

Theory and Applications of Bio-inspired Algorithms

A DISSERTATION PRESENTED
BY
MARKUS WAGNER
TO
THE SCHOOL OF COMPUTER SCIENCE
IN FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE SUBJECT OF
COMPUTER SCIENCE

THE UNIVERSITY OF ADELAIDE
ADELAIDE, SOUTH AUSTRALIA
JULY 2013

©2013 – MARKUS WAGNER
ALL RIGHTS RESERVED.

Thesis advisors:
Associate Prof. Frank Neumann
Prof. Zbigniew Michalewicz

Markus Wagner

Theory and Applications of Bio-inspired Algorithms

ABSTRACT

Evolutionary algorithms, which form a sub-class of bio-inspired algorithms, mimic some fundamental aspects of the neo-Darwinian evolutionary process. They simultaneously search with a *population* of candidate solutions and associate an objective score as a fitness value for each one. The algorithms then select among the population to favour those solutions that are more fit. The next generation (i.e. a new population) consists of replicates of the fitter solutions that have been *genetically mutated and crossed over* in a biological metaphor: the decision variables were perturbed such that they inherit characters of their *parents*, as well as change in random ways.

For the past decades, the algorithms' success has led to strongly practical-oriented interests. Although the theory of them is far behind the knowledge gained from experiments, there are theoretical investigations about some of their properties. This thesis spans theoretical investigations, theory-motivated algorithm engineering, and also the real-world application of evolutionary algorithms.

First, we analyse different algorithms that work with solutions of variable length. We show theoretically and experimentally that certain design choices can have drastic impacts on the ability of an algorithm to find optimal solutions.

Second, motivated by recent theoretical investigations, we design a framework for solving problems with conflicting objectives. We demonstrate that it can efficiently handle problems with many such objectives, which most existing algorithms have difficulties dealing with.

Finally, we consider the problem of maximising the energy yield of wind farms. Our problem-specific algorithm achieves higher quality results than existing approaches, and it allows for an optimisation within minutes or hours instead of days or weeks.

Contents

1	INTRODUCTION	1
1.1	Part I: Computational Complexity of Variable-Length Algorithms	2
1.2	Part II: Design of Evolutionary Multi-Objective Algorithms	3
1.3	Part III: Applications to Wind Farm Optimisations	5
I	Computational Complexity of Variable-Length Algorithms	7
2	ALGORITHMS AND METHODS FOR THEIR ANALYSIS	9
2.1	Motivation	9
2.2	Problems	11
2.3	Algorithms	15
2.4	Methods for the Analysis	18
3	THEORETICAL ANALYSIS OF SORTING	21
3.1	Standard Approach Without Bloat-Control	21
3.2	Local Optima and the Parsimony Approach	25
3.3	Multi-Objective Approach	30
3.4	Experimental Supplements	34
3.5	Conclusions	47
4	EXPERIMENTAL ANALYSIS OF ORDER AND MAJORITY	51
4.1	Introduction	51
4.2	Preliminaries	52
4.3	(1+1)-GP	54
4.4	SMO-GP	55
4.5	Conclusions	59

II	Design of Evolutionary Multi-Objective Algorithms	61
5	EVOLUTIONARY MULTI-OBJECTIVE OPTIMISATION	63
5.1	Introduction	64
5.2	Preliminaries	67
6	APPROXIMATION-GUIDED EVOLUTION	71
6.1	Simple Algorithm	71
6.2	Fast Algorithm	73
6.3	Experimental Study	74
6.4	Conclusions	77
7	APPROXIMATION-GUIDED EVOLUTION II	79
7.1	The Algorithm	80
7.2	Speed-up through approximative archives	83
7.3	Benchmark Results	84
7.4	Conclusions	86
III	Applications to Wind Farm Optimisation	91
8	WIND FARM OPTIMISATION	93
8.1	Introduction	93
8.2	State of the Art	94
8.3	Wake Modelling	96
9	OPTIMISING THE LAYOUT OF 1000 WIND TURBINES	101
9.1	CMA-Evolutionary Strategy	101
9.2	Results and Discussion	104
9.3	Conclusions	108
10	A PROBLEM SPECIFIC LOCAL SEARCH METHOD	111
10.1	Turbine Distribution Algorithm	112
10.2	Experimental Investigations	115
10.3	A Real-World Problem: Dealing With Infeasible Areas	119
10.4	Conclusion	121
IV	Summary and Future Work	123
	REFERENCES	127

Listing of Figures

2.1	WORDER and WMAJORITY: example for evaluations	12
3.1	SORTING: maximum tree sizes observed when init_0 is used	39
3.2	SORTING: maximum tree sizes observed when init_n is used	40
3.3	SORTING: number of evaluations required when init_0 is used	41
3.4	SORTING: number of evaluations required when init_n is used	42
3.5	SORTING: maximum population size for INV in SMO-GP	45
3.6	SORTING: number of evaluations required until an optimum is found .	46
4.1	WORDER/WMAJORITY: number of evaluations required until an optimum is found	56
4.2	WORDER/WMAJORITY: number of evaluations required until the optimum and the entire true Pareto front is found	58
5.1	Multi-objective optimisation: example objective space	64
6.1	Visualization of the Pareto fronts for $d = 3$	75
6.2	Performance comparison: our algorithm AGE with IBEA, NSGA-II, SMS-EMOA and SPEA2	75
6.3	Performance comparison: our algorithm AGE with IBEA, NSGA-II, SMS-EMOA and SPEA2	76
7.1	Visualisation of the ϵ -approximation in the objective space.	82
7.2	Influence of ϵ_{grid} on the archive size, the runtime, and the final quality .	88
7.3	Performance comparison: our AGE-II with the original AGE, IBEA, NSGA-II, SMS-EMOA and SPEA2	89
7.4	Performance comparison: our AGE-II with the original AGE, IBEA, NSGA-II, SMS-EMOA and SPEA2	89
7.5	Performance comparison: our AGE-II with the original AGE, IBEA, NSGA-II, SMS-EMOA and SPEA2	90
9.1	Comparison of Kusiak et al.'s algorithm SPEA-2 and our adjusted CMA-ES	106

9.2	Performance of CMA-ES for 10 to 100 turbines under Scenario 2	106
9.3	Performance of CMA-ES for 200 to 1000 turbines under Scenario 2 . . .	107
9.4	Ratio of energy loss due to wake to total capture	107
9.5	Displacement of the turbines from the initial positions	108
10.1	Illustration of vector displacement using TDA with $nn = 2$	114
10.2	Influence of the number of considered nearest neighbours	116
10.3	Performance comparison: our algorithms TDA, CMA-ES*, and OpenWind	119
10.4	Demonstration of the TDA infeasible area modelling capability	120

List of Tables

3.1	SORTING: runtime results for (1+1)-GP* variants.	25
3.2	SORTING: runtime results for single-objective variants	34
3.3	SORTING: runtime results for multi-objective variants	35
3.4	SORTING: maximum tree sizes encountered until an optimum is found	37
3.5	Meaning of line styles in Figures 3.3 and 3.4.	43
3.6	SORTING: maximum tree sizes and maximum population sizes encountered for SMO-GP	44
3.7	Meaning of line styles in Figure 3.6.	45
3.8	SORTING: our average case conjectures for the single-objective problems	48
3.9	SORTING: our average case conjectures for the multi-objective problems	49
4.1	Computational complexity results from [29, 62]. Question marks indicate combinations for which we do not know any bounds.	53
4.2	WORDER/WMAJORITY: maximum tree sizes encountered until an optimum is found	54
4.3	WORDER/WMAJORITY: maximum tree sizes and maximum population sizes encountered for SMO-GP	57
4.4	WORDER/WMAJORITY: summary of our average case conjectures for the single-objective problems	60
4.5	WORDER/WMAJORITY: summary of our average case conjectures for the multi-objective problems	60
8.1	Wake modelling: symbol definitions	96
9.1	Wind Scenario 1 and Scenario 2	104
9.2	CMA-ES and experiment parameters	105
9.3	Metrics used to evaluate multiple layouts.	105
10.1	All results: our algorithms TDA, CMA-ES*, and OpenWind	118

List of Algorithms

2.1	WORDER(X)	12
2.2	WMAJORITY(X)	13
2.3	Derivation of $F(X)$ for SORTING	14
2.4	HVL-Prime mutation operator	16
2.5	(1+1)-GP*-single for maximisation	16
2.6	(1+1)-GP-single for maximisation	17
2.7	SMO-GP	18
5.1	Measure approximation quality of a population	68
6.1	Simple $(\mu + \lambda)$ -Approximation-Guided EA	72
6.2	Insert point into archive	72
6.3	Fast $(\mu + \lambda)$ -Approximation-Guided EA	73
7.1	Outline of Approximation-Guided EA II	81
7.2	Function <i>floor</i>	81
7.3	Function <i>increment</i>	82
8.1	Procedure for evaluation of wake effects due to the Park model [51]	99
9.1	Covariance Matrix Adaptation Based Evolutionary Strategy [43]	102
10.1	Turbine Distribution Algorithm	114

Statement of Originality

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library catalogue and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signed

Date

TO MY LOVING PARENTS, AND MY BELOVED FAMILY.

Acknowledgments

It was a great honour to work in such lively, inspiring, and outstanding places such as the Algorithms and Complexity Group at the Max Planck Institute for Informatics, and the School of Computer Science at the University of Adelaide.

First of all, I would like to thank Frank Neumann. He was the one to draw my attention to the Max Planck Institute for Informatics. I am most grateful to him for supporting my research on evolutionary algorithms within his research groups, and for inviting me to join him in South Australia. I also would like to thank Benjamin Doerr and Zbyszek Michalewicz for their support of my studies and for asking many thought-provoking questions. Furthermore, I am in debt to the heads of both institutions Kurt Mehlhorn and David Suter for allowing me to be part of their teams.

The best experiences during my research arouse from collaborations with others. Therefore, I am grateful to my co-authors Thomas Ackling, Bernhard Beckert, Bernd Bischl, Cody Boisclair, Thorsten Bormer, Jakob Bossek, Karl Bringmann, Gerd Beuster, Jareth Day, Christopher Denison, Carola Doerr, Tobias Friedrich, Sophia Gao, Niklas Henrich, Daniel Johannsen, Dora Jordan, Timo Kötzing, Per Kristian Lehre, Trent Kroeger, Olaf Mersmann, Samadhi Nallaperuma, Anh Quang Nguyen, Una-May O'Reilly, Tomasz Oliwa, Claudia Schon, Dirk Sudholt, Raymond Tran, Heike Trautmann, Tommaso Urli, Kalyan Veeramachaneni, Katya Vladislavleva, Junhua Wu, and Joseph Yuen. There are many more who I would like to thank, and amongst them are my fellow students in the labs, the professional staff at the institutions, and many conference attendees. Many invaluable friendships were formed.

My work was supported by several scholarships and grants, which I gratefully acknowledge. The Max Planck Research School Postgraduate Scholarship and the School of Computer Science Postgraduate Scholarship allowed me to focus on my studies by covering my living expenses and the numerous conference travels. Further generous travel support includes a Google PhD Top-Up Grant, a Google PhD Travel Prize, a Bupa Postgraduate Travel Grant, and the direct support from the conference organisers of the International Joint Conference on Artificial Intelligence (IJCAI) 2011 and of the Genetic and Evolutionary Computation Conference (GECCO) 2013.

I would also like to thank the University of Adelaide for selecting me as their representative at the China Nine / Group of Eight HDR Forum on “Clean Energy and Sustainable Future 2011” in Beijing, China (related to Part III). The attendance of that event has significantly broadened my personal and professional horizons.

Lastly, I would also like to express my gratitude to several anonymous reviewers and the members of my examination committee. Their comments significantly helped to improve the presentation of the results presented here.