consider the concepts of First- and Second-order Stochastic Dominance \cite{78, 79} to be a promising incorporation into the particles of PSO. Stochastic dominance in PSO will allow particles to compare the current and personal best solutions based on the cumulative distributions estimated from the sets of evaluations. As such, if the estimated cumulative distribution for the current solution stochastically dominates that for the personal best solution, the particle decides to update.

Another interesting approach to compare the current and personal best solutions of a particle is to use Fuzzy Logic \cite{152, 153} and treat their sets of evaluations as fuzzy numbers each. Then, the natural order of the fuzzy numbers can be determined using a variety of different methods available in the literature of fuzzy ranking \cite{13, 85}. Thus, particles decide to update their personal best solutions based on the natural order of the fuzzy numbers that represent the sets of estimated objective values.

### 7.4.5 Hybrid PSO Algorithms

The hybrid PSO algorithms explored in this thesis merged the resampling-based PSO algorithms with the centroid solution of PSO-AN. However, another potential opportunity for improvement is to merge resampling-based PSO algorithms with the evaporation mechanism, where the evaporation factors are determined dynamically according to the estimated dis-
tribution of the objective values. Moreover, since the resampling method provides an estimated distribution of the objective values, we expect that hybrids with the additive evaporation mechanism will become relevant given that it will provide a uniform decay over the range of objective values, whereas the decay caused by multiplicative evaporation will be exponential. As with evaporation, hybrid algorithms could also be created with update probabilities.

Another potential approach to hybrid PSO algorithms is to utilize heterogeneous swarms \[33, 45, 46, 95, 136\], where some particles are configured to utilize single-evaluation methods and some others to utilize resampling methods. In addition, we expect that the next step to hybrid heterogeneous swarms will be to have particles decide between the methods during the search process utilizing different strategies \[77, 97, 98, 149\].
Bibliography


[100] NGUYEN, T., AND YAO, X. Evolutionary optimization on continuous dynamic constrained problems - an analysis. In Evolutionary Computation for Dynamic Optimization Problems, S. Yang and X. Yao,


[140] Syberfeldt, A., Ng, A. H. C., John, R. I., and Moore, P. Evolutionary optimisation of noisy multi-objective problems using


[147] **Wang, L., and Singh, C.** Stochastic combined heat and power dispatch based on multi-objective particle swarm optimization. *In-


