UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Niching in Particle Swarm Optimization

by

Isabella Lodewina Schoeman

Submitted in partial fulfillment of the requirements for the degree
Philosophiae Doctor
in the Faculty of Engineering, Built Environment and Information Technology
University of Pretoria, Pretoria

June 2010

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Niching in Particle Swarm Optimization

by

Isabella Lodewina Schoeman
E-mail: lona.schoeman@gmail.com

Abstract

Optimization forms an intrinsic part of the design and implementation of modern systems, such as industrial systems, communication networks, and the configuration of electric or electronic components. Population-based single-solution optimization algorithms such as Particle Swarm Optimization (PSO) have been shown to perform well when a number of optimal or suboptimal solutions exist. However, some problems require algorithms that locate all or most of these optimal and suboptimal solutions. Such algorithms are known as niching or speciation algorithms.

Several techniques have been proposed to extend the PSO paradigm so that multiple optima can be located and maintained within a convoluted search space. A significant number of these implementations are subswarm-based, that is, portions of the swarm are optimized separately. Niches are formed to contain these subswarms, a process that often requires user-specified parameters, as the sizes and placing of the niches are unknown. This thesis presents a new niching technique that uses the vector dot product of the social and cognitive direction vectors to determine niche boundaries. Thus, a separate niche radius is calculated for each niche, a process that requires minimal knowledge of the search space. This strategy differs from other techniques using niche radii where a niche radius is either required to be set in advance, or calculated from the distances between particles.

The development of the algorithm is traced and tested extensively using synthetic benchmark functions. Empirical results are reported using a variety of settings. An analysis of the algorithm is presented as well as a scalability study. The performance of the algorithm is also compared to that of two other well-known PSO niching algorithms. To conclude, the vector-based PSO is extended to locate and track multiple optima in dynamic environments.

Keywords: Niching, speciation, particle swarm optimization, dynamic.

Supervisor : Prof. A. P. Engelbrecht
Department : Department of Computer Science
Degree : Philosophiae Doctor

Niching in Particle Swarm Optimization

by

Isabella Lodewina Schoeman
E-mail: lona.schoeman@gmail.com

Opsomming

Optimering vorm 'n intrinsieke deel van die ontwerp en implementering van moderne stelsels soos industriële stelsels, kommunikasienetwerke en die konfigurasie van elektriese of elektroniese komponente. Populasiegebaseerde, enkel-oplossing optimeringsalgoritmes soos Partikel Swerm Optimering (PSO) het reeds getoon dat dit goeie resultate lewer indien 'n aantal optimale of suboptimale oplossings bestaan. Sommige probleme vereis egter algoritmes wat al hierdie optimale en suboptimale oplossings, of die meeste daarvan, opspoor. Sulke algoritmes staan bekend as algoritmes vir nisvorming of spesiasie.

Verskeie tegnieke is voorgestel om die PSO paradigma uit te brei sodat veelvuldige optima opgespoor en onderhou kan word in 'n soekruimte met talle hoogtepunte of pieke. 'n Betekenisvolle aantal van sulke implementasies is subswerm-gebaseerd, wat beteken optimering van gedeeltes van die swerm vind afsonderlik plaas. Nisse word gevorm om hierdie subswerms in te sluit, 'n proses wat dikwels gebruiker-gedefinieerde parameters benodig, aangesien die groottes en posisionering van die nisse nie bekend is nie. Hierdie tesis stel 'n nuwe tegniek vir nisvorming voor wat die vektor dotproduk van die sosiale en kognitiewe rigtingvektore gebruik om nisgrense te bepaal. Sodoende word 'n afsonderlike nisradius vir elke nis bereken, 'n proses wat minimale kennis van die soekruimte vereis. Hierdie strategie verskil van ander tegnieke wat nisradiusse gebruik waar 'n nisradius òf vooraf gestel word, of van die afstande tussen partikels bereken word.

Die ontwikkeling van die algoritme word nagespeur en uitvoerig getoets deur van kunsmatige toetsfunksies gebruik te maak. Empiriese resultate met 'n verskeidenheid stellings word gerapporteer. 'n Ontleding van die algoritme word aangebied sowel as 'n skalingstudie. Die prestasie van die algoritme word ook vergelyk met dié van twee ander bekende PSO nisvormingalgoritmes. Ten slotte word die vektor-gebaseerde PSO uitgebrei om veelvuldige optima in dinamiese omgewings op te spoor en te volg.

Acknowledgements

I would like to sincerely thank the following people who were instrumental in the completion of this dissertation:

- Professor Andries Engelbrecht for his enthusiasm, inspiration and assistance during the development and completion of this thesis;
- My family, friends and colleagues for their patience and support.

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Contents

1 Introduction **15**

1.1 Motivation . . . 16

1.2 Objectives . . . 17

1.3 Methodology . . . 17

1.4 Contributions . . . 18

1.5 Thesis outline . . . 19

2 Optimization **21**

2.1 Introduction . . . 21

2.2 Mathematical optimization . . . 22

2.2.1 Basic concepts . . . 22

2.2.2 Maximization . . . 23

2.2.3 Problem classification . . . 23

2.2.4 Mathematical optimization methods . . . 27

2.3 Stochastic local search algorithms . . . 29

2.4 Optimization in nature . . . 30

2.5 Computing paradigms inspired by natural optimization processes . . . 32

2.6 Conclusion . . . 34

3 Particle Swarm Optimization **35**

3.1 Introduction . . . 35

3.2 Particle swarm optimization: The algorithm . . . 36

3.2.1 Global and local particle swarm optimizers . . . 38

3.2.2 Cognition-only and social-only models . . . 39

3.2.3 The inertia weight . . . 40

3.2.4 The constriction factor 43
3.2.5 Neighbourhoods 44
3.3 Variations on the particle swarm optimizer 47
3.3.1 Information sharing strategies 48
3.3.2 Subswarm-based approaches 51
3.3.3 Memetic PSO algorithms 52
3.4 Conclusion 54

4 Niching **55**
4.1 Introduction 55
4.2 Genetic algorithm niching techniques 56
4.2.1 A sequential niching technique 57
4.2.2 A species conserving genetic algorithm 61
4.3 PSO niching techniques 63
4.3.1 Objective function stretching 64
4.3.2 The nBest PSO 65
4.3.3 NichePSO 67
4.3.4 The species-based PSO 70
4.3.5 The adaptive niching PSO 73
4.3.6 The fitness Euclidian-distance ratio-based PSO 74
4.3.7 The waves of swarm particles algorithm 76
4.4 Conclusion 77

5 Vector-Based PSO **78**
5.1 Introduction 78
5.2 Objectives of this chapter 79
5.3 Using vector properties 80
5.4 Identifying niches 88
5.5 Vector-based PSO algorithms 92
5.5.1 The sequential vector-based PSO 92
5.5.2 The parallel vector-based PSO 96
5.5.3 The enhanced parallel vector-based PSO 101
5.6 Conclusion 104

6 Vector-Based PSO Applied to Static Environments **106**
6.1 Introduction . . . 106
6.2 Experimental procedure . . . 107
6.2.1 General procedures and settings . . . 107
6.2.2 Statistical procedures . . . 110
6.2.3 Test functions . . . 111
6.2.4 Initial swarm sizes and granularity . . . 117
6.3 Results of the sequential vector-based PSO . . . 122
6.4 Results for the parallel and enhanced parallel vector-based PSO . . . 128
6.4.1 Test results . . . 128
6.4.2 Tracking merging of niches . . . 145
6.4.3 Comparing the parallel and enhanced parallel algorithms . . . 154
6.5 Analysis of the vector-based particle swarm optimizer . . . 156
6.5.1 Analysis of the VBPSO on additional functions . . . 157
6.5.2 Sensitivity of the vector-based PSO to granularity . . . 168
6.5.3 Scalability of the vector-based PSO . . . 176
6.6 A comparative study of three PSO niching approaches . . . 194
6.6.1 Experimental procedure . . . 194
6.6.2 Results . . . 195
6.6.3 Discussion . . . 195
6.7 Conclusion . . . 196

7 Dynamic Vector-Based PSO **199**
7.1 Introduction . . . 199
7.2 Dynamic optimization problems . . . 200
7.3 Changes in the environment . . . 201
7.4 Particle swarm models for tracking a single solution in a dynamic environment 202
7.5 Tracking multiple optima in a dynamic environment . . . 204
7.5.1 A dynamic test function generator . . . 205
7.5.2 PSO models for multiple dynamic optima . . . 205
7.6 The dynamic vector-based particle swarm optimizer . . . 207
7.7 Conclusion . . . 211

8 The Vector-Based PSO Applied to Dynamic Environments **212**

8.1 Introduction 212

8.2 Experimental setup and results 213

8.3 Comparing the performance of the dynamic vector-based PSO to that of the dynamic species-based PSO 229

8.3.1 Experimental settings and results 229

8.3.2 Discussion 233

8.4 Sensitivity of the dynamic vector-based PSO to changes in severity 233

8.5 Conclusion 238

9 Conclusion **239**

9.1 Summary 239

9.2 Future work 243

References **244**

A Test Results **254**

A.1 Minimum subswarm sizes 254

A.1.1 Experimental procedure 255

A.1.2 Results and discussion 255

A.2 Merging intervals 257

A.3 Minimum swarm size for tracking optima 259

A.3.1 Experimental procedure 259

A.3.2 Results and discussion 260

B Derived publications **261**

List of Figures

2.1 Two-dimensional linear programming problem 24

2.2 Function of single variable with minimum at x^* 25

2.3 Function of two variables with minimum at x^* 25

2.4 Function of a single variable with global minimum at x^* 26

3.1 Stochastic particle trajectory, obtained using $w = 0.7$ and $c_1 = c_2 = 1.4$ [100] . 42

3.2 Social network structures 46

4.1 Suppression of one peak by a derating function 60

4.2 Determining the species seeds from the population at each iteration [57] 71

5.1 The inverted one-dimensional Rastrigin function, showing particles with associated vectors 82

5.2 Regions of positive and negative dot products for the inverted one-dimensional Rastrigin function 85

5.3 Regions of positive and negative dot products for equation (??) 85

5.4 The two-dimensional Ursem F1 function 87

5.5 Contour map of the two-dimensional Ursem F1 function 88

6.1 One-dimensional functions 112

6.2 The Himmelblau function showing maxima. 118

6.3 The Griewank function. 118

6.4 The Rastrigin function. 119

6.5 The Ackley function. 119

6.6 The Ursem F1 function. 120

6.7 The Ursem F3 function. 120

6.8 The six hump camel function. 121
6.9 Merging of one-dimensional functions 147
6.10 Merging of two-dimensional functions - part 1 152
6.11 Merging of two-dimensional functions - part 2 153
6.12 The Styblinski-Tang function showing maxima 157
6.13 The Bird function showing maxima 158
6.14 The generalized Schwefel function 2.26 showing maxima 161
6.15 The tabular holder function 162
6.16 The tube holder function. 165
6.17 Number of optima vs. granularity for the Himmelblau function 169
6.18 Interniche distances for the Himmelblau function 170
6.19 Number of optima vs. granularity for the Griewank function 171
6.20 Interniche distances for the Griewank function 172
6.21 Number of optima vs. granularity for the Rastrigin function 173
6.22 Interniche distances for the Rastrigin function 175
6.23 The absolute Sine function in one and two dimensions 177
6.24 The Griewank function in one and two dimensions 177
6.25 The Rastrigin function in one and two dimensions 178
6.26 % Optima found versus swarm sizes for the absolute Sine function 182
6.27 % Optima versus swarm sizes for the Griewank function 186
6.28 % Optima versus swarm sizes for the Rastrigin function 190
6.29 Optimal swarm sizes versus actual number of optima for three benchmark functions 193
6.30 Comparing performance of three niching strategies for seven functions. 198

7.1 An environment with three peaks at positions $(-0.5, -0.7)$, $(-0.5, 0.5)$ and $(0.5, 0)$, generated by Morrison and De Jong's test problem generator 206

8.1 Scenario 1: Function landscapes 217
8.2 Scenario 2: Function landscapes 218
8.3 Scenario 3: Function landscapes 219
8.4 Scenario 4: Function landscapes 220
8.5 Movement of one peak through a landscape containing 5 peaks 235
8.6 Average #optima versus severity for 5-peak function 237

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

List of Tables

5.1 Dot products of a range of particle positions for the inverted one-dimensional Rastrigin function . . . 84

5.2 Dot products of a range of particle positions for equation (??) . . . 86

6.1 Positions of maxima of one-dimensional functions . . . 111

6.2 Optima of the Himmelblau function . . . 113

6.3 Optima of the Griewank function . . . 114

6.4 Optima of the Rastrigin function . . . 115

6.5 Optima of the Ackley function . . . 115

6.6 Optima of the Ursem F1 function . . . 116

6.7 Optima of the Ursem F3 function . . . 117

6.8 Optima of the Six Hump Camel function . . . 117

6.9 Initial swarm sizes and granularity for test functions . . . 121

6.10 Sequential vector-based PSO results for one-dimensional functions $F1$ and $F2$. 124

6.11 Sequential vector-based PSO results for one-dimensional functions $F3$ and $F4$. 125

6.12 Sequential vector-based PSO results for the Himmelblau function . . . 126

6.13 Results of the Mann-Whitney U test. Success rates of the sequential VBPSO using a system-supplied random number generator are compared to those using Sobol sequences. . . . 127

6.14 Parallel vector-based PSO results for one-dimensional functions $F1$ and $F2$ . . 130

6.15 Parallel vector-based PSO results for one-dimensional functions $F3$ and $F4$ . . 131

6.16 Enhanced parallel vector-based PSO results for one-dimensional functions $F1$ and $F2$ . . . 132

6.17 Enhanced parallel vector-based PSO results for one-dimensional functions $F3$ and $F4$ . . . 133

6.18 Summary of ranks based on the Mann-Whitney U test for comparing success rates of algorithms using a system-supplied random number generator to those using Sobol sequences . . . 134
6.19 Results of the Mann-Whitney U test. Success rates of the parallel and enhanced parallel VBPSO algorithms using a system-supplied random number generator are compared to those using Sobol sequences . . . 135
6.20 Parallel and enhanced parallel vector-based PSO results for the Himmelblau function . . . 136
6.21 Parallel and enhanced parallel vector-based PSO results for Griewank function 137
6.22 Parallel and enhanced parallel vector-based PSO results for Rastrigin function 138
6.23 Parallel and enhanced parallel vector-based PSO results for Ackley function . . 139
6.24 Parallel and enhanced parallel vector-based PSO results for Ursem $F1$ function 140
6.25 Parallel and enhanced parallel vector-based PSO results for Ursem $F3$ function 141
6.26 Parallel and enhanced parallel vector-based PSO results for the six hump camel function . . . 142
6.27 Summary of ranks based on the Mann-Whitney U test for comparing success rates of algorithms using a system-supplied random number generator to those using Sobol sequences . . . 143
6.28 Results of the Mann-Whitney U test. Success rates of the parallel and enhanced parallel VBPSO algorithms using a system-supplied random number generator are compared to those using Sobol sequences . . . 144
6.29 Merging of niches for one-dimensional functions $F1$ to $F4$ . . . 146
6.30 Merging of niches using the parallel VBPSO - part 1 . . . 148
6.31 Merging of niches using the parallel VBPSO - part 2 . . . 149
6.32 Merging of niches using the enhanced parallel VBPSO - part 1 . . . 150
6.33 Merging of niches using the enhanced parallel VBPSO - part 2 . . . 151
6.34 Average % optima located for functions $F1$ to $F4$ . . . 154
6.35 Summary of VBPSO success rates for two-dimensional functions . . . 155
6.36 Comparing algorithms for functions $F1$ to $F4$ . . . 155
6.37 Comparing the enhanced parallel VBPSO to the parallel VBPSO . . . 156
6.38 Optima of the Styblinksi-Tang function . . . 158
6.39 Optima of the Bird function . . . 159
6.40 VBPSO results for Styblinksi-Tang function . . . 160
6.41 VBPSO results for the Bird function . . . 160

6.42 Optima of the generalized Schwefel function 2.26 162
6.43 Optima of the tabular holder function in the range $x_1, x_2 \in [-4.5, 4.5]$ 163
6.44 VBPSO results for generalized Schwefel function 2.26 164
6.45 VBPSO results for tabular holder function 164
6.46 Optima of the tube holder function in the range [-3,4] and [3,4] 166
6.47 Optima of the tube holder function in the range [-3,4] and [3,4] 166
6.48 VBPSO results for tube holder function 167
6.49 Testing granularity for the Himmelblau function 169
6.50 Interniche distances for the Himmelblau function 170
6.51 Testing granularity for the Griewank function 171
6.52 Interniche distances for the Griewank function 172
6.53 Testing granularity for the Rastrigin function 173
6.54 Interniche distances for the Rastrigin function. 174
6.55 Average number of solutions versus swarm sizes for the absolute Sine function - part 1 179
6.56 Average number of solutions versus swarm sizes for the absolute Sine function - part 2 180
6.57 Average number of solutions versus swarm sizes for the absolute Sine function - part 3 181
6.58 Average number of solutions versus swarm sizes for the Griewank function - part 1183
6.59 Average number of solutions versus swarm sizes for the Griewank function - part 2184
6.60 Average number of solutions versus swarm sizes for the Griewank function - part 3185
6.61 Average number of solutions versus swarm sizes for the Rastrigin function . . . 187
6.62 Average number of solutions versus swarm sizes for the Rastrigin function - part 2188
6.63 Average number of solutions versus swarm sizes for the Rastrigin function - part 3189
6.64 Minimum swarm sizes required to locate 98% of possible optima for the absolute Sine function 191
6.65 Minimum swarm sizes required to locate 98% of possible optima for the Griewank function 192
6.66 Minimum swarm sizes required to locate 98% of possible optima for the Rastrigin function 192
6.67 Comparing three niching algorithms 197

8.1 Scenario 1: Positions of 3 optima over 6 steps 216