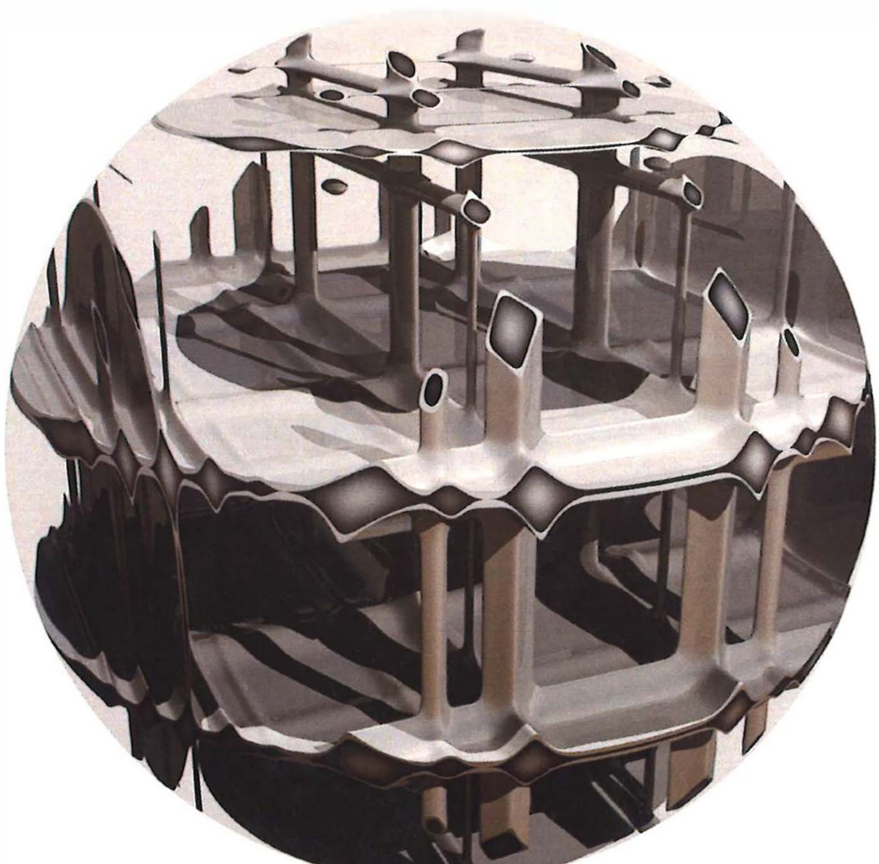
JYVÄSKYLÄ STUDIES IN COMPUTING
98

Ville Tirronen

Global Optimization using Memetic Differential Evolution with Applications to Low Level Machine Vision



JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 98

Ville Tirronen

Global Optimization using Memetic Differential Evolution with Applications to Low Level Machine Vision

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksessa (Ag Aud. 3)
joulukuun 13. päivänä 2008 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in the building Agora, Ag Aud. 3, on December 13, 2008 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2008

Global Optimization using Memetic Differential Evolution with Applications to Low Level Machine Vision

JYVÄSKYLÄ STUDIES IN COMPUTING 98

Ville Tirronen

Global Optimization using Memetic Differential Evolution with Applications to Low Level Machine Vision

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2008

Editors
Timo Männikkö
Department of Mathematical Information Technology, University of Jyväskylä
Pekka Olsbo, Marja-Leena Tynkkynen
Publishing Unit, University Library of Jyväskylä

Cover image by Ville Tirronen: An isosurface of the Michalewicz function

URN:ISBN:978-951-39-8102-0
ISBN 978-951-39-8102-0 (PDF)
ISSN 1456-5390

ISBN 978-951-39-3423-1
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2008

ABSTRACT

Tirronen, Ville
Global Optimization using Memetic Differential Evolution with Applications to Low Level Machine Vision
Jyväskylä: University of Jyväskylä, 2008, 98 p.(+included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 98)
ISBN 978-951-39-3423-1
Finnish summary
Diss.

In recent years meta-heuristic optimization has gained popularity in industry as well as academia due to increased computational resources and advances in the algorithms employed. As it has become apparent that no single algorithm can be declared the best in all cases, a rise in hybrid, or memetic algorithms that can be designed on case by case basis can be observed. In this work memetic algorithms are studied in depth in the context of Differential Evolution Algorithm that is among the best modern generic optimization algorithms. A whole setting of Memetic Differential Evolution methods is discovered and empirically validated with the inclusion of fitness diversity based co-ordination and stochastic adaptation schemes.

This work also studies the industrial problem of real-time paper web defect detection. This is a problem that is extremely constrained in time and as such permits no complex runtime solution. Thus, the problem has been formulated as an optimization problem utilizing a simple and efficient run-time model that is tuned to precision with rather more complex methods before applying it in the industrial field. The tools used for this are meta-heuristic optimization and especially, memetic optimization.

Keywords: memetic computation, evolutionary algorithms, image processing, machine vision, gabor function , direct search, local search, paper, paper mill, web defect, quality control, defect detection

Author Ville Tirronen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Supervisors Tommi Kärkkäinen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Tuomo Rossi
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Kirsi Majava
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Ferrante Neri
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Reviewers Han le Poutre
Head of research group at CWI, Amsterdam, the Netherlands, Full professor at Eindhoven University of Technology, the Netherlands

Yew Soon Ong
Nanyang Technological University

Opponents Han le Poutre
Head of research group at CWI, Amsterdam, the Netherlands, Full professor at Eindhoven University of Technology, the Netherlands

Joni-Kristian Kämäräinen
Lappeenranta University of Technology

ACKNOWLEDGEMENTS

This work has been done during years 2005-2008 at the Department of Mathematical Information Technology at the University of Jyväskylä. I would like to acknowledge the hard work and loyalty of my supervisors and co-authors during this time period, as well as financial support and co-operation from Viconsys Oy and the Centre of Excellence Programme in Paper technology coordinated by Jyväskylä Innovation Oy.

LIST OF FIGURES

FIGURE 1	Illustration of various steps in NMa	22
FIGURE 2	Differential evolution crossover in 2D	26
FIGURE 3	Generic Memetic DE schematic. The circular path represents the steps taken by each full iteration of the algorithm and different styles of line describe different components. The dotted line represents conditionality – some steps may be skipped according to internal logic of the method.	37
FIGURE 4	MDE schematic, showing the order of DE, HJa and SLS algorithms and control scheme used in the coordination	38
FIGURE 5	Probability of activating local searcher vs. fitness diversity index in MDE	39
FIGURE 6	Probability of activating local searcher vs. fitness diversity index in EMDE	40
FIGURE 7	EMDE schematic. This schematic details the probabilistic control scheme applied and definition of the probabilities employed (P).	40
FIGURE 8	Probability of activating local searcher vs. fitness diversity index in SFMDE	41
FIGURE 9	SFMDE schematic. This schematic details the initial refinement of the population and the probabilistic control scheme applied along with the definition for function for calculating them (p).	42
FIGURE 10	Representation of image number 2000000000 of Flickr	44
FIGURE 11	A General outline of machine vision process	46
FIGURE 12	Defect detection process	47
FIGURE 13	Various defects encountered in paper production	48
FIGURE 14	Parameters of equation (46) explained	52
FIGURE 15	Filter design process	55
FIGURE 16	Examples of training image pair in defect detection study	55
FIGURE 17	Behaviour of MDE	64
FIGURE 18	Behaviour of EMDE	66
FIGURE 19	Demonstration of search behaviour of EMDE. X-axis shows the number of fitness evaluations and which element of the algorithm was employed at each step.	67
FIGURE 20	Performance of SFMDE over the set of test problems	69
FIGURE 20	(continued)	70
FIGURE 21	Performance of SFMDE against modern meta-heuristics in the vision problem	71
FIGURE 22	Algorithmic performance	71
FIGURE 23	Best results on a training set image	72
FIGURE 24	Best results on a validation image	72
FIGURE 25	Behaviour of *MDEs on various test functions	74

FIGURE 25 (continued) 75
FIGURE 26 Behaviour of various adaptation schemes with FAMA algorithm 77
FIGURE 27 Activation functions for MDE 78
FIGURE 28 Activation functions for FAMA 78
FIGURE 29 Behaviour of FDSADE on 30 dimensional test functions . . . 80
FIGURE 30 Behaviour of FDSADE on 100 dimensional test functions . . . 81
FIGURE 31 Comparison with $\xi-$, $\psi-$ and $\nu-$ FAMA 83
FIGURE 31 (continued) 84

LIST OF TABLES

TABLE 1 Test cases used in this work 59
TABLE 1 (continued) 60
TABLE 1 (continued) 61

CONTENTS

ABSTRACT
ACKNOWLEDGEMENTS
LIST OF FIGURES AND TABLES
CONTENTS
LIST OF INCLUDED ARTICLES

1 INTRODUCTION 13
1.1 Motivation 14
1.2 Objectives 14
1.3 Contribution of this thesis 15
1.4 Authors contribution 15
1.5 Structure 16

2 GLOBAL OPTIMIZATION AND META-HEURISTICS 17
2.1 Introduction 17
2.2 Notation 19
2.3 Methods for Local Search in Continuous spaces 19
2.4 Basic Meta-heuristics in continuous spaces 22
2.4.1 Evolutionary Programming (EP) 22
2.4.2 Evolution Strategies (ES) 23
2.4.3 Real Encoded Genetic algorithm (rGA) 23
2.4.4 Simulated Annealing (SA) 24
2.4.5 Particle Swarm Optimization (PSO) 24
2.4.6 Differential evolution (DE) 25
2.5 Some examples of modern meta-heuristics 28
2.5.1 Self-Adaptive Control Parameters Differential Evolution (SACPDE) 28
2.5.2 Differential Evolution with Local Neighborhood (DEGL) .. 29
2.5.3 Self-Adaptive Differential Evolution with Neighbourhood Search (SaNSDE) 29
2.5.4 Differential Evolution with Adaptive hill Climbing Simplex Crossover (DEahcSPX) 30
2.5.5 Comprehensive Learning Particle Swarm Optimization (CLPSO) 31

3 ON MEMETIC ALGORITHMS 33
3.1 Memetic optimization 33
3.1.1 On classification of memetic approaches 33
3.1.2 Fitness Diversity 35
3.2 Novel Memetic Computing Schemes 36
3.2.1 Memetic Differential Evolution (MDE) 37
3.2.2 Enhanced Memetic Differential Evolution (EMDE) 38

3.2.3 Super-Fit Memetic Differential Evolution (SFMDE) 41

4 APPLICATION TO LOW LEVEL MACHINE VISION 43
4.1 Introduction 43
4.2 Machine Vision in Paper Industry 46
4.2.1 Defect detection in paper manufacturing process 47
4.3 Low-level Vision 49
4.3.1 Convolution 50
4.3.2 Gabor filters 50
4.3.3 Thresholding 51
4.3.4 Measurement of low-level vision system efficiency 53
4.4 Optimization based approach to weak defect detection 54

5 RESEARCH RESULTS AND INCLUDED ARTICLE SUMMARY 58
5.1 Research Framework 58
5.2 List of included articles 62
5.3 Summary of results 63
5.3.1 A Memetic Differential Evolution in Filter Design for Defect Detection in Paper Production 63
5.3.2 An Enhanced Memetic Differential Evolution in Filter Design for Defect Detection in Paper Production 63
5.3.3 Super-fit Control Adaptation in Memetic Differential Evolution Frameworks 65
5.3.4 Memetic Differential Evolution Frameworks in Filter Design for Defect Detection in Paper Production 68
5.3.5 On Memetic Differential evolution Frameworks: A Study of Advantages and Limitations in Hybridization 73
5.3.6 Fitness Diversity Adaptation in Multimeme Algorithms: A Comparative Study 73
5.3.7 The "Natura Non Facit Saltus" Principle in Memetic Computing 76
5.3.8 Integrating Fitness Diversity Self-Adaptation in Differential Evolution 76
5.3.9 A Fast Randomized Memetic Algorithm in Highly Multimodal Problems 79

6 CONCLUSIONS 85

YHTEENVETO (FINNISH SUMMARY) 86

ERRATA 87

REFERENCES
INCLUDED ARTICLES

LIST OF INCLUDED ARTICLES

PI V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava, T. Rossi. A Memetic Differential Evolution in Filter Design for Defect Detection in Paper Production. *Applications of Evolutionary Computing, Lectures Notes in Computer Science, Volume 4448, pp. 320–329*, 2007.

PII V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava, T. Rossi. An Enhanced Memetic Differential Evolution in Filter Design for Defect Detection in Paper Production. *Evolutionary Computation Journal, MIT Press, Volume 16, no 4*, 2008.

PIII A. Caponio, F. Neri, V. Tirronen. Super-fit Control Adaptation in Memetic Differential Evolution Frameworks. *Soft Computing-A Fusion of Foundations, Methodologies and Applications, to appear*, 2008.

PIV F. Neri, V. Tirronen. Memetic Differential Evolution Frameworks in Filter Design for Defect Detection in Paper Production. *Evolutionary Computer Vision; Studies in Computational Intelligence, to appear*, 2008.

PV F. Neri, V. Tirronen, T. Kärkkäinen, T. Rossi. Fitness Diversity Based Adaptation in Multimeme Algorithms: A Comparative Study. *Proceedings of the IEEE Congress on Evolutionary Computation, pp. 2374–2381*, 2007.

PVI F. Neri, V. Tirronen. On Memetic Differential Evolution Frameworks: A Study of Advantages and Limitations in Hybridization. *Proceedings of the IEEE World Congress on Computational Intelligence, pp. 2135–2142*, 2008.

PVII V. Tirronen, F. Neri, K. Majava, T. Kärkkäinen. The "Natura Non Facit Saltus" Principle in Memetic Computing. *Proceedings of the IEEE World Congress on Computational Intelligence, pp. 3882–3889*, 2008.

PVIII V. Tirronen, F. Neri. Differential Evolution with Fitness Diversity Self-Adaptation. *Nature-Inspired Algorithms for Optimisation, Studies in Computational Intelligence, to appear*, 2009.

PIX V. Tirronen, F. Neri. A Fast Randomized Memetic Algorithm in Highly Multi-modal Problems. *Proceedings of EuroGEN 2007*, 2007.

1 INTRODUCTION

Global Optimization is an ongoing challenge in the field of computational intelligence. As a wide field of study, it has applications in almost all areas of industry, ranging from simple control problems to finding best possible routes for spacecraft or design of most efficient and compact machinery possible.

In their simplest form, optimization problems are formed by a fitness function which maps a candidate solution to a performance score, such as an electrical motors' control system design into a number describing its reaction time under change. The parameters may be either discrete or continuous and in classical cases bounded by a hyper rectangular volume. Such problems have been solvable for long time. However, many real-world processes and even corresponding mathematical models yield systems for which fitness function is not unimodal or even a function but a noisy relationship between parameter values and possibly several goal values. Bounds are no longer nicely rectangular or very well definable.

In the papers included in this thesis, various evolutionary methods are designed in accordance with memetic computing philosophy and verified by application to real world problems such as the low-level machine vision problem and the electrical engine control problem and to several commonly known, but artificial test cases. Of the two real world applications discussed in the included papers, the machine vision problem is handled in more detail, as it is a contribution by the author. That problem arises from paper industry and quality control and has to do with automatic detection of defects in paper web directly in paper mills. Visual inspection is the only way of achieving 100% coverage at the moment, and this gives a rise to a very hard real-time problem, a problem that is due to a massive amount of image data gathered from the paper.

Hard real-time constraints lead to systems that are by necessity rather simple and efficient instead of being highly sophisticated and intelligent. This leads to our approach of using simple and generic vision model and tuning it for task specific optimal performance. This idea leads back to the field of optimization: if you can measure performance and adjust performance, an optimization algorithm can be employed to automate this process beyond human means.

1.1 Motivation

Optimization is a typical task for almost all industrial processes. Its sub-field of global optimization using meta-heuristics is becoming more and more commonplace due to the advent of efficient computational systems. One field that has risen in recent times is evolutionary computation or processes that perform optimization by imitating the theory of evolution. Such methods usually maintain a population of candidate solutions that mutate, produce offspring, and die according to the dynamics of the function that is being optimized.

However, these classical meta-heuristics are constantly being replaced by more elaborate and/or efficient schemes. One of these is differential evolution (DE) which, despite its relatively simple structure has proven its worth as a general optimization method. It is commonly acknowledged that no single algorithm will perform in all cases, providing a "market" for algorithms that can be composed by the algorithm designer for problem properties. This, in part, has worked for the success of memetic algorithms that in essence are a class of hybridizations between meta-heuristics and local search algorithms, allowing great flexibility for the algorithm designer to compose his algorithm.

The themes in which memetic algorithms are studied here consist of examining the difficult problem of co-ordination of algorithmic components which arises from hybridization and practical applications that benefit from these. In this field the study of algorithmic coordination is advanced, and several new algorithms are presented in order to investigate the space of successful meta-heuristic designs.

1.2 Objectives

The objective of this work is to introduce several new algorithms for the meta-heuristic optimization community. It is generally acknowledged that there is "no free lunch" [WMCJ97] in optimization, i.e., there is no single best algorithm for all tasks. New methods in the field provide samples of the landscape of optimization problems and algorithms. The algorithms in this work are expected to be superseded by others in future but they hopefully have something to contribute to future research of the subject.

The work is also bound to industrial machine vision processes, and the objective here is to both solve a single challenging problem and to point out a way of solving a large class of problems by means of optimization and machine learning. As per author's vision there exists a large class of machine vision problems that are both very task specific, non-generalizable and solvable by a rather simple engineering effort of combining filtering and thresholding. The conjecture pursued here is that this class of problems is essentially an optimization problem solvable with known methods better than by a human operator in time that can be spent.

1.3 Contribution of this thesis

The studies conducted display promising results in both sectors of optimization and its applications. The application area can be considered solved in the instance it was investigated on, but the remaining area of solving whole class of convolution and thresholding problems remains open. The work dealing with low-contrast wrinkle type defects has been met with success. As shown in numerical results in the included papers the introduced method of defect detection outperforms the classical methods applied to this problem ([PI],[PII],[PIII]).

In the field of optimization, progress has been made on adaptation schemes by the introduction of stochastic fitness-diversity based adaptation in [PII, PVII, PV] and [PIII] as well as by the increase in basic knowledge of proper hybridization of DE. Several new algorithms, memetic differential evolution (MDE) in [PI], and enhanced memetic differential evolution in [PII] using these concepts have been developed and tested in both real-world applications and in sets of test problems. These algorithms have been found competitive against classical methods as well as DE. Also, study comparing parameter setting vs. memetic frameworks has been conducted in [PVI] and the numerical results show that although DE is highly performing, it requires extensive parameter setting unlike the proposed memetic algorithms which can operate automatically without any costly tuning. As a concrete example, three new memetic algorithms have been introduced and tested. They are summarized in Section 3.2

The resulting findings show that DE is well suited as a basic framework for memetic purposes. Various basic local search methods have been evaluated for their suitability to inclusion within the memetic framework and their performances have been studied in different algorithmic scenarios.

1.4 Authors contribution

Publications presented in this thesis are the product of co-operation within a research group composed of several authors. Before that co-operation which lead to the publication of these articles the author's main topic of research was (and still remains) machine vision, leading to model formulations in machine vision problems presented in papers [PI]-[PIV]. The author's work in papers [PV]-[PIX] begun with implementation and experimentation and grew to full sharing of ideas. The numerical experiments in the included papers have been executed solely by the author. The main ideas originating from the author are the stochastic control scheme and its expansions as well as modelling of the machine vision problem.

The most notable relationship to this thesis is [Ner07] which is a direct precursor of this work. The concept of fitness diversity in memetic algorithms is studied in depth in that work and continued here. These two theses contain three papers ([PI, PV, PIX])in common. Authors make the case of including these works in both theses due to their different contributions. In the case of article [PI] the work is separated into two parts: machine vision and memetic computation. In context of this thesis, this paper is fundamental in order to introduce memetic differential evolution and also as an introduction to the vision problem which is a contribution by the author.

Article [PV] is included in this thesis for the reason that it summarizes the work on different fitness diversity measures in an external context to memetic differential evolution, thus providing validation for the schemes used herein. As the paper contains no new algorithms and by its nature makes comparisons through empirical research, the contribution of the author is harder to state. In this case, the author participated in elaborating the test set and actually implemented the test framework.

Lastly, in [PIX] the idea of a completely stochastic control is due to the intuition of the author, while the co-author dealt with the testing framework and helped to derive conclusions from the resulting algorithm.

1.5 Structure

The rest of this thesis is roughly divided into four parts.

Chapter 2 details the background of optimization describing classical meta-heuristics. Differential Evolution is given a special emphasis since it forms the basis of all the algorithms introduced later. Also, in order to support conclusions from a comparison to classical meta-heuristics, some modern optimization algorithms are presented in detail. The second part concludes with the topic of memetic algorithms and presents the methods designed by authors. The exact results obtained by these new methods as well as an analysis of their performance are given in Chapter 5 of this work.

Chapter 3 focuses on the concept of Memetic Algorithms and the related concepts of coordination and fitness diversity. That chapter also contains novel algorithms presented in the thesis. The chapter concludes the part about optimization.

Chapter 4 details machine vision processes in the scope applied here. The machine vision part starts with a general overview of the topic and then focuses on a low-level machine vision and especially on processes of convolution and thresholding utilized in the included articles. The last part of the chapter gives the author's formulation of a defect detection process as an optimization process.

The thesis concludes with Chapters 5 detailing the results obtained in the included articles as well as describing how the articles are interlinked to each other and conclusions drawn from them in Chapter 6

Although valiant efforts have been undertaken by several reviewers as well as by the authors, an errata of unfortunate mistakes in included articles is given after the conclusions for this thesis.

2 GLOBAL OPTIMIZATION AND META-HEURISTICS

2.1 Introduction

In order to clarify the goals let us introduce a general optimization problem in the following form:

$$\begin{aligned} &\text{Minimize} && f(x) \\ &\text{subject to} && x \in D \subset \mathbb{R}^n, \end{aligned} \tag{1}$$

where f is used to denote the real-valued fitness function and x the set of parameters to be optimized. D is commonly called a decision space or domain of the problem, describing from which set of values the solution may be composed of. Vectors drawn from this set are called candidate solutions. Independently, the elements of these vectors are called design-variables.

The goal is to find the *global* optimum x_{glo}, i.e., any such a point for which $f(x_{glo}) \leq f(x)$ for all $x \in D$. Also, to simplify matters further we assume that the domain of the problem D is a hyper-rectangle. In practical cases, as observed before, the problem may contain more complex constraints of acceptable values of x which are not considered here. Suitable ways of dealing with complex constraints are surveyed in [Mic95, Coe02, RY05].

A traditional way of solving optimization problems of the form of (1) is by gradient search, which remains an excellent way of solving differentiable and convex problems within convex domains. In such cases a properly formulated gradient-based method will yield the global minimum x_{glo} [NW00, GMW82], but these rather restrictive conditions often do not apply in real-world problems. Instead, they often are multi-modal and can only be solved via direct search methods. Due to increasing availability of computational resources it has become commonplace to study algorithms for problems of this nature.

In this work, the focus lies on direct search methods. The definition for direct search used in this thesis comes from [Wri96]: Direct search uses only function values and does not intend to develop an approximate gradient. Within the class of direct search methods another rough categorization is used. First category comprises of (direct) Local Search Algorithms (LSA) that are used to solve the simplified problem of finding a local minimum of f called x_{loc}. Here x_{loc} denotes any point for which there exist a small positive δ so that $f(x_{loc}) \leq f(x)$ for all x $||x - x_{loc}|| \leq \delta$. Local search can benefit from several other characteristics that may be attainable by limiting the search area. Such local patches of f can, in some cases, be considered to have properties such as linearity, convexity, or other attributes that f as a whole lacks but which are beneficial for the optimization process. Local search methods are usually 'fast' to converge to a local minimum but of course, this usually is not the global minimum sought for.

Local search methods include direct search methods such as Nelder-Mead algorithm (NMa) [NM65], Rosenbrock algorithm (RA) [Ros60], Hooke-Jeeves algorithm (HJa) [HJ61] and Stochastic local search (SLS) [HS05] (for the adaptation of SLS concept for real valued problems under inspection see included articles [PI, PII]). All these methods are based on making changes to a candidate vector in hopes of replacing it with a better one. These methods operate with a greedy philosophy, accepting any change that improves upon the original solution (with the exception of SLS). As per their nature, these methods find only local minimae.

The second category consists of so called global methods. In this context the main interest lies in global meta-heuristics. Meta-heuristics contain high level strategies for optimization that are often more inspired by some natural phenomenon, such as evolution rather than by any mathematical properties of the problem. As word meta-heuristics suggest, it is usual that many of these methods have no theoretical proof of convergence. Some proofs for convergence exist (See for example, [Loc00, OLZW06]), but they are often considered only as a minor detail from the practical viewpoint. Such methods are instead empirically evaluated and validated.

Sources of inspiration for such meta-strategies include evolution (yielding evolution strategies and genetic algorithms) or mundane source like dynamics of various swarming animals like flocking birds or ants searching for their sustenance (giving rise to particle swarm optimization and ant colony optimization).

As a general overview the term meta-heuristics can be seen to encompass method-families such as genetic algorithms (GA) [Gol89], evolution strategies (ES) [Rec73], particle swarm optimization (PSO) [EK95], ant colony optimization (ACO) [DMC96, DC99] and simulated annealing (SA) [KGV83]. Other less nature-inspired methods are differential evolution (DE) [SP95, SP97] and tabu search (TS) [Glo89, Glo90]. In many cases such methods operate on a population of possible solutions which is refined towards optima according to the heuristic rules of each algorithm, making the term meta-heuristic often synonymous to "population based". In this work special focus is laid on evolutionary methods, or meta-heuristics originating from the idea of theory of natural evolution, since they lay also foundation for another class of meta-heuristics, namely memetic algorithms (MAs) that form the core of this work.

The rest of the chapter will give descriptions of various direct search methods starting from simple methods applicable for local search and ending up with state-of-the-art methods published in recent years. Basic meta-heuristics such as ES and SA are detailed in the form that they are used in this work disregarding the hundreds of variants in existence. After these classical methods DE is studied at depth. The chapter is concluded with the author's work on memetic variants of DE.

2.2 Notation

This section gives the basic notation for this chapter. To start with, $f(x)$ denotes the value of the fitness function at point x and $P^i = \{p_1^i, p_2^i \dots p_N^i\}$ is the set of active solution candidates at iteration i. In case of single solution methods, such as local search by Hooke-Jeeves method, P^i contains only one candidate $P^i = \{p_1^i\}$. In some cases various trials that the algorithms make are explicitly denoted as an ordered sequence $T = t_1, t_2, \dots, t_n$.

A common notation for several basic operations is defined as follows. Notation g_σ is used to denote a Gaussian distributed random variable and u an uniformly distributed one in the range $[0, 1]$. Notation u_n is used to describe situations where the generated random number is used more than once in the same equation. Random vectors with uniform (in range $[0, 1]$) and Gaussian distributions are denoted with $\hat{u}$ and $\hat{g}_\sigma$, dimension of vectors is equal to that of the optimization problem. In this case σ is used as a shorthand to denote the standard deviation of the distribution related to the size of the fitness domain. The standard deviations for each dimension are defined as σ times the width of that dimension. Notation $\text{rank}_{P_i}^f(p_j)$ denotes the rank of an individual p_j in generation P_i according to fitness f. Also $\arg\min_{x \in P}^N f(x)$ is used to denote the selection of N smallest elements according to f from the set P. Pointwise multiplication of vectors is denoted by $a : b =< a_1b_1, a_2b_2, \dots a_Nb_N >$, and when necessary, the k^{th} element of vector a is noted as $a[k]$.

2.3 Methods for Local Search in Continuous spaces

In this section the local search methods are described. They are detailed in their basic version discarding the fact that as classical methods they have remarkable number of variants and improvements. The versions here are the ones used in the algorithms described in the included papers. For further details see the included papers.

Hooke-Jeeves Algorithm (HJa) is a simple local search method. It belongs to the class of pattern search methods with examine the fitness landscape around a base point according to a pattern. If this pattern search yields a better point, the algorithm continues in that direction until failure. After this the pattern search is restarted at the new point. In case of failure in the pattern search the step size is decreased.

This kind of algorithm can be described in two parts: the pattern move and the exploratory move. A basic pattern move can be read as follows. Let $A = \left\{a_1, a_2 \ldots a_{N_p}\right\}$ be a set of vectors forming a pattern for a pattern move and $t_0 = p_1^0$ a randomly initialized basepoint where the search begins. The sequence of trials done by the algorithm is denoted by $\{t_n\}$. A trial t_n is successful if $f(t_n) < f(t_k)$ for all $k < n$. In that case the algorithm takes this step.

The sequence of trials taken reads as:

$$t_{n+1} = \begin{cases} t_n + \frac{h}{2^{c_n}}(t_{n-1} - t_n), & \text{if } t_n \text{ is a successful trial ,} \\ t_s + \frac{h}{2^{c_n}} a_{((n-s) \mod N_p+1)}, & \text{otherwise,} \end{cases} \tag{2}$$

where t_s is the last successful trial, $\left((n-s) \mod N_p\right)$ is the number of failed trials since the last successful trial and, c_n denotes the number of times that the entire pattern has failed to generate a successful trial. In other words, how many times the entire pattern has failed to produce any successful solution. The step-size h is a parameter for this algorithm.

The actual path taken by the algorithm is formed out of the successful trials, i.e., the sequence of steps $\{p_1^i\}$ is thus a subsequence of $\{t_n\}$.

In case of HJa the pattern is $A = \{e_1, -e_1, e_2, -e_2, \ldots, e_n, -e_n\}$ where e_j is the j^{th} basis vector. HJa applies the pattern in a greedy manner, selecting first a pattern move that leads to a decrease in fitness.

The exploratory move takes advantage of the pattern move by generating new search points along the line pointed by a successful pattern step. In case of HJa the steps have a fixed length h. The search returns to the pattern phase when the exploratory moves stop working and halves the h when the pattern moves fails to work.

Rosenbrock Algorithm (Ra) is also a pattern search method. Unlike HJa, there is no exploratory move and the pattern is updated at every trial. When a point in the pattern is successfully tested the respective offset is multiplied by a scalar a and respectively decreased by b when the trial is unsuccessful. When every move in the pattern has been applied to produce both a successful move and a failed trial, the pattern is rotated to the direction from the starting point of the iteration to the final point and the search begins anew.

Stochastic Local Search (SLS) can also be seen as a pattern search method consisting of only the pattern moves. However, the pattern in SLS is randomly sampled from Gaussian distribution centered around the basepoint and is exhaustively searched for the best point. Formally, the sequence of steps taken with SLS is

$$p_1^{i+1} = \underset{x \in \left\{p_1^i + g_\sigma\right\}^k \cup \left\{p_1^i\right\}}{\arg\min} f(x), \tag{3}$$

where a set of k randomly generated elements is denoted as $\{\ldots\}^k$. The parameter k controls the explorativeness of the algorithm along with σ.

Nelder-Mead Algorithm (NMa) is different from the pattern search methods. NMa is also known as a simplex method for its operating principle. NMa evolves a simplex, or in other words the simplest complex hull of $n+1$ vertices with a non-zero volume, where n equals the dimension of the problem. The main working principle is to examine the vertices in the simplex and replace the worst of them with a better point estimated from the others. Other rules exist in case that this process fails. As this method has a 'population' of points or vertices that get updated, the simplex at iteration i is denoted as $P^i = \{p_1^i, p_2^i \dots p_{n+1=N}^i\}$. This set is assumed to be sorted according to fitness so that p_1^i is always the point with the best fitness and p_{n+1}^i is the worst. For shorthand, let $P_h^i = \{p_1^i, p_2^i \dots p_n^i\}$ denote the set of vertices excluding the worst one. Then, steps taken by NMa can be characterized with the following iteration formula:

$$P^{i+1} = \begin{cases} P_h^i \cup \{x_c + \chi(x_r - x_c)\}, & \text{if } f(x_e) < f(x_r) < f(p_1^i) \\ & \textit{(Expansion)}, \\ P_h^i \cup \{x_r\}, & \text{if } f(x_r) < f(x_e) \text{ and } f(x_r) < f(p_1^i) \\ & \textit{(Reflection)}, \\ P_h^i \cup \{x_{oc}\}, & \text{if } f(p_n^i) \leq f(x_r) \leq f(p_{n+1}^i) \text{ and } f(x_{oc}) < f(p_{n+1}^i) \\ & \textit{(Outside Contraction)}, \\ P_h^i \cup \{x_{ic}\}, & \text{if } f(p_{n+1}^i) \leq f(x_r) \text{ and } f(x_{ic}) < f(p_{n+1}^i) \\ & \textit{(Inside Contraction)}, \\ \{s_1, s_2 \dots s_{n+1}\}, & \text{otherwise} \\ & \textit{(Shrink)}, \end{cases} \tag{4}$$

where

$$x_c = \frac{1}{n}\sum_{j=1}^{n+1} p_j^i, \tag{5}$$

$$x_r = x_c + \chi(x_c^i - p_{n+1}^i), \tag{6}$$

$$x_e = x_c + \chi(x_r - x_c), \tag{7}$$

$$x_{ic} = x_c + \gamma(x_r - x_c), \tag{8}$$

$$x_{oc} = x_c + \gamma(x_c - p_{n+1}^i), \tag{9}$$

$$s_j = p_1^i + \sigma(p_j^i - p_{n+1}^i), \quad j = 1, 2, \dots, n+1. \tag{10}$$

The centroid of the simplex is x_c and x_r is a reflection of the worst point to the other side of the simplex. The extension of the reflection is called x_e and it can be considered to perform an exploitative step. These possible steps are illustrated in Figure 1. In case of total failure to find a better point, the simplex is replaced with a shrunk version consisting of points $\{s_i\}$. General parameter setting used for this method is $\rho = 1$, $\chi = 2$, $\sigma = \gamma = \frac{1}{2}$.

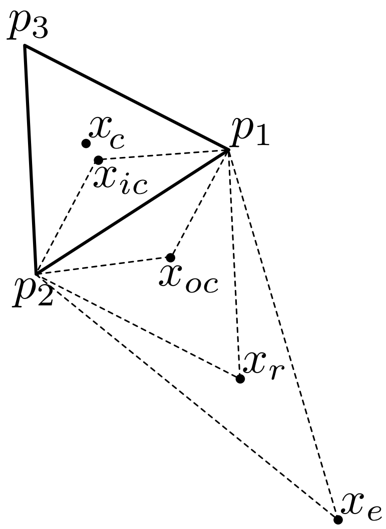


FIGURE 1 Illustration of various steps in NMa

2.4 Basic Meta-heuristics in continuous spaces

In this section the basic meta-heuristics are described. These are the methods that are used in the included papers as comparisons for DE based frameworks.

There are numerous variants of all classical meta-heuristics as well as choices taken in their use. In the following only one particular version is described and that is the one used for comparison purposes in the included papers. For some important variants a citation may be provided. The notation intentionally differs from that given in the articles in order to provide a compact representation and a different point of view for the reader. This difference arises also from the author's use of the functional programming paradigm in which the representation such as this is more commonplace and more easily transformed into an executable code than an usual pseudocode, such as is found in the included articles.

2.4.1 Evolutionary Programming (EP)

Evolutionary Programming (EP) is among the first meta-heuristics drawing inspiration from evolution [ES03] and is a precursor for a more advanced ES. In EP all individuals are organized in generations P^i containing candidate solution vectors and a vector of self-adaptive parameters S^i having the same dimension as the domain of fitness.

During the generation cycle, each individual generates an offspring. For the generic j^{th} design variable (or parameter), the update is performed in the following way:

$$S^{i+1} = S^i : \left(\hat{1} + \alpha \hat{g}_\sigma\right), \tag{11}$$
$$o^i_j = p^i_j + S^{i+1}\hat{g}_\sigma, \quad j = 1, 2, \ldots, N, \tag{12}$$
$$\tag{13}$$

where α is a constant value traditionally set equal to 0.2 and o^i_j represents the candidate off-spring. When all the offspring solutions are generated, a population composed of parents and an offspring ($P^i \cup \left\{o^i_1, o^i_2 \ldots o^i_N\right\}$) undergoes a survivor selection. The fitness value of each individual in this combined population is compared with the fitness values of a set of other randomly selected solutions. Solutions with the highest number of best fitness scores in the comparisons are selected for the subsequent generation.

2.4.2 Evolution Strategies (ES)

Evolution Strategies are developed with inspiration from natural evolution over generations. Semantically ES functions by creating an offspring population by crossing over existing individuals and applying mutations to them. Then the resulting set of parents and offspring are culled to provide the next generation of individuals of the same size as the previous one. (The version used here is commonly called $(\mu/\rho+,\lambda)$-ES in literature [BHS91].)

Let P^i denote a sequence of populations of vectors $\left\{p^i_j\right\}$. Population P^i is defined in the following equations, by using the offspring population O^i as the starting point. In the following, o^i_j is used to denote the elements of O^i:

$$o^i_j = \hat{u}_0 : p^i_a + \left(\hat{1} - \hat{u}_0\right) : p^i_b + \hat{g_\sigma}, \quad j = 1, 2, \ldots, N, \tag{14}$$

where p^i_a and p^i_b are randomly selected members from P^i. Then

$$P^{i+1} = \underset{x \in P^i \cup O^i}{\arg\overset{N}{\min}}\, f(x). \tag{15}$$

The solution after i generations is noted as $s_i = \arg\min_{x \in P_i} f(x)$. The sequence of $\{s_i\}$ is usually converging towards an optimum. It is common to improve performance of ES by introduction of 1/5-rule which states that σ is increased by a ratio $c \in [0.817, 1]$ (determined empirically in [Sch77]) when $1/5$ of the offspring benefit from the addition of $\hat{g}_\sigma$ and decrease by the same ratio otherwise [ES03, Rec73]. This is usually done in every n generations.

2.4.3 Real Encoded Genetic algorithm (rGA)

The variant of rGA used in this work is in essence the same algorithm as ES without the $1/5$ rule and the added probability of selecting the parents for the next generation [ES03]. Let P^i denote the sequence of generations as a sequence of sets of individual trial vectors. First we define the probability for selecting an individual p_j^i for mating which is given by the following equation:

$$\operatorname{prob}(p_j^i) = \frac{2-s}{\#P^i} + \frac{2\,\operatorname{rank}_{P^i}^{f}(p_j^i)(s-1)}{\#P^i\,(\#P^i-1)}, \tag{16}$$

where s is the selection pressure that is a parameter for this algorithm. The offspring corresponding to population P^i are defined as:

$$p_j^{i+1} = u_0 p_a^i + (1-u_0)p_b^i + \hat{g}_\sigma, \quad j = 1, 2, \ldots, N, \tag{17}$$

where p_a^i and p_b^i are selected from population P^i with probabilities assigned to them according to equation (16).

2.4.4 Simulated Annealing (SA)

Simulated annealing is a meta-heuristic that is inspired from annealing in metallurgy. When metal is heated, atoms in the material are displaced from local positions of minimum energy and can shift around in states of higher energy. A slow cooling process allows them more time to get fixed to locations with less internal energy. Simulated annealing works by randomly permutating an initial solution and with decreasing probability accepting solutions worse than the previous ones (i.e., SA works with 'population' of a single point.) Formally SA is the following:

Let the set of steps taken by SA be denoted as:

$$t_{i+1} = p_1^i + \hat{g}_\sigma, \tag{18}$$

$$p_1^{i+1} = \begin{cases} t_{i+1}, & \text{if } f(t_{i+1}) \leq f(t_s) \text{ or} \\ & \exp\left(\frac{f(p_1^i)-f(t_{i+1})}{T}\right) \geq u, \\ p_2^i, & \text{otherwise.} \end{cases} \tag{19}$$

Then the sequence of points $\{p_1^i\}$ describes the path taken by Simulated Annealing leading towards the optimal solution. Temperature variable T guides the annealing. A common scheme is to have T decrease according to $T = \frac{1}{c}$ where c is the number of accepted solutions.

2.4.5 Particle Swarm Optimization (PSO)

Particle swarm optimization is inspired from the work of Reynolds in [Rey87] defining a model for swarm behavior. At a general level, individuals in PSO populations are aware of the movement and position of other individuals and adjust their own movement to match other individuals, creating a swarm-like behavior. One essential difference to other population based meta-heuristics is also the introduction of velocity and inertia: individuals prefer to keep their direction and thus make the search very directional.

PSO can be defined as a sequence of populations like ES and RGA. The velocities associated with members of P^i are noted as v_j^i:s. Variables c_1 and c_2 are parameters determining how strongly the particles are attracted to good solutions found thus far. Then P^{i+1} is defined as follows:

$$v_j^{i+1} = \max\left(M_v, v_j^i + c_1 u(\mathrm{pb}_j^i - p_j^i) + c_2 u(\mathrm{gb}^i - p_j^i)\right), \tag{20}$$
$$p_j^{i+1} = p_j^i + v_j^{i+1}, \tag{21}$$

where pb_j^i denotes the location of minimum fitness achieved by this particle:

$$\mathrm{pb}_j^i = \operatorname*{arg\,min}_{x \in \left\{p_j^n \text{ for all } n \leq i\right\}} f(x), \tag{22}$$

and gb^i denotes the best solution found thus far

$$\mathrm{gb}^i = \operatorname*{arg\,min}_{x \in \bigcup_{n \leq i} P^n} f(x). \tag{23}$$

Maximum velocity M_v is a parameter for this method. The solution found by PSO after i iterations is

$$s_i = \operatorname*{arg\,min}_{x \in \bigcup_{n \leq i} P^n} f(x).$$

2.4.6 Differential evolution (DE)

As the basis of algorithms introduced in this thesis, Differential Evolution (DE) is presented in greater depth than the other algorithms. DE [SP97] is a modern real valued optimization algorithm with characteristic versatility and reliability. DE can be characterized as a population based evolutionary method with a steady state logic. The basic form of DE is usually denoted as *DE/rand/1/bin* [SP97]. This variant is referred to as DE throughout the thesis. The parameters for this method are exploration rate F, crossover rate Cr and population size S_{pop}.

DE is a steady state algorithm updating population of candidate vectors $P^i = \left\{p_1^i, p_2^i, p_3^i \ldots p_n^i\right\}$ according to equations (24)–(26). In those equations the intermediate steps in generation of population P^{i+1} are denoted by o_j' and o_j. We have

$$o_j' = p_a^i + F\left(p_b^i - p_c^i\right), \tag{24}$$

where p_a^i, p_b^i, and p_c^i are randomly selected members from population P^i. Thus $\left(p_b^i - p_c^i\right)$ is the vector difference of two randomly selected points, giving the direction in which the offspring will be generated. Finally,

$$o_j = \mathrm{Crossover}_{Cr}(p_j^i, o_j'), \tag{25}$$

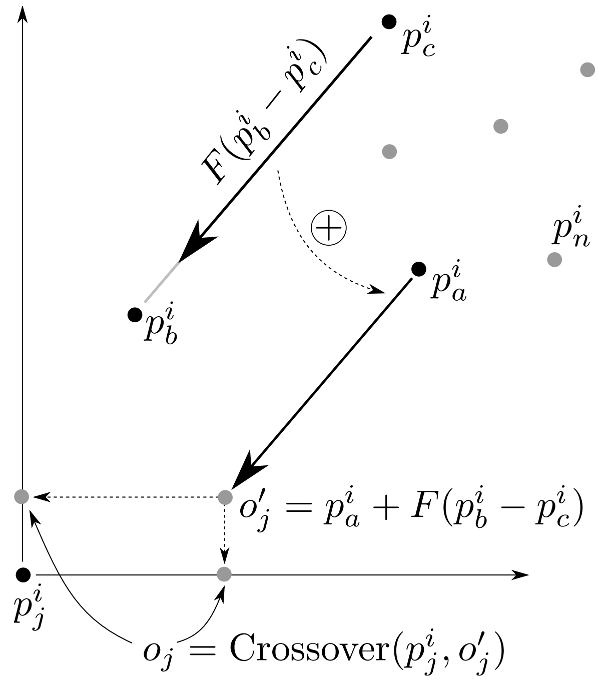


FIGURE 2 Differential evolution crossover in 2D

$$p_j^{i+1} = \begin{cases} o_j, & \text{when } f\left(o_j\right) < f\left(p_j^i\right), \\ p_j^i, & \text{otherwise.} \end{cases} \tag{26}$$

Here crossover is an operator, switching elements of two vectors with a uniform probability:

$$\mathrm{Crossover}_{Cr}(a, b)_k = \begin{cases} b_k, & \text{with probability } Cr, \\ a_k, & \text{otherwise.} \end{cases} \tag{27}$$

Usually though, this is complemented by a condition that at least one element is switched between the members.

The main novelty of DE is in its way of generating candidate offspring by sampling direction vectors contained in the population depicted as in Figure 2 and formalized in equation (24). This allows the algorithm to employ the shape of the population to guide the search over the fitness landscape by exploiting promising exploration directions [PSL05].

The directional search is augmented by a uniform crossover (equation (27)) which swaps each element in o'_j to its corresponding element in p_j^i with a probability of Cr.

DE has been applied to continuous function optimization, constrained optimization, and even to discrete problems (see survey in [PSL05]). It has been most successful for real valued function optimization and for many practical purposes has defeated in efficiency many 'classical' evolution algorithms (see [SP95, VT04] and work done in articles [PI]–[PIV] of this thesis). Having the virtues of simplicity and competitive performance, DE can be considered at least a peer to many modern optimization algorithms.

DE is generally considered to offer a remarkably stable performance in many cases, but it contains three parameters that control its performance. Like in other EAs population size presents a trade-off between convergence speed and reliability. Small sized populations can converge faster but are more likely stuck in local minimums than slowly moving large populations. Two other parameters control the exploration strategy, but their interpretation is not trivial. For example, F is just a scaling factor and the real step size is dependent on the shape of the population. Similarly, when adjusting Cr several things change at once. The coherence along cardinal axes increases and likewise the speed of advance along those directions increases. As per author's conjecture, this importance of parameter Cr increases with dimensionality of the problem.

According to tests conducted in article [PIV] and articles [LL02, GMK02] along with a more recent article detailing a real world problem [ZWLK06], the optimal parameter setting is indeed problem dependent. These works also present guidelines for proper settings, such as selecting $F \in [0.5, 1)$ and $Cr \in [0.8, 1)$. A rule for selecting a population size as 10 times the number of dimensions is given at [SP97]. The tests in that work also seem to hint that the optimal population sizes for DE in the given classical test problems might be much smaller.

To further discuss the peculiarities of DE, two definitions are necessary. Although the word convergence has been used before, it is used in this context to mean a state where an algorithm has stopped improving upon the best individual and the diversity of the solutions is approaching zero. Stagnation on the other hand is a similar condition with the exception that the population of the algorithm has retained its diversity but still lost its potential for further improvements.

Thus, returning to the topic of small population size, it should be noted that small populations pose a problem of stagnation [LZ00]. Hypothetically, stagnation is caused by the semi-deterministic functioning of DE. When the population size is limited, the number of possible directions to employ in the search becomes also limited along with the number of possible trial vectors. If none of these potential new solutions improve on the predetermined parent vector the optimization cannot continue. In practice, as seen in the many tests presented in this work and in the conclusion of article [LZ00] it seems that stagnation is a rare event affecting mostly small population sizes. [1]

The number of parameters makes a tuning process of DE a laboursome endeavor. The amount of possible parameter combinations is large and due to the stochastic nature of the process several simulations are needed for each combination. Also, due to the fact that a single problem usually needs to be solved only once, this causes a huge amount of extra computational effort that is eventually discarded. Also, there is no guarantee that the same parameter setting would

[1] In the included articles, stagnation is also used to describe situations where convergence becomes impractically slow.

be optimal during the entire optimization process. It is conceivable that properties of fitness landscape change when the population enters another area in the landscape.

The aforementioned difficulty of parameter setting has given rise to a body of work trying to modify DE to automatically adapt its parametrization. Such works were spearheaded by work in [LL02] which incorporated a fuzzy logic tuning of the parameters of DE to form an algorithm called Fuzzy Adaptive Differential Evolution (FADE). FADE is further studied in articles [LL03] and later in [LL05]. Other works in this field include such publications as [BGB$^+$06] which presents self adaptation for adjusting the parameters with similar logic that has been used with ES (see [ES03]). The introduced algorithm Self-Adapting Control Parameters in Differential Evolution (SACPDE) has also been further refined for constrained problems in [BZM06]. Methods such as these detail the background and field in which methods in this thesis ”compete”.

2.5 Some examples of modern meta-heuristics

This chapter details some examples of other modern meta-heuristics in order to give background for research done here. As the field is large, they are but examples of a vast selection of methods. The methods were selected for their relative efficiency, simplicity and their close ties to DE and PSO that were used in methods developed in this thesis. In order to implement these methods, the reader is advised to study the original papers where they were presented. Methods that are this intricate may suffer from being re-described from different points of view.

2.5.1 Self-Adaptive Control Parameters Differential Evolution (SACPDE)

SACPDE is DE which has an inscrutable adaptation logic with separate values for each of the control parameters F_j^i and Cr_j^i for each individual p_j^i in the population. These values are used to generate the offspring p_j^{i+1} matching the parent that the values are associated with. These parameters are re-randomized with a uniform probability for successful offspring $\left\{p_j^{i+1}\right\}$. The scale factor is set as:

$$F_j^{i+1} = \begin{cases} F_l + uF_u, & \text{with probability } \tau_1, \\ F_j^i, & \text{otherwise}, \end{cases} \tag{28}$$

where F_l and F_u denote the range of variability for the scale factor F. Moreover, the crossover rate for each individual is:

$$Cr_j^{i+1} = \begin{cases} u, & \text{with probability } \tau_2, \\ Cr_j^i, & \text{otherwise}. \end{cases} \tag{29}$$

The parameters τ_1 and τ_2 are set to 0.1 for both parameters, as suggested in [BGB$^+$06]. The bounds F_l and F_u are set to 0.1 and 0.9. This scheme allows for stochastic sampling of the parameters space in order to find such parameter settings that work in a given state of optimization.

2.5.2 Differential Evolution with Local Neighborhood (DEGL)

Local neighborhoods are commonly used with Particle Swarm [EK95] and they are unified with the DE scheme in [CDK06]. In this variant of DE the population order is considered stable and each particle has a k-neighborhood of other particles from which to draw influence. This is formulated by considering that a population forms a cycle such that

$$\left\{\dots p^i_{-2} = p^i_{N-1}, p^i_{-1} = p^i_N \dots p^i_{N+1} = p^i_1, p^i_{N+2} = p^i_2 \dots\right\}.$$

Thus the k-neighborhood for vector p^i_j is $\left\{p^i_{j-k} \dots p^i_j \dots p^i_{j+k}\right\}$.

Let p^i_{kbest} be the best candidate vector within the k neighborhood of particle i and p^i_{gbest} the best overall vector in the population P^i. The population is updated by a local update formula which, for particle j, can be defined as

$$L^i_j = p^i_j + \alpha\left(p^i_{\mathrm{kbest}} - p^i_j\right) + \beta\left(p^i_a - p^i_b\right),$$

where p^i_a and p^i_b are random members of the k-neighbourhood of p^i_j. A local update complements the global update formula (also used in variant DE/best/1):

$$G^i_j = p^i_j + \alpha\left(p^i_{\mathrm{gbest}} - p^i_j\right) + \beta\left(p^i_c - p^i_d\right),$$

where p^i_c and p^i_d are random members of the population P^i and α and β have a role equivalent to scale factor F of standard DE.

These update formulae are combined when generating the offspring individual o'_j by

$$o'_j = wG^i_j + (1-w)L^i_j.$$

The weight w is set according to the following equation:

$$w = w_{\min} + (w_{\max} - w_{\min})\left(\frac{i}{I_m}\right),$$

where I_m is the maximum number of iterations that the algorithm is allowed to run. The rest of the process follows that of DE given in Section 2.4.6. The parameters for this algorithm include neighborhood radius k and weight factors $w_{\max} \in [0,1]$ and $w_{\min} \in (0,1]$.

2.5.3 Self-Adaptive Differential Evolution with Neighbourhood Search (SaNSDE)

SaNSDE [YTY08] is an amalgamation of two different algorithms, Differential Evolution with Neighborhood Search (NSDE) introduced in [YYH08] and Self-Adaptive Differential Evolution (SADE) introduced in [QS05].

This algorithm consists of three different additions to the basic DE scheme: Mutation strategies self-adaptation, scale factor self adaptation, and, weighted crossover rate self adaptation. The first of these components hybridizes the two different schemes of DE, *DE/rand/1* introduced in Section 2.4.6, equation (24), and *DE/current to best/2* which uses the following formula for the off-spring generation:

$$o_j' = p_j^i + F\left(p_{best}^i - p_j^i\right) + F\left(p_b^i - p_c^i\right), \tag{30}$$

where p_{best}^i is the fittest member of i^{th} generation and p_b^i and p_c^i are randomly selected individuals of the same generation. As before, F denotes the scale factor.

SaNSDE chooses between equations (24) (*DE/rand/1*) and (30) (*DE/current to best/2*) with probability ρ of selecting the first and $1-\rho$ for the latter. The value of ρ can be seen as rate of success between the different strategies. It is updated after every 50 generations to the following value:

$$\rho = \frac{s_1(s_2+f_2)}{s_2(s_1+f_1)+s_1(s_2+f_2)}, \tag{31}$$

where s_1, s_2, f_1, f_2 are, respectively, the number of successful and unsuccessful attempts of generating offspring by either equation (24) (s_1 successes and f_1 fails) or equation (30) (s_2 successes and f_2 fails) since the last update of ρ.

Scale factor self adaptation is done with the same principle. Unlike the static F in plain DE, SaNSDE uses a choice between value $0.5 + g_{0.3}$ (i.e. 0.5 centered gaussian random number with $\sigma = 0.3$) and random sample from cauchy distribution with scale parameter of 1. The choice is made in identical manner to mutation strategy adaptation.

The most complex adaptation is the weighted crossover rate self-adaptation. Every five generations new Cr values for each individual are generated for the new population using the following equation:

$$\mathrm{Cr}_j^i = \mathrm{Crm} + g_{0.1}. \tag{32}$$

Crm is the weighted mean of those crossover rates that have produced a successful off-spring:

$$\mathrm{Crm} = \sum_{k=1}^{\#\mathrm{crs}} \mathrm{crs}_k \frac{\mathrm{fs}_k}{\mathrm{fa}}, \tag{33}$$

where crs is the set of recorded Cr values which were used to generate successful off-spring and fs_k is the corresponding improvement in fitness, normalized by the average fitness improvement fa.

2.5.4 Differential Evolution with Adaptive hill Climbing Simplex Crossover (DEahcSPX)

Differential Evolution with Adaptive hill Climbing Simplex Crossover (DEahcSPX) is a memetic variant of DE introduced in [NI08] which employs a local search within the generation cycle. More specifically, at each generation the best individual undergoes a simplex crossover local search.

This local search method works by updating a set of points sampled from the current population P^i, denoted as $X^0 = \{x_1^0, x_2^0 \dots x_n^0\}$, until the search converges.

Let O^k be the center of mass of X^k at k^{th} iteration of SPX local search:

$$O^k = \frac{1}{n} \sum_{j=1}^{n} x_j^k,$$

and r_j a diminishing random sequence

$$r_j^k = u^{\frac{1}{j+1}},$$

which is later used to weight the candidate points y_j^k.

The candidate points y_j^k are generated by

$$y_j^k = O^k + \epsilon \left(x_j^k - O_j^k \right).$$

The parameter ϵ is a user defined control parameter for this method. The y_j^k is then applied in generation of the candidate off-spring t^k:

$$t'_j = \begin{cases} 0, & \text{when } j = 1, \\ r_{j-1}^k \left(y_{j-1}^k - y_j^k + t'_{j-i} \right), & \text{otherwise,} \end{cases} \tag{34}$$

for $j = 1, 2, \dots, N$, which yields the final candidate point

$$t^k = y_n + t'_n.$$

At each generation i the best solution is selected along with $n - 1$ randomly picked solutions as the set X^1. Then a sequence of trials $\{t^k\}$ is generated using the above process and replacing x_1^k by t^k in the set X^{k+1}. After this process the best solution is replaced by the best solution discovered by this method.

2.5.5 Comprehensive Learning Particle Swarm Optimization (CLPSO)

The original PSO update rule allows particles to follow only the best individual in the flock. In [LQSB06] this is conjecturized to cause premature convergence when the best particle lands in a local optimum causing others to follow. Comprehensive Learning Particle Swarm Optimization (CLPSO) is proposed to alleviate this by changing the update rule in a way that each particle can contribute to the trajectory of others.

This version is identical to the one given in Section 2.4.5 except that the update formula in equation (21) is replaced by the following:

$$v_j^{i+1} = \max \left(M_v, c_1 v_j^i + c_2 u (\mathrm{sb}_j^i - p_j^i) \right). \tag{35}$$

Here sb is a vector composed element-wise from the pb:s in the flock:

$$\mathrm{sb}_j^i = \begin{cases} \mathrm{sb}'_j, & \text{if } \mathrm{f}(p_j) \text{ has not improved for } m \text{ generations,} \\ \mathrm{sb}_j^i, & \text{otherwise,} \end{cases} \tag{36}$$

where

$$\mathrm{sb}'_j[m] = \begin{cases} \mathrm{pb}_j^i[m], & \text{with probability } Pc_j, \\ \mathrm{pb}_a^i[m], & \text{otherwise,} \end{cases} \tag{37}$$

where pb_a^i is the location of smallest value discovered by a particle selected by tournament selection from the current population P^i. Vector sb_j^i is not updated until m generations pass without improvement by the particle. Value of Pc_j for particle j has been set empirically to

$$Pc_j = 0.05 + 0.45 \frac{\exp\left(\frac{10(i-j)}{N-1} - 1\right)}{\exp(10) - 1}, \tag{38}$$

in the original paper. Tests in that paper and in this thesis ([PIV]) support the conclusion that CLPSO is much more robust than the original PSO.

3 ON MEMETIC ALGORITHMS

3.1 Memetic optimization

Memetic algorithms (referred to later as MAs) are the result of hybridization of evolutionary optimization algorithms with other methods, such as various local search schemes. The original philosophical basis of MAs, introduced by Moscato in [MN89], is the concept of memes or elements of lifetime learning. Whereas genetic algorithms model optimization as generations of solutions that pass their performance onwards in genes, MA model tries to include lifetime learning that is passed on to other members of the population [ES03]. The idea has drifted largely from the original inspiration, but memetic algorithms remain a successful way to both speed up optimization and insert domain specific knowledge into the process. Memetic algorithms form another lane of improvements to DE next to adaptive and other schemes discussed in the previous section.

3.1.1 On classification of memetic approaches

Article [LHKM04] proposes a distinction between MAs and algorithms that employs more than one local searcher. While algorithms consisting of only one lifetime learning mode or local searcher are called MAs, algorithms that employ a set of different strategies are named Multimeme Algorithms (MMA). In the latter case the algorithm designer has several simpler parts from which to compose the algorithm, but runs into a problem of coordination between different algorithms. Having more than one meme can be seen as an important feature for an algorithm, since each local search method (aside from true random search) contains a search bias. Thus there is large number of possibilities for actual algorithmic designs.

In order to map the contribution of this Thesis in the field, a small amount of background has to be established. According to the classification given in [OLZW06] various memetic algorithms have been put into the following three categories.

The first category according to Ong is hyperheuristics, in accordance with article [CKS01], which proposes the name. (The reader should note that this term is often used to mean heuristics of selecting other heuristics, in sense of [ÖBK08], which is not to be confused with the use in here.) This class of algorithms works with sets of a priori-rules that govern the behaviour of the search. These rules can be customized to each problem, but are in a sense rigid, accepting no feedback from the dynamic functioning of the algorithm. Hyperheuristics can be contrasted with the second grouping of algorithms, namely meta-lamarckian learning.

Meta-lamarckian learning [OK04] comprises a multi-meme system that is co-ordinated by performance of memes. The improvements due to different memes contribute to a process of selecting the memes to be applied in the future. Thus in order to clarify the differences of our approach, meta-lamarckian systems can be said to co-ordinate by local properties of the population. The motivation for such systems lies in ideas of co-operation and competition among the memes as proposed by [MN89].

Third grouping can be seen in works such as [Smi07] and has also been proposed by Krasnogor in [CHK$^+$02, Kra04]. This set of methods can be described as self-adaptive and co-evolutionary MAs. The important feature is that memes are undergoing their own evolution, either as part of candidate vectors (self-adaptive) or in a co-operative population of memes (co-evolutionary). To contrast this against meta-lamarckian learning, there is no direct feedback loop from performance of the memes into their selection, but on the contrary MAs in this class rely on the principles of evolution also on the memes.

To introduce another axis of classification, the notion of generations of memetic algorithms is given according to [NOL08]. In that work, algorithms are said to be 1^{st} generation MAs when cultural evolution, or memes, are present, but they do not evolve themselves. Meta-lamarckian algorithms may be called 2^{nd} generation MA. Thus 2^{nd} generation is characterized by the adaptation of fixed memes. The 3^{rd} generation label is for an algorithm that encompasses true cultural evolution, such as co-evolutionary MMAs. Now, the memes are not fixed before evolution, but can be created during it.

In the work [Ner07] and articles leading to that ([CCN$^+$07b, NTM07] and [NTCO07]) another scheme has been proposed. This scheme deals with fitness diversity based algorithms, that are also focused upon in this thesis. This scheme can be argued to form a fourth grouping in the Ongs scheme of classifying algorithms. Fitness diversity based algorithms handle the issue of co-ordination in a fashion similar to meta-lamarckian by taking into account the improvements caused by different memes. However, this is done by studying global effects on the population instead of local ones as in meta-lamarckian scheme. Also, thus far, no fitness diversity algorithm directly maps the effect of diversity caused by a meme to a probability of activating that said meme. Instead, the systems are pre-designed by the properties of the memes used, so they can be classified as 2^{nd} generation.

3.1.2 Fitness Diversity

The two concepts of exploitation and exploration play a large role in the following. Although they have no general definition, they can be understood in the following way: exploitation works in the area of fitness landscape already entered by the algorithm and exploration by entering new areas of the landscape. As noted in [ES98], coordination is required for a proper balance of exploration and exploitation which seems to be the basic component of a successful EA. This result is also empirically validated in the context of multi-objective methods in [IYM03]. As noted before, each optimization problem has its own properties which reflect on where this balance is found. In addition, it is clear that this balance shifts as the algorithm proceeds [ES03] and the exploration/exploitation ratio should vary over time in order to adapt to the current conditions of the search.

However "current conditions" is not a clear concept. It is not trivial to deduce which phase of optimization is underway due to the fact that we usually don't know the actual optima nor the scale features of the fitness landscape. Some information can be gleaned from the statistics of the population, such as how diversely it is distributed in the landscape and speed of its progression. In this work this coordination of local searchers and balancing of the exploration and exploitation is done by various measurements of quality called Fitness Diversity that was first introduced by this name in [CCN$^+$07a] (Although similar ideas were studied, for example, in [BGK04]).

Fitness diversity proposes to assign a numerical value of the effective diversity of the population by inspecting the fitness values in the active population. This value can then be utilized in guiding the algorithm to select a suitable balance between explorative and exploitative features. When contrasted to a simple spatial diversity measure, fitness diversity only accounts for fitness features of population. A population having a high spatial diversity might be scattered along a large plateau and thus be unable to proceed without further tuning up the explorative features of the algorithm. Conversely, population that has spatially narrow distribution might actually have encountered an area of the fitness landscape that is highly interesting and suitable to be exploited with a local search algorithm. Fitness diversity has been studied in contexts of noise [NKV08, NM07], multiple objectives [CN], and in medical applications [NTCO07, NTM07].

In the equations below, f_{best}, f_{worst}, f_{avg}, and σ_f are the best, worst, average, and standard deviation of fitness values measured at current population and $\max|f_{best} - f_{avg}|_i$ is used to denote the maximum difference observed in the i iterations of the process thus far. The fitness measures studied in this thesis are the following, each being fitted to the $[0,1]$ range in a way that 0 indicates the least diversity while 1 indicates the most:

$$\nu = \min\left\{1, \frac{\sigma_f}{|f_{avg}|}\right\}, \tag{39}$$

which measures how sparsely the fitness values are distributed,

$$\xi = \min \left\{ 1, \left| \frac{f_{best} - f_{avg}}{f_{best}} \right| \right\} \tag{40}$$

is used to measure the advantage the best individual has compared to the average,

$$\psi = 1 - \left| \frac{f_{avg} - f_{best}}{f_{worst} - f_{best}} \right| \tag{41}$$

measures how far in the scale of fitnesses does the average fitness value lie, resembling measurement of skewness. Furthermore,

$$\phi = \frac{\sigma_f}{|f_{worst} - f_{best}|} \tag{42}$$

is an amalgam of equations (41) and (39) measuring how sparsely the fitness values are distributed within their range of variability. Unlike ν it is co-domain (i.e., a range of fitness function) independent measure. Lastly, we define

$$\chi = \frac{|f_{best} - f_{avg}|}{\max |f_{best} - f_{avg}|_i}, \tag{43}$$

which differs from the previous measures of fitness diversity in that it takes into account the history of the process. The purpose of χ is to measure how much the leading individual outperforms the rest of the population. The existence of a super-fit individual is noted by high values of χ and is considered beneficial for the DE process. Measures ν, ϕ, and χ have been introduced for the first time in the included papers.

3.2 Novel Memetic Computing Schemes

This section introduces the author's work on Memetic Differential Evolution Frameworks. The algorithms are presented in this section in detail while the results achieved with them are provided in Chapter 5. The whole family of algorithms is denoted as *MDE.

In this work memetization is approached by dividing the optimization cycle to several phases in contrast to solutions applying local search within the offspring generation of an algorithm. A multi-meme algorithm constructed in this way can be represented with the schematic in Figure 3. The initial generation of population is done here by pseudo-randomly sampling the fitness domain for the number of individuals equal to the size of the population. Other alternatives include methods such as using structured point sets surveyed in [Maa04].

Further refinements can be employed to speed up the search. The methods for doing so include such strategies as generation of extra individuals and selection of best performing individuals for the optimization or methods that attempt

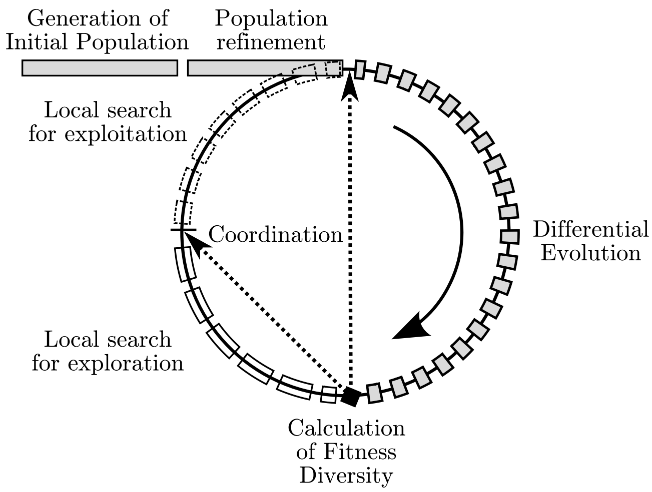


FIGURE 3 Generic Memetic DE schematic. The circular path represents the steps taken by each full iteration of the algorithm and different styles of line describe different components. The dotted line represents conditionality – some steps may be skipped according to internal logic of the method.

to, for example, introduce super-fit individuals into the population in order to generate efficient search directions as is done in article [PIII]. The DE search is then applied for a fixed number of fitness evaluations and the fitness diversity of the resulting population is measured. This measure determines whether to apply a local search to the population or not. If the local search is applied, it is done from the most explorative to the most exploitative in order to be able to refine potentially successful solutions discovered in this phase. Each local searcher is applied for a fixed limit of fitness evaluations.

It can be noted that each algorithm introduces a set of new parameters in addition to those in DE. This might seem contradictory to the goal of having a more automatic algorithm, but the tests with general test functions have proved (in articles [PV, PVI, PVIII]) that these parameters are not extremely sensitive and seem to cover a large range of conditions. Naturally, tuning is beneficial, but not essential.

Although there are several possible design choices, the most studied in this work are the adaptation schemes (in [PVII]), different fitness diversity measures (in [PV]), and suitability of different local search heuristics (in [PII] and [PIII]).

3.2.1 Memetic Differential Evolution (MDE)

Memetic Differential Evolution (MDE) is introduced in article [PI]. MDE is an adaptive evolutionary algorithm which combines the powerful explorative features of Differential Evolution (DE) with the exploitative features of two local searchers. The local searchers are adaptively activated by means of a novel control parameter based on measures of fitness diversity within the population.

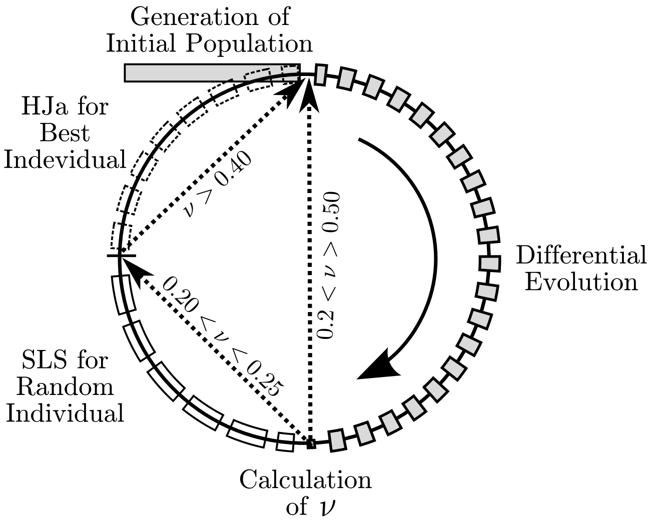


FIGURE 4 MDE schematic, showing the order of DE, HJa and SLS algorithms and control scheme used in the coordination

MDE scheme is depicted in Figure 4 and consists of DE framework detailed in Section 3.2 augmented with HJa and SLS. The coordination of searchers is done according to the fitness diversity measurement. DE process is applied and in every 1000 fitness evaluations there is a possibility of a local searcher activation according to ν-index in equation (39).

According to our basic conjectures about functioning of memetic algorithms, local searchers are activated when diversity becomes low in order to offer new exploratory perspectives by revealing new and promising genotypes. Also, in a possible case where search has entered the optimal basin of attraction the local search will find the solution of the problem in a quick fashion.

The two searchers have different characteristics and are thus activated by means of a different logic based on simple threshold values. To provide uniform presentation across the various algorithms, this is depicted in Figure 5 as the probability of activating the searcher versus the diversity measure. The actual activation scheme averaged over several runs can be seen in Figure 17b.

Local Search activations start with SLS which is supposed to assist DE by improving available genotypes in order for them to be exploited by DE. Deterministic HJa is applied in a later stage to the best individual in hopes of finalizing the optimization.

3.2.2 Enhanced Memetic Differential Evolution (EMDE)

The Enhanced Memetic Differential Evolution (EMDE) is a direct descendant of MDE. During the testing phase of the MDE algorithm it became apparent that often slight changes in the threshold of local search activation made a large difference to the outcome, especially to the development of the diversity measure. This

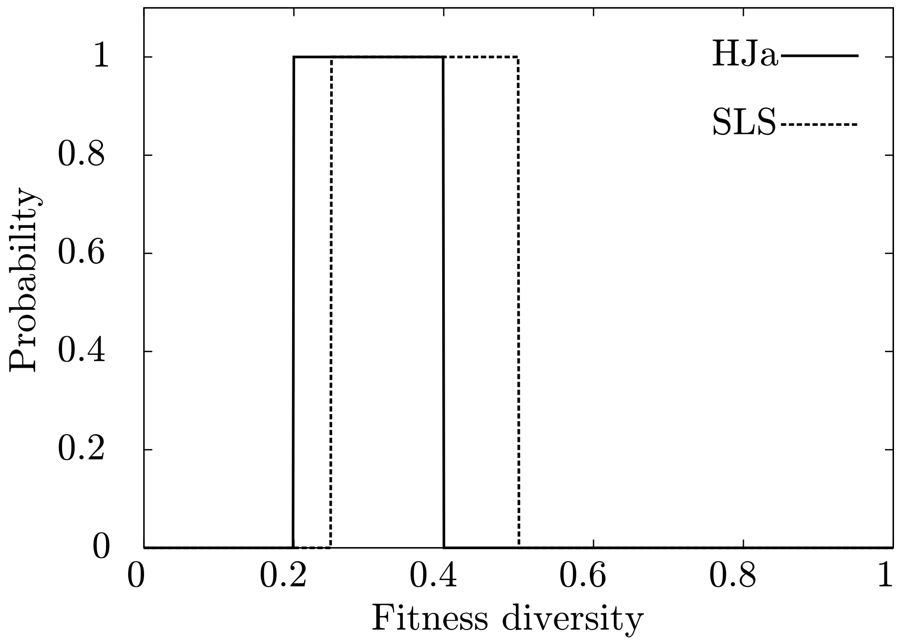


FIGURE 5 Probability of activating local searcher vs. fitness diversity index in MDE

problem was exacerbated by the introduction of more local searchers. To alleviate the problem, static thresholds for local search were discarded and replaced with a probabilistic scheme, where LS algorithms are activated with a higher probability when fitness diversity approaches the set limit. The probability for activating a local searcher is given by

$$\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right), \tag{44}$$

where μ is the point of highest probability of LS activation and σ denotes the range where a local searcher is active.

This allows for an easier specification of a local search scheme and is at the same time empirically more robust over sets of runs than static thresholds. Random selection stops the algorithm from getting stuck in a rut when one searcher systemically fails. The graph displaying the LSA activation probabilities is given in Figure 6.

Also, the repertoire of local searchers has been increased from that of MDE. Efficiencies of local searches for the target domain of filter design have been empirically measured in order to estimate when and to which individual they are applied to. Also, SA is added in order to allow the algorithm to escape local minimae. The resulting scheme can be seen in Figure 7.

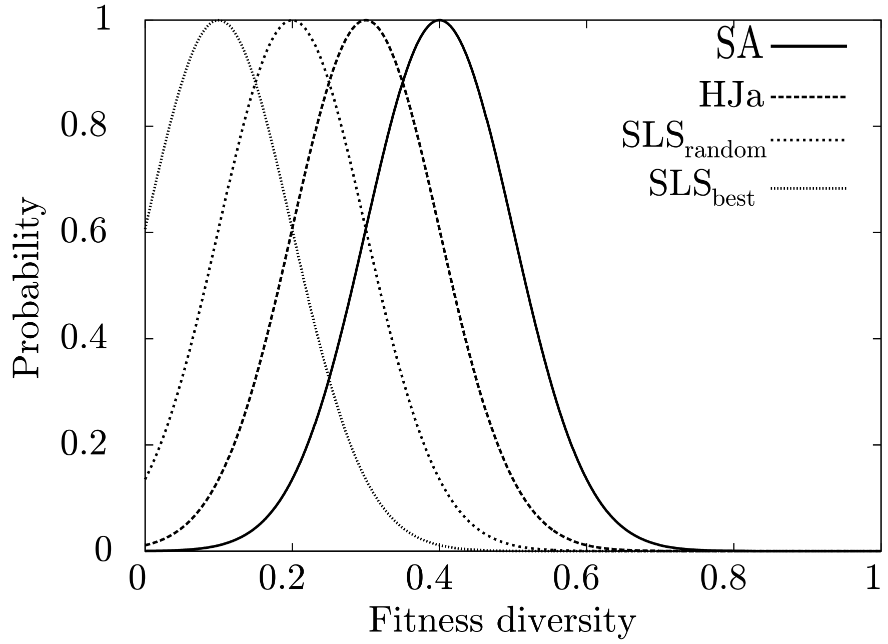


FIGURE 6 Probability of activating local searcher vs. fitness diversity index in EMDE

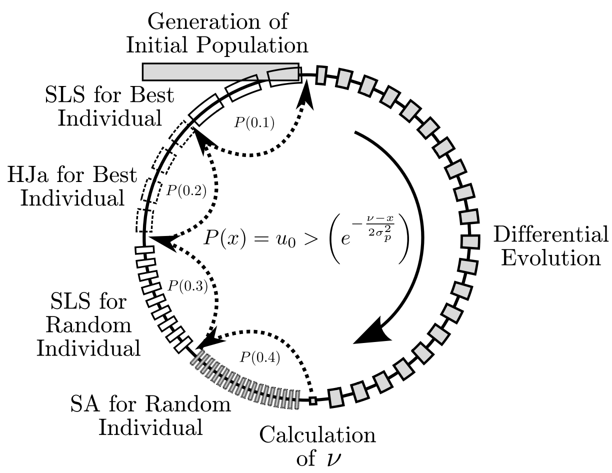


FIGURE 7 EMDE schematic. This schematic details the probabilistic control scheme applied and definition of the probabilities employed (P).

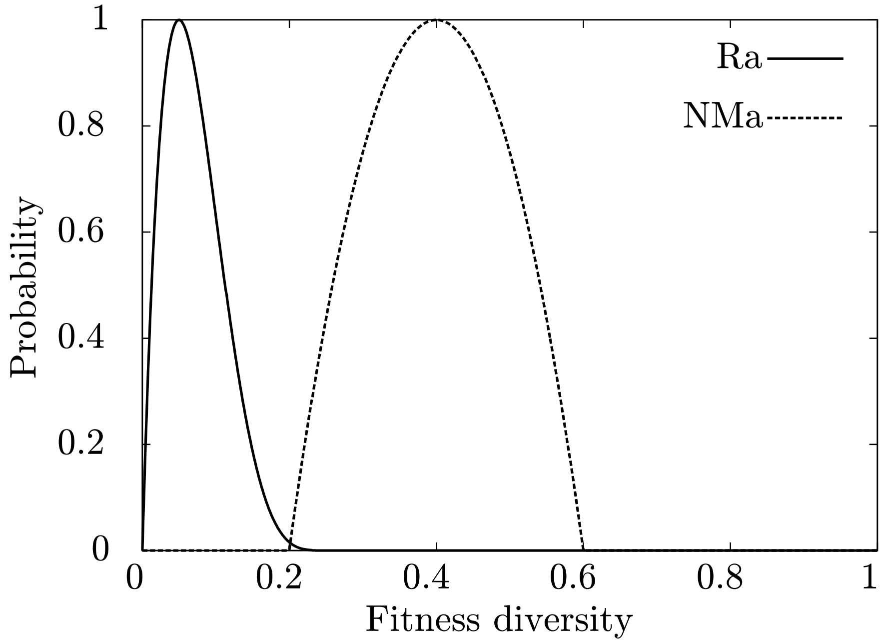


FIGURE 8 Probability of activating local searcher vs. fitness diversity index in SFMDE

3.2.3 Super-Fit Memetic Differential Evolution (SFMDE)

In design philosophy, Super-Fit Memetic Differential Evolution (SFMDE) deviates from the previous work. In this case the coordination is based on a measure of describing how large improvements are achieved when contrasted to improvements in the previous generations. Also, as an important addition, the search is accelerated by applying PSO to the initial generation. This hopefully results in an individual that is vastly better than its peers. According to DE logic, this individual will then contribute good search directions to the DE process, speeding up the search.

Local searchers have been replaced by NMa and Ra, which are potentially more efficient. The probabilistic local search activation scheme has been retained from EMDE. The difference lies in the shape of functions used to determine the activation probabilities. EMDE uses simple symmetrical Gaussian functions while SFMDE uses non-symmetrical ones inspired from beta-distribution, giving more control for the design of the local search activation scheme. According to experiments in paper [PIII], the activation functions were selected as shown in Figure 8. The basic operating principle and the activation function details are provided in EMDE schematic in Figure 9.

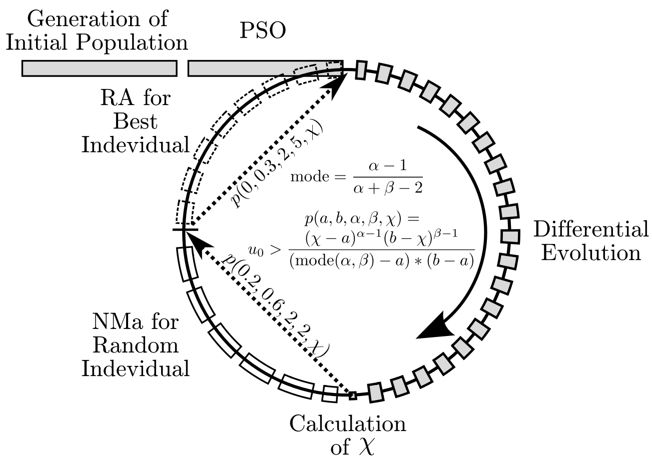


FIGURE 9 SFMDE schematic. This schematic details the initial refinement of the population and the probabilistic control scheme applied along with the definition for function for calculating them (p).

4 APPLICATION TO LOW LEVEL MACHINE VISION

4.1 Introduction

We have entered the world of excessive digital media. Much of the massive amount of data generated is in images. Internet magazines, databases and especially user oriented image and video services such as *Flickr* or *youtube* are acquiring image data at immense rates. In August 2006 The Wall Street Journal [Lee06] reported that there were already 6 million user submitted videos in YouTube. Communal photo management service *Flickr* has reported [Ote07] to have arrived to 2 billion images in November 2007 (the image no. 2 billion is presented in Figure 10). The needs for automatic processing of image data are enormous.

The human way of perceiving the world is also applied to process industry, which provides no exception to the tendency for increasing amounts of acquired data. More and more sensors are developed as the processes are improved and more tightly controlled. Vision is a key part in advanced sensors, since visual inspection is both easy to achieve with modern cameras and is in concordance with 'the human way'. Modern imaging equipment have far surpassed human vision but the problem of automatic image processing remains. The increase of the capacities of computational equipment gives rise to increasing possibilities of processing both static and video images.

Process industry and product monitoring by human operators is, and is usually depicted as [Cha36], comically boring and monotonous work performed by people. Machine vision has gained a strong foothold in the process industry due to the fact that computers are extremely well suited for these monotonic and repetitive processes that humans are distinctly unsuited for. Process industries have been among the first to adopt vision systems for their purposes.

Vision systems are used in both quality and process control. Many materials can be characterized at suitable depth by their visual properties, indeed these are mostly the properties that humans use to estimate quality. Automated quality control can measure properties such as texture or shape in order to evaluate quality, and these same measurements can be used in process control in order to

FIGURE 10 Representation of image number 2000000000 of Flickr

adjust processes. Also, visual systems have the ability to measure many things that otherwise would require specialized sensors.

The field of machine vision combines a wide range of scientific fields that aim to bring the power of vision for mechanical devices. This cross-disciplinary field consists of digital image processing, signal processing, engineering fields dealing with image acquisition and lighting, machine learning and is applied widely throughout the industry. The three main areas of machine vision are remote sensing, quality control, and process control. Machine vision is a growing field due to advanced computational machinery and vision systems becoming more commonplace.

Almost all fields of industry have vision related processes, such as medical uses of angiography, x-ray tomography, and even ultrasound for health care leading to, e.g., whole journals like IEEE Transaction on Medical Imaging and books such as [Web88, CJS93, BA94]. Even traditional areas such as agriculture benefit from automated vision [CCK02] in forms such as quality control of foodstuffs like apples [LMD98] and precise spraying of pesticides, thus reducing costs and even environmental impacts [GS97]. Other fields have their own equivalent examples even from their rather early period of machine vision, as among them metal industry [LHN89], robotics [Hor86] as well as numerous other fields.

Divided to different tasks, machine vision problems can consist of such tasks as separating objects from background like the one in this thesis or in several medical applications [HKG00, SLCJ+05, BSdJ+99], enhancing images for human inspection [GSZ04], identifying objects such as pedestrians or cars in cluttered backgrounds [XLF05, SBM06], classifying found objects into different categories [BLE+04], sensing changes in an object such as facial expression recognition [LKCL00], content based image retrieval [LSDJ06] and, naturally, controlling machinery in situations where human presence is not possible such as in planetary landers [RSC+07].

Machine vision of today is in an immature state where the studies conducted have a huge diversity and only a relatively small part of the field is considered to be well established, although individual solutions for many problems exist. The field can be roughly partitioned to different approaches. Some problems are tackled with image transform based approaches that examine the image in some other domain than the normal spatial view. Such approaches include Fourier transforms to frequency domain and wavelet transforms. These methods have been utilized for both identification [CC06] and denoising [KMM01]. This view also leads to perceiving the field of image processing as an subfield of two-dimensional signal processing. Other method families process the image in spatial domains (examples of which include Local Binary pattern methods, see for example [OPM02] or SUSAN method family [SB97]).

Figure 11 provides a rough depiction of a machine vision process. For practical purposes, the process begins with the selection of radiation sources suitable for properties of a target object and the selection of a suitable sensor. This creates arguably the core of the system, but from the point of view of a computer scientist the process begins after an acquisition of a digital image. From there on, tasks include identification of object boundaries, silhouettes and details such as corner points and calculation of various descriptors for objects in the image. These low-level tasks provide information that can be used to support higher level image-understanding processes that generate models or information (by, for example, identifying objects) that can be used to control industrial processes or support the human operator in decision making.

Figure 11 can be contrasted with definitions in classical handbooks such as [GW07], or [SHB98] which have focus on computer vision. The latter divides the process after image acquisition into *low-level* and *high-level* processing. Low level deals with data that closely matches the image and such processes are said to have very little knowledge of the content of the images. Low level tasks comprise of preprocessing such as noise filtering, edge extraction and in general work on the images themselves. High level processes are about image understanding and semantics of the situation where the image contents are fitted into some model of the world. Usually, according to Sonka, high level operations do not operate directly on images, but on features derived by low level operations. A high level functionality need not appear in a computerized form. A human operator can just as well benefit from image enhancements acquired via low level systems.

In [GW07], the process is divided into steps of *image acquisition*, *preprocessing*, *segmentation*, *representation&description* (aka feature extraction), and *recognition&interpretation*. The middle three can be said to be low level operations and the last a high level one. Also, it is necessary to note that all of these low level operations need not appear in all machine vision applications. In some cases, what would have been done with preprocessing can be achieved by, for instance, different lighting. Also, it is easy to imagine cases where the user is more interested in, for example, the texture of product than its shape. The imaging problem in this thesis follows the scheme of [GW07] although focusing on low level operations.

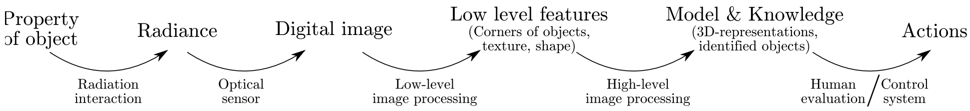


FIGURE 11 A General outline of machine vision process

4.2 Machine Vision in Paper Industry

In recent years, machine vision systems have started playing essential roles in process industry of which paper production is a definitive example. A modern paper mill converts huge amounts of pulp into paper. Such a mill consist of a long process line that converts pulp into paper with rates nearing 30 meters per second. This process is hardly perfect and the quality of product is sometimes compromised. In worst of the cases, if the paper being processed develops a large hole or tear the whole process may come to an abrupt and costly halt. Other defects, such as insects squashed onto what is to become a side of a milk carton or small holes that allow newspaper printer's ink stain expensive equipment, are undesirable as well.

The process of handling such defects is far from simple. Without tools, a human cannot hope to detect such a flaw when system is running nor react to it in time. Within seconds a small flaw that existed at the starting end of the process is rolled up with the rest of the paper in the finished product (or has caused a total failure and none of the paper ends up in the roll). Thus an automated inspection system is required.

Such systems are used for fault detection as well as for quality control of the product. If the process produces a defect, such as a hole in the paper, it is important that this is detected before it ends up in the final product. It is mandatory that these faults can be detected and marked in such a way that they can be removed from the final product and their causes identified and corrected before they happen again.

Development of machine vision systems is due to both advances of imaging technology and in paper production machinery, which currently surpass by far the capabilities of human senses. Modern paper mills have the capacity of producing hundreds of square meters of paper in a second while humans have the capacity of perceiving defects less than a millimeter in size when handling finished products. A huge amount of data has to be analysed every second. The task induces a significant data flow due to high resolution and vast area to be imaged. Besides being outside of human reach, such demands place considerable stress even on modern computational equipment.

One of largest applications are quality control systems that perform defect detection. A defect detection process scans the paper web for possible defects in paper formation so that such defects can be removed from the final product and their causes be rectified for future production. The outline of this process is given in Figure 12. The process begins with a defect detection phase, which gives rise

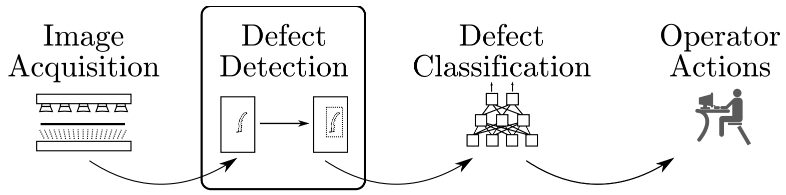


FIGURE 12 Defect detection process

to the highest real-time requirements.

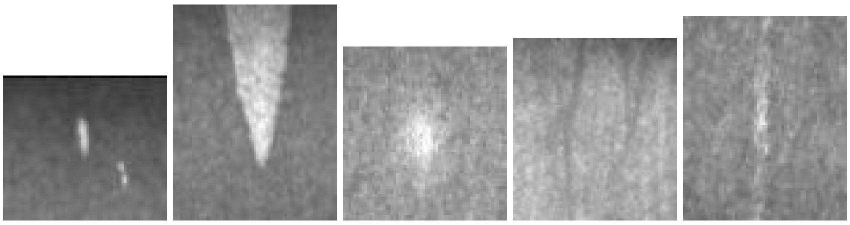
The detection is followed by a high level identification process consisting of an image classification problem. A simple flaw indicator is of limited use if the defect detection system cannot tell whether it is worth an action or not. Some defects like water droplets are important only in the quality control sense but certain defects will require immediate action and thus discriminating these is highly important. Also, even those defects which are not harmful towards the process need to be cataloged properly in order to enact suitable quality control measures. Classification process is naturally a multi-class problem. In basic work cataloguing various defects [otPcS95] there are 76 paper web defects that have been named. This process has some real-time constraints but not in the scale of detection phase: as a high level system, this part operates on the small amount of suspicious data extracted by low level processes. Usually most of the paper area is defectless.

Also, machine vision systems have other applications in paper industry, such as monitoring nozzles and other equipment as well as statistical quality control and measurement of other qualities of paper, such as printability and texture [TPS$^+$03].

4.2.1 Defect detection in paper manufacturing process

The defect detection process is eminently a low level vision problem. The goal is to sense various defects from the acquired digital images and consists of constructing an imaging system capable of operating in conditions of paper mill and low-level algorithms that are both enough computationally efficient for the task and precise enough to detect potentially very weak defects.

In the defect detection phase image is acquired from an on-line system. Usually trans-illumination is utilized as described in article [PI]. Reflective illumination is used when either the defect is of such nature that different lighting helps to reveal it or there is no suitable space for attaching proper lighting equipment. Each millimeter of paper is imaged and processed for detecting the presence of defects. Defects vary from simple holes (see Figure 13a) that are readily apparent in trans-illumination to harder-to-see defects, such as wrinkles and streaks (Figures 13d-13e) that are of faint contrast and possibly of larger size. Some defects like holes may result in interrupted processes, but others such as water droplets or thin spots (Figures 13c and 13b) are benign. They are either quality defects or



(a) Holes (b) Water droplet (c) Thin spot (d) Wrinkle (e) Streak

FIGURE 13 Various defects encountered in paper production

vanish during the process entirely.

This detection phase is the most time critical one, since it is mandatory to process the whole massive data-flow in its entirety. This requires special hardware for computational requirements alone and no complex method can yet be implemented. The higher level processes associated with the task consist of classification of the detected anomalies and using accumulated information in process control. Although hard tasks, these higher level functions have relaxed real-time requirements (due to low-level processes filtering out the non-defective parts of paper, which consist of well over 99% of the information processed) and can be tackled with well known methods from machine learning [RN95].

This low-level and real-time task is commonly solved in industry by various thresholding methods. Thresholding can successfully be used to detect the most obvious defects and defects that could violently interrupt the process (e.g. in [PC02]). Unfortunately, there also exists defects that are characterized by such low visibility (i.e., Low Contrast-to-Noise Ratio, CNR) that they are not amenable to thresholding but are still harmful to the quality of the product. Also the background of the acquired images is inherently noisy and characterized by irregular texture. Thus methods estimating the background such as in [CP00, SZ05] are not applicable.

Weak wrinkle and streak type defects are the focus of application oriented articles in this thesis. They appear as weak line-like structures in the paper (see Figure 16a) and are hard to detect via thresholding methods. More advanced methods are required for this class of defects. Such methods must always balance between computational efficiency and accuracy. The main approaches include many techniques known from basic low-level vision studies, such as edge detection, texture analysis etc. Some of these methods even combine elements from both high- and low-level processing, and include self-organizing maps (SOM) and variants thereof as mentioned in [IHR$^+$00, IRPR04]. These solutions are inherently texture-based due to their reliance on histogram-like features that apply over wide areas of the image. Although more precise than thresholding based methods, reliance on texture may harm the process in case of weak and especially small defects which do not cause statistically observable deviation from the texture. There also exists methods relying on various image transforms such as [Sob05] which is based on a set of tailored filters applied to different wavelet sub-bands.

4.3 Low-level Vision

This section delves in more depth into operations used in the included papers and to the concept of low-level vision. Low-level machine vision is not formally a defined concept. In the context of this work it is taken to mean such set of operations that do not belie any impression of image understanding. Low level operations provide data, not information (in the sense of [Ack89]). Such operations are also characterized by relative efficiency and by their simple nature. Examples of low level processes include edge detection, feature extraction, corner detection, and noise reduction and thresholding. [1] Another characterization might be: operations done after image acquisition but before image understanding.

The field of low level processing is wide and methods contained often overlap significantly with the field of image processing. The classical methods in this area consist of linear filters, image transforms, morphological and other rank filters, and colour space mappings such as contrast stretching. Most of these methods appear in any basic image processing and/or machine vision handbooks (for reference see, [GW07, SHB98, Bov05, Jai89].) In modern times these basic methods have been complemented by approaches using classical high level methodologies such as neural networks in low level tasks [SBO94, AK94]. Also, one important part of low level vision is feature extraction (or "description" according to [GW07]) which reduces massive amounts of data in an image to several descriptive features that can be utilized by higher level processes.

We focus here on a small sector of low level machine vision, namely on the processes that can be formulated by a combination of linear convolution and thresholding, which are arguably one of the most used and most important operators in low-level vision. They are most likely present in some form in nearly all low-level processes. Also, in the included articles it is precisely these two processes that are used. The following sections aim to explain these two in detail. For the rest of the field of low level processing, the reader is advised to refer to textbooks mentioned in the previous section.

In the following, images are represented as functions $[0..w] \times [0..h] \rightarrow \mathbb{R}$, where $w, h \in \mathbb{N}$. These functions map pixel-coordinates into real-valued pixels approximated by floating point arithmetic. Pixels are represented by real numbers in the range of $[0, 1]$, but in many cases allowing values outside these limits simplifies matters considerably, when considering operations such as convolution or arithmetic on images.

[1] Though noise reduction in many cases is neither fast nor simple in nature

4.3.1 Convolution

Convolution is denoted by $*$ and is defined as follows [Bra65]:

$$(F * K)(x, y) = \sum_{(i,j) \in \mathrm{Dom}(K)} F(i, j) K(x - i, y - j), \tag{45}$$

where F denotes the convolved image, K is the convolution kernel (which also can be seen as an image), and (x, y) are the coordinates of a pixel. The notation $\mathrm{Dom}(K)$ denotes the domain ($[0..w] \times [0..h]$) of the kernel matrix. A convolution operation can be used to implement various linear finite impulse response filters. Also, due to the convolution theorem [Jai89], they can be implemented efficiently even with large kernel sizes via fast fourier transforms (For details, see [SHB98], or any basic textbook).

Various convolution filters are used to perform many low-level image processing tasks such as noise reduction via running average or Gaussian filters, edge detection with suitable edge masks, such as Sobel, and image enhancements like unsharp masking. Although in most of these application areas newer methods have gained ground, simple filtering remains very effective and efficient method for many tasks.

4.3.2 Gabor filters

The name for Gabor filters comes from the work of Nobel laureate Dennis Gabor. In 1946 he proposed modelling signals with combinations of elementary functions that later became known as Gabor functions [Gab46]. This work originated partly from the fact that spatial and frequency components of a signal can be seen as idealised duals,i.e., there is often cases where both frequency and time (or rather place when dealing with images as signals) are required, thus giving rise to various time-frequency transforms for which the Gabor functions can be used.

The formulation of Gabor function in the context of image processing comes from Granlund [Gra78] who was "In Search of General Picture Processing Operator". This work was done independently of Gabor functions and specifically for image processing and yet yielding the same function as that of Gabors, generalized to two dimensions. In this work this function is primarily used as a signal analyzing filter instead of as signal synthesis function. The general idea is to use this single operator in all of the levels of the process of machine vision.

2D-Gabor function is often attributed to Daugman [Dau85], who among other things documented relations between Gabor functions and the mammalian vision cortex, hinting that the Gabor function is especially suitable for machine vision purposes. This appears true, since studies have successfully been conducted on many subjects of low and high level vision where Gabor filters have many functions such as, edge detection [MNR92], texture segmentation [MM00], face recognition [LW02] and notably, in Daugmans work, iris identification (see, for example [Dau03]). As can be seen from this set of works, the Granlund's idea is certainly not invalid: Gabor functions can be applied in both low level such as edges and in construction of feature spaces for high level processes.

Mathematically Gabor functions are essentially, in a spatial domain, sinusoidal functions caught in the Gaussian envelope:

$$Gb[\theta, \psi, \sigma_x, \sigma_y, \lambda](x, y) = \exp\left(\frac{-(x\cos\theta - y\sin\theta)^2}{2\sigma_x^2}\right)\exp\left(\frac{-(x\sin\theta + y\cos\theta)^2}{2\sigma_y^2}\right)$$
$$\cos\left(\frac{2\pi(x\cos\theta - y\sin\theta)}{\lambda} + \psi\right). \quad (46)$$

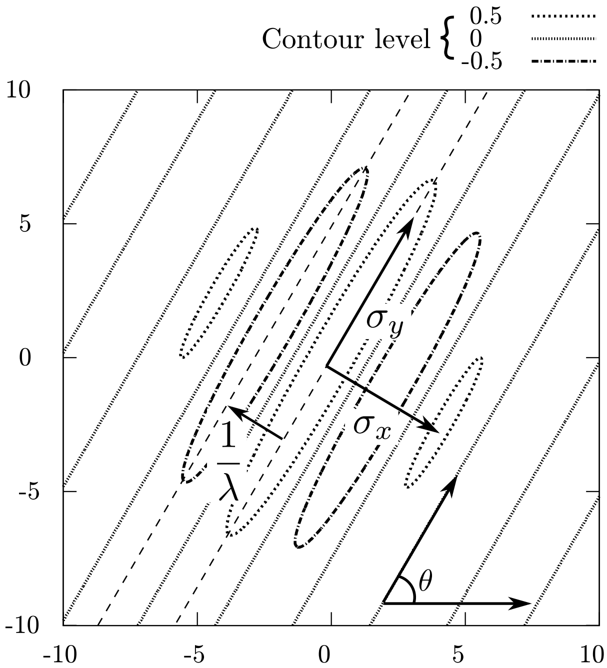
Here θ is the angle between the vertical axis and parallel stripes of the filter, ψ is the phase offset (the filter is symmetrical when $\psi = 0$ and anti-symmetrical when $\psi = \frac{\pi}{2}$). Furthermore, σ_x, σ_y specify both the size of the filter and its ellipticity, and λ is the wavelength of the filter. In other words, we use a directed bandpass filter with a bandwidth determined by λ and the ratio $\frac{\sigma_x}{\sigma_y}$. The effect of various parameters is depicted in Figure 14. Gabor filtering is a linear filtering of the source image by a suitable Gabor function or a discrete sampling of a Gabor function. Thus it is implementable via convolution.

Works dealing with Gabor filters in the context of machine vision include, for example, [DH95]. It deals with texture segmentation which was also the application area of Gabor models as early as 1989 in [FS89]. The methodology has also been applied to common tasks of object recognition (for example, by Jain [JRL97] and Kämäräinen in [KÖ3]) and to the complex task of face recognition in [LW02].) Also, Gabor filtering schemes have been found to be useful in other low level tasks such as edge detection in [MNR92].

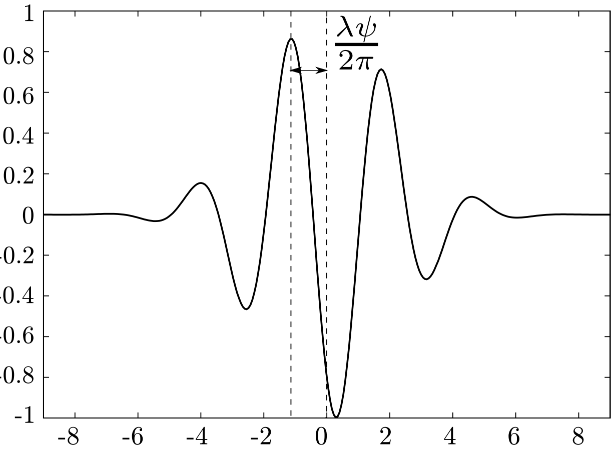
4.3.3 Thresholding

Thresholding refers to binarization of an image according to individual intensities of its pixels. Thresholding can be considered as the simplest of segmentation processes [SHB98] used to extract shapes and areas of a target within the image. Various thresholding methods exist, but they can in general be placed into two categories: Global thresholding methods that determine single threshold for entire image and local methods that use varying threshold in different parts of the image.

A more detailed taxonomy is given in [SS04], where various thresholding methods are grouped into six categories. *Histogram shape* based methods select suitable threshold based on peaks and valleys of a histogram [Sez90], *clustering* methods assume that histograms come from a sum of distinct distributions in the image which can be discovered by unsupervised clustering [Ots79]. Studies of entropy of background or target areas, or cross-entropy between those yield *entropy* methods [JKW85] and direct comparisons between properties, such as edge similarity, of thresholded image and original image yield so called *object attribute methods* [Tsa95]. *Spatial methods* apply a study of higher order probability distributions and correlations in order to find the proper threshold value [KR79].



(a) Spatial parameters on 2D-Gabor function



(b) Effect of ψ in 1D-Gabor function

FIGURE 14 Parameters of equation (46) explained

The last category in the Sezgins survey is *local* methods that use statistics around a given pixel to decide its class [Nib85].

Thresholding methods are most visibly encountered in a task such as optical character recognition (OCR) where usually dark letters are extracted from a light background (for historical examples, see [WR83]). However, it is important to notice that thresholding is often applied to a processed image, such as an edge-strength map thus making this almost a universally required operation.

As can be deduced from the number of categories, the number of different thresholding methods is large. For example, the aforementioned semi-recent survey of common thresholding methods ([SS04]) contains 40 *selected* thresholding methods in order to perform this relatively straightforward operation. It can be deduced from this number of methods that thresholding is both problem dependent and a required component for most machine vision tasks.

Many elementary machine vision systems can be formulated as a sum of proper convolution and proper thresholding technique. Thus, they can in essence be reduced to an optimization problem of finding the proper convolution kernel weights and a suitable thresholding method and its parametrization.

4.3.4 Measurement of low-level vision system efficiency

As is noted in, for example, in a handbook by [Jah00], differing from many engineering fields, machine vision does not yet have a standardized way of measuring efficiency and error besides the simple method of 'eye-balling the results'. Yet such a measurement is doubly important in this work since the aim is to create an automatic system of tuning the low-level vision process.

The following tries to provide an answer to the question *How can quality of an image be measured in reference to separation of target and background areas*. Two concepts commonly used to determine the quality of foreground-background separation are Contrast Resolution (CR) and Contrast-To-Noise ratio (CNR). They are measured here according to definitions given in [PII], although they have several measures in different contexts, for example [SPJ$^+$04] gives three task specific definitions for CNR. Intuitively, CNR measures how visible the target area is through the noise and CR how visible the target area is when compared to the background. High absolute values in each are favoured. We define

$$\mathrm{CNR} = \frac{1}{\sigma(F)} \left(\mu(C) - \mu(D)\right), \tag{47}$$

$$\mathrm{CR} = \frac{\mu(C) - \mu(D)}{\mu(C) + \mu(D)}, \tag{48}$$

where $\sigma(F)$ is the standard deviation of the image F and $\mu(C)$ and $\mu(D)$ denote the average of the background and target area, respectively. Both of these metrics measure the prominence of the target object with respect to the background.

To actually measure these values, a ground truth must be known for each case, making such measurement eminently a supervised action. In the context of this work, where we deal mainly with a supervised task, this is not really a problem. These measures can be used to characterise the difficulties facing machine vision systems.

4.4 Optimization based approach to weak defect detection

This section details the author's work towards a solution for weak defect detection. Defect detection is formulated as a supervised learning problem, yielding a fitness value. This approach is taken since the defect detection problem is not really well suited for executing complex algorithms on-line. This is due to hard real-time and massive data-flow requirements. As such, computational intelligence must be applied to produce and properly tune extremely simple models that can cope with a given amount of data-flow. In an application considered here a finite impulse response (FIR) filter prepossessing and thresholding is used as a model for defect detecting apparatus. FIR filters are linear digital filters that have the property of finite area of support – the filter response settles to zero after finite and defined range from an input. Such a simple convolution filter with a small kernel is efficient enough to be applied in real time and flexible enough to provide accurate solutions. Although other methods exist, optimization of FIR filters with meta-heuristic algorithms is a promising direction used in other real time problems such as texture segmentation [DH95] and for complex task of vehicle detection in [SBM05]. Based on these examples of optimization based work, the detection task is formulated as an optimization problem in article [PI].

When formulating convolution as an optimization problem with even relatively small filter kernels it becomes apparent that the amount of weights contained in the kernel would result in a very highly dimensional problem. Reasonably sized 7×7 kernel contains 49 weights, which already is too much to be efficiently optimized when coupled with an expensive fitness function containing a traversal over the pixels of the entire training set. As per aforementioned articles, an optimizer friendly problem can be derived by generating the weights according to some simpler function. In this case, the filter is formulated as a combination of Gabor filters introduced in Section 4.3.2.

The benefit of this kind of formulation is that the number of parameters remains constant even if the filter size is changed. This formulation can be considered suitable since it provides a compact representation of a family of filters that not only have been empirically rather successful but are also considered a cybernetic solution as they strongly resemble a functionality found in the mammalian vision cortex [Dau85].

The general scheme of weak defect filter design process introduced in this work is outlined in Figure 15. The process starts with a set of source images and corresponding label images (see Figure 16). The filter model parameters are then optimized in regard to matching the filter output to the structure in the label image.

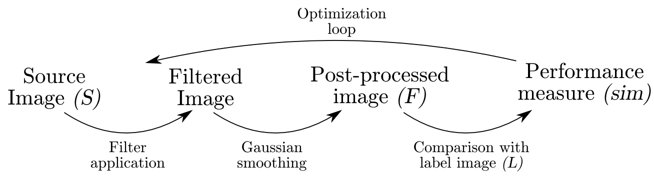
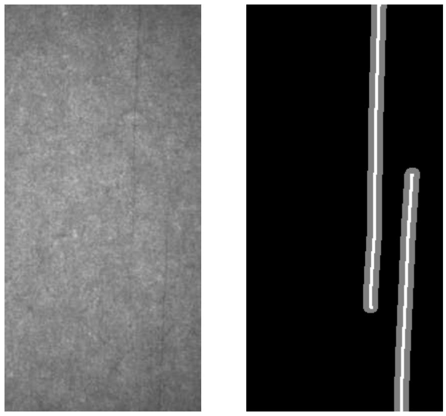


FIGURE 15 Filter design process



(a) Source image (b) Label image

FIGURE 16 Examples of training image pair in defect detection study

Label images define three sets of pixel coordinates (x, y). First is pixel set D which contains those pixel coordinates which correspond to the defect area in the source image. The second set, called C, corresponds to the background area. The third set is unlabeled and is considered non-interesting for the problem. This region relaxes the problem at the boundaries of the defect and suppresses other defects that might interfere with learning.

As usual in machine vision problems, evaluating the end result is not a trivial matter. In this case this post-processed image is compared with the user supplied label image (equation (50)) and the resulting value is used as a fitness value for the optimization.

Although the environment for this system is stable in the sense that defects are always oriented to the running direction of the machine, two Gabor filters are used by the system in order to attain a modicum of invariance for handling minor shifts in the orientation and curves of the defect:

$$\begin{aligned} G_f(\alpha, F) &= G_{15,15} * \max \quad [(\alpha_1 Gb(\alpha_2, ...\alpha_6) * F), \\ & \qquad (\alpha_7 Gb(\alpha_8, ...\alpha_{12}) * F)], \end{aligned} \tag{49}$$

where α_i are the parameters for the filters, F is the source image and $G_{15,15}$ represents the Gaussian kernel with a width and height of 15 pixels and a standard deviation of $\sigma = 3$. This smoothing is used in the training phase in order to remove spurious and noisy responses, similar to the approach in [KP02].

In order to complete modelling and to provide a fitness function to be optimized the resulting image must be compared to the ground truth in the label image. Various viewpoints for this exists: The simplest case would be to compare some norm between the label images and filtered images, but this penalises the algorithm for over/undershooting in target and background areas, which is undesirable. Why to penalize when the algorithm is making a stronger separation? The second viewpoint is similar and considers label image just as a ground truth and utilizes measures such as in Section 4.3.4. However directly using either of those measurements as an optimization goal will (empirically) result in an inferior filter. CR can be fooled by a single point of extremely high intensity in the defect area (or respectively low intensity in the background), which is a possible result of using weighted filters. For optimization purposes CNR is a better measure, but it penalizes high separation of defect areas due to increased standard deviation of the picture; the standard deviations in equation (47) also change when the ratio of defect-to-background changes. If not suited for direct optimization, these measures, however, are efficient in evaluating the resulting filters for the verification of the results obtained by optimization.

In order to give a numerical value for a comparison between the produce filtering result (S) and label image (L), it is necessary to consider the noise of the image and the separation, which is realized with the following equation:

$$\mathrm{sim}(S, L) = a\sigma(D) + b\sigma(C) + c\sqrt{|\mu(D) - \mu(C)|}, \tag{50}$$

where μ and σ denote, respectively, the mean and standard deviation of the pixel-values of the filtered image in a given pixel-coordinate set. The set C is a set of pixels of filtered image $G_f(\alpha, S)$ that correspond to the area defined as background in label image L and likewise for the defect area D. The weight factors a, b, and c determine the preferences of the system and they are empirically determined for values $a = 1$, $b = 2$, and $c = -3$. Thus, in order to minimize noise in the filtered image and maximize the separation of defect and the background sim must be minimized.

Problems of using CR and CNR for the purpose of optimizing filter performance are solved by equation (50). This formulation does not explicitly measure edges of defect as noise and is invariant to relative sizes of objects optimizing equation. The defect detection system is trained over a set of training pairs in order to achieve generality. The final fitness value is then counted as the average performance in each training case:

$$f(\alpha) = \frac{1}{N}\sum_{k=1}^{N} \mathrm{sim}\left(G_f(\alpha, S_k), L_k\right), \tag{51}$$

where S_k and L_k are k^{th} training and label images in a set containing N images. This formulation gives rise to a minimization problem that can be solved with general optimization algorithms. A successful minimization of this problem yields a real-valued filter that emphasises the defect in such manner that it can be segmented by thresholding. The several thresholding methods were compared for this purpose in the included article [PII] according to [SS04] including Otsu-, Kittler-Illingworth- and Bernsen methods. However, for this particular problem with a well defined estimate of background, a simple method has been developed[2]: As the histogram of the background seems to empirically follow a rather normal distribution, the threshold value t can be set according to

$$t = \mu(F) + K\sigma(F) + \epsilon,$$

where $\mu(F)$ is the average and $\sigma(F)$ the standard deviation of the image. Parameter K deals with sensitivity of the method: low K will yield lower thresholds. The value ϵ provides small tolerance for extreme cases such as bad contrast of the imaging device. Although usually considered a hindrance, this method has the benefit of having adjustable sensitivity (important in industrial applications) with easily understood settings, while being somewhat illumination invariant.

[2] Such an obvious solution is most likely used elsewhere, but is not interesting enough to warrant a name.

5 RESEARCH RESULTS AND INCLUDED ARTICLE SUMMARY

5.1 Research Framework

The study conducted in this thesis is constructive and experimental in nature. As per its nature, the field of meta-heuristics is not amenable to highly theoretical work when approaching the algorithms of the highest level of performance and/or complexity. Also, due to the nature of the methods investigated, the evaluation of results remains nontrivial. Many statistical tests would require assumptions on distributions of various algorithms' end results which may or may not be satisfied by the complex dynamic processes employed. Thus, in the context of the included papers, minimal assumptions are usually made and the solutions are mainly compared with two criteria: average end result of the optimization and standard deviation of sufficient amount of runs. Also, in order to circumvent the difficult problem of estimating the rate of convergence[1], the articles included follow the convention of visualizing results as average convergence graphs.

The results are supplied with comparative results of other meta-heuristics. Naturally, the main benchmark for DE based algorithms is the performance of the plain DE, but to give the reader the proper scale, classical meta-heuristics are also used. Besides the basic research of determining algorithm efficiencies in test problems the work consist also of practical application of optimized machine vision. As is common in machine vision communities [Jah00], there is no ultimate way of measuring performance of a vision system. As arbitrary but often used choice, CNR and CR measurements (see Section 4.3.4) are used in article [PII] to give a numerical validation to what can be seen in the resulting images.

The test problems employed in the papers are a machine vision application, a motor design problem, and some common test functions. The full set of the test functions with definitions is included in Table 1.

[1] For example, in the usual case we cannot measure the steps required to reach the optimum in case even a single run fails to do so.

Function	n	Equation	Domain
Ackley	20	$-20+e+\exp\left(-\frac{0.2}{n}\sqrt{\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi\cdot x_i)x_i\right)$	$[-1,1]^n$
Alpine	50	$\sum_{i=1}^{n}\lvert x_i\sin x_i+0.1x_i\rvert$	$[-10,10]^n$
Camelback	2	$4x_1^2-2.1x_1^2+\frac{x_1^6}{3}+x_1x_2-4x_2^2+4x_2^4$	
DeJong	10	$\lvert\lvert x\rvert\rvert^2$	$[-5.12,5.12]^n$
DropWave	50	$-\frac{1+\cos\left(12\sqrt{\lvert\lvert x\rvert\rvert^2}\right)}{\frac{1}{2}\lvert\lvert x\rvert\rvert^2+2}$	$[-5.12,5.12]^n$
Easom	2	$\cos x_1\cos x_2\exp\left(-(x_1-\pi)^2-(x_2-\pi)^2\right)$	$[-100,100]^n$
Griewangk	30	$\frac{\lvert\lvert x\rvert\rvert^2}{4000}-\prod_{i=0}^{n}\cos\frac{x_i}{\sqrt{i}}+1$	$[-600,600]^n$
Rotated Griewangk	30	$\frac{\lvert\lvert y\rvert\rvert^2}{4000}-\prod_{i=0}^{n}\cos\frac{y_i}{\sqrt{i}}+1$, where $y=Rx$	$[-600,600]^n$
Michalewicz	30	$-\sum_{i=1}^{n}\sin x_i\left(\sin\left(\frac{i\cdot x_i^2}{\pi}\right)\right)$	$[0,\pi]^n$

TABLE 1 Test cases used in this work

Function	n	Equation	Domain
Rotated Michalewicz	30	$-\sum_{i=1}^{n} \sin y_i \left(\sin \left(\frac{i \cdot y_i^2}{\pi} \right) \right)$, where $y = Rx$	$[0, \pi]^n$
Pathological	50	$\sum_{i=1}^{n-1} \left(0.5 + \frac{\sin^2\left(\sqrt{100x_i^2 + x_{i+1}^2} - 0.5\right)}{1 + 0.001 * \left(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2\right)^2} \right)$	$[-100, 100]^n$
Rosenbrock valley	30	$\sum_{i=0}^{n-1} \left((x_{n+1} - x_i^2)^2 + (1 - x)^2 \right)$	$[-2.048, 2.048]^n$
Rastrigin	30	$10n + \sum_{i=0}^{n} \left(x_i^2 - 10\cos(2\pi x_i) \right)$	$[-5.12, 5.12]^n$
Rotated Rastrigin	30	$10n + \sum_{i=0}^{n} \left(y_i^2 - 10\cos(2\pi y_i) \right)$, where $y = Rx$	$[-5.12, 5.12]^n$
Schwefel	30	$\sum_{i=1}^{n} x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-500, 500]^n$
Rotated Schwefel	30	$\sum_{i=1}^{n} y_i \sin\left(\sqrt{\lvert y_i \rvert}\right)$, where $y = Rx$	$[-500, 500]^n$
Sum of powers	50	$\sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	$[-1, 1]^n$

TABLE 1 (continued)

Function	n	**Equation**	**Domain**
Tirronen	50	$3\exp\left(-\frac{\|\|x\|\|^2}{10n}\right) - 10\exp\left(-8\|\|x\|\|^2\right)$ $+\frac{2.5}{n}\sum_{i=1}^{n}\cos\left(5x_i(1+i \mod 2)\right)$	$[-10,5]^n$
Whitely	50	$\sum_{i=1}^{n}\sum_{j=1}^{n}\left(\frac{y_{i,j}^2}{4000} - \cos\left(y_{i,j}\right) + 1\right)$, where $y_{i,j} = \left(100(x_j - x_i)^2 + (1-x_i)^2\right)^2$	$[-100,100]^n$
Zakharov	50	$\|\|x\|\|^2 + \left(\sum_{i=1}^{n}\frac{ix_1}{2}\right)^2 + \left(\sum_{i=1}^{n}\frac{ix_1}{2}x_i\right)^4$	$[-5,10]^n$
Filter Design	12	See Formulae (49)-(51)	

TABLE 1 (continued)

Generating new test cases by introducing rotation and/or by increasing dimensionality of the functions generates problems that are usually much harder to solve than the original ones and are thus excellent material for algorithmic testing. Rotation is achieved by generating a random orthonormal matrix that is used to map points in the fitness domain into the rotated one (denoted as R in Table 1). Rotation turns separable functions into non-separable ones and usually destroys correlation of fitness values with the axes of coordinate system. Some algorithms rely upon searching along the axes or diagonals of coordinate systems, so this will present a harder and possibly more realistic test case.

Dimensionality increases, as this thesis advances, from 10 up to 50 dimensions in order to generate "hard enough" problems for later algorithms.

5.2 List of included articles

The included articles fall naturally to two distinct groups and they are not chronologically ordered, in order to form a cohesive grouping by the content of the publications. The application oriented articles ([PI]-[PIV]) are introduced first. These articles are also essential as a motivator for the rest of the publications ([PV]-[PIX]) that delve into basic research of memetic algorithms.

The basic research further feeds into refinements to application oriented publications in form of accumulated knowledge of enhanced parametrization and coordination schemes, which is detailed in Sections 5.3.3–5.3.4.

The articles start by outlining the vision problem and the memetic approach and offering numerical results validating the use of the memetic framework in the context of the problem. The algorithms introduced in this section are Memetic Differential Evolution (MDE), Enhanced Memetic Differential Evolution (EMDE) and Super-fit Memetic Differential Evolution (SFMDE). Each is surveyed in detail for this context, offering performance statistics and conjectures of their behavior. This part is concluded with an article consisting of a survey of the frameworks introduced here and their relative performance. Also this article overviews the various coordination mechanics and fitness diversity based adaptation schemes.

The theoretical group of publications begins with a study of hybridization in a general context studying the possible benefits from hybridizing DE with local search algorithms. The rest of the part deals with analyzing the algorithms and the effect of their adaptation schemes progressing from deterministic coordination methods to stochastic ones. Stochastic coordination is considered in depth at [PVII] where the effects of different selections of probability distributions for coordination are studied and compared with the original static solutions presented before. Also, in this part an algorithm called Fast-Adaptive-Memetic-Algorithm (FAMA) is studied to analyze generalization of stochastic scheme outside DE frameworks. FAMA is further studied in the last paper of the set analyzing the uniform stochastic scheme for adaptation in MAs.

5.3 Summary of results

5.3.1 A Memetic Differential Evolution in Filter Design for Defect Detection in Paper Production

This article proposes a Memetic Differential Evolution (MDE) for designing digital filters which aim at detecting defects of the paper produced during an industrial process. This is the initiating article of this research track, combining basic level machine vision studies with first attempts to optimize them with a global optimizer. The application problem of finding the defects is previously described in Section 4.2.1. The focus of the problem dealt in this paper is about detection of streak and wrinkle type defects that have traditionally posed difficulties within industry.

Optimization problem is formulated as the optimization of a set of Gabor filters as given in Section 4.2.1. The algorithm used to optimize the filter, MDE, is detailed previously in Section 3.2.1.

The numerical results visualized in the average performance curve in Figure 17 show that the DE framework is efficient for the class of problems under study and employment of exploitative local searchers is helpful in supporting the DE explorative mechanism in avoiding stagnation and thus detecting solutions having a high performance. The second picture in the figure illustrates the operating principle of static fitness diversity coordination according to ν measure in action.

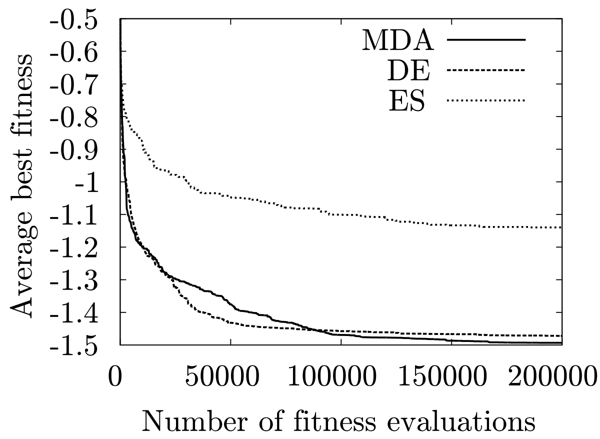
In context of the thesis this article lays out the starting point by introducing the problem formulation of the machine vision experiment as well as in general sketching the outlines for various memetic DE schemes utilizing a measure of fitness diversity in coordination of the algorithm.

5.3.2 An Enhanced Memetic Differential Evolution in Filter Design for Defect Detection in Paper Production

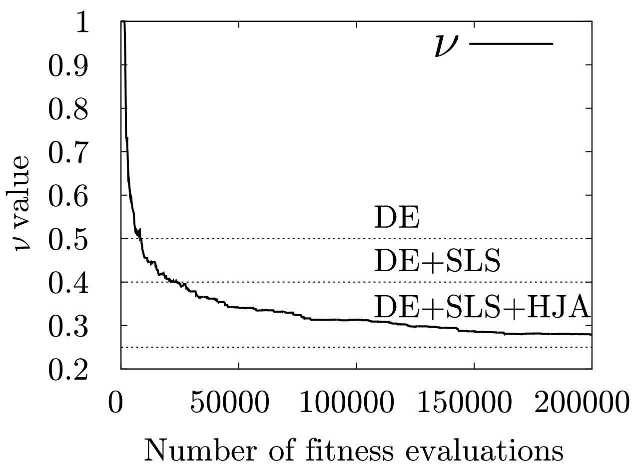
This article proposes an Enhanced Memetic Differential Evolution (EMDE) for a similar purpose of designing digital filters which aim at detecting defects of the paper produced during an industrial process. The filter formulation stays constant from the previous paper and more consideration is given to the actual optimization algorithm, EMDE, described in Section 3.2.2.

This article includes numerical tests for local searchers in various states of optimization in the context of the filter design application. These tests indicate clearly the correct phases and when to apply which searcher.

The numerical results show that the EMDE performs well for the problem under study and leads to a design of an efficient filter. The comparison with three standard meta-heuristics in Figure 18a show the effectiveness of the EMDE in terms of convergence speed, stagnation prevention, and capability in detecting solutions having a high performance. A comparison of the algorithms is presented in Figure 18. In comparison to DE the previously introduced algorithm



(a) Performance of MDE



(b) Trend of ν

FIGURE 17 Behaviour of MDE

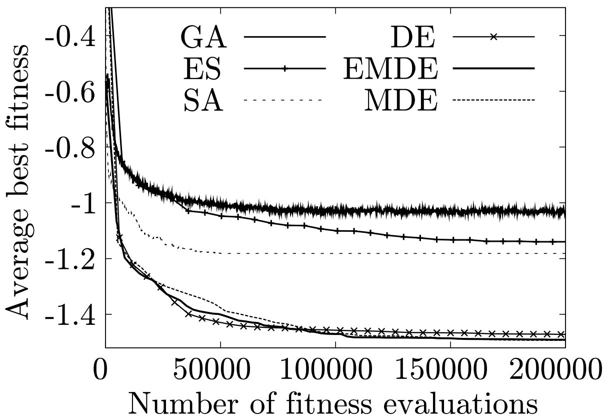
MDE is able to reach a better filter design but is slightly slower in reaching a sufficient performance. EMDE outperforms MDE in terms of convergence speed, being almost as quick as an ordinary DE yet converging to final values well equivalent to those of MDE. The convergence properties of EMDE can be compared to other algorithms through the analysis of average ν-measured fitness diversity as illustrated in Figure 18b. For both DE and MDE the fitness diversity remains high during the entire run of the algorithm indicating that the search might be stagnating. EMDE seems more promising in this regard as the fitness diversity is decreasing systemically to a small value most likely due to the improved control scheme. The interpretation given in this paper is that this is mainly due to the stochastic control scheme and to a lesser extent to the shorter runs of the local searchers. This result can be considered as important as the final performance in respect to any future work. As a demonstration the behaviour of EMDE, a plot of search frames, and the corresponding fitness values are provided in Figure 19. This plot demonstrates the efficiency of local searchers in the middle of the optimization run. The x-axis shows which algorithms have been employed at each step in order to demonstrate their effect on the fitness (y-axis). As can be seen from the plot, improvements of the fitness tend to take place in the beginning of the search frames as a new search perspective is applied.

The main contributions towards this thesis come from the introduction of probabilistic coordination scheme that provides for more efficient and flexible coordination. We can assume that the model of the filter is nearly exhausted as far as improvements are concerned and can thus consider the work as 'done'. The work carried out in this paper lead to further understanding of coordination and the probabilistic scheme is re-utilized in the subsequent papers.

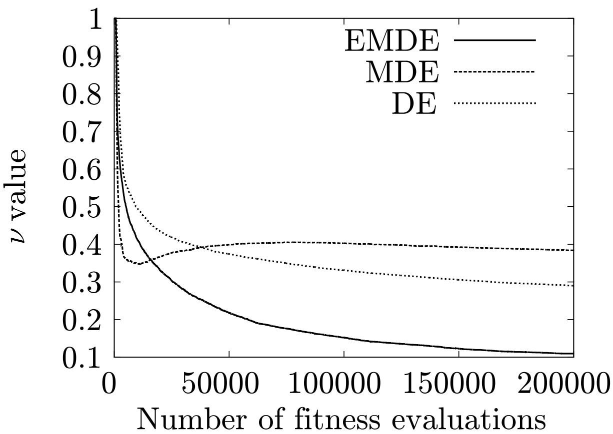
5.3.3 Super-fit Control Adaptation in Memetic Differential Evolution Frameworks

This paper presents the Super-Fit Memetic Differential Evolution (SFMDE). This algorithm employs a DE framework, similar to those introduced in the earlier papers, hybridized with three meta-heuristics each having different roles and features. The Particle Swarm Optimization assists DE in the beginning of the optimization process by helping to generate a super-fit individual. The other two meta-heuristics are local searchers adaptively coordinated by means of an index measuring the quality of the super-fit individual with respect to the rest of the population and a probabilistic scheme resorting to generalized beta distribution. These two local searchers are the Nelder-Mead and Rosenbrock algorithm. This paper introduces the asymmetric activation functions for local searchers.

This method is applied to both real-world problems including the aforementioned filter design problem as well as to electric motor controller optimization and to test problems. Numerical results show that the DE framework is efficient for the class of problems under study and employment of exploitative local searchers is helpful in supporting the DE explorative mechanism in avoiding stagnation and thus detecting solutions having a high performance.



(a) Performance of EMDE



(b) Trend of ν

FIGURE 18 Behaviour of EMDE

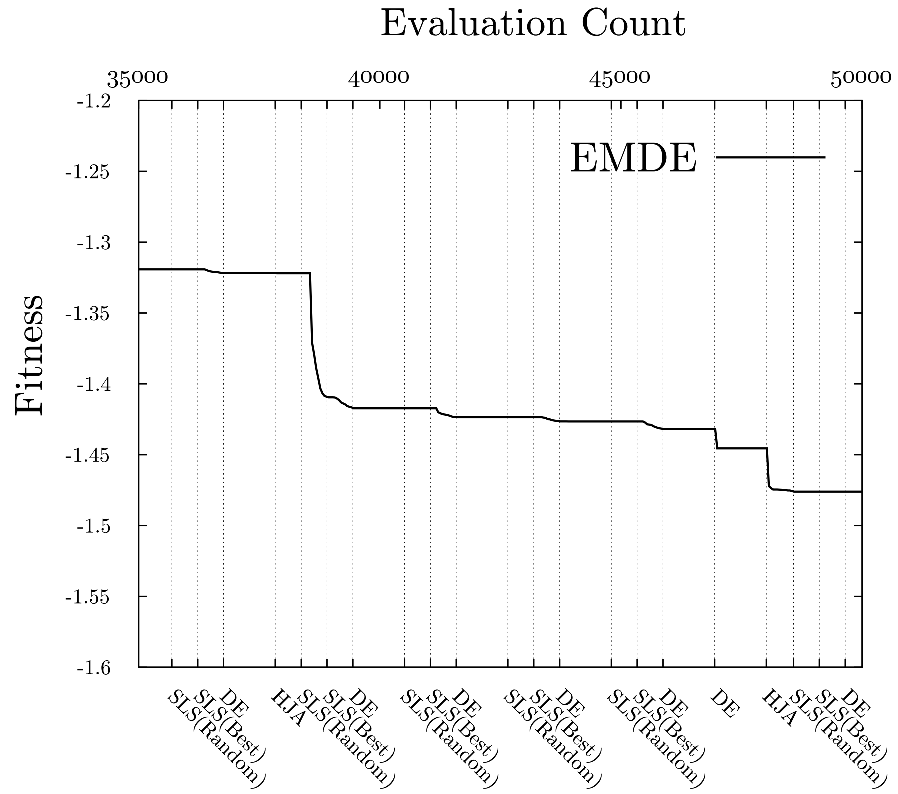


FIGURE 19 Demonstration of search behaviour of EMDE. X-axis shows the number of fitness evaluations and which element of the algorithm was employed at each step.

The research includes an analysis of local searcher performance in various stages of optimization. The performance of various optimizers changes during the progress of the search. These results justify the adaptation scheme. Also, convergence plots in Figure 20 show the fact that SFMDE outperforms algorithms it is tested against in several test functions. The same result can be observed in the filter design application (Figure 21), in which SFMDE also outperforms the other algorithms. (This result is included in addition to those presented in the paper.)

The contribution of this paper is improved local searchers as well as an asymmetric co-ordination scheme. Relevant to this thesis, the paper also extends the study of suitability of local search.

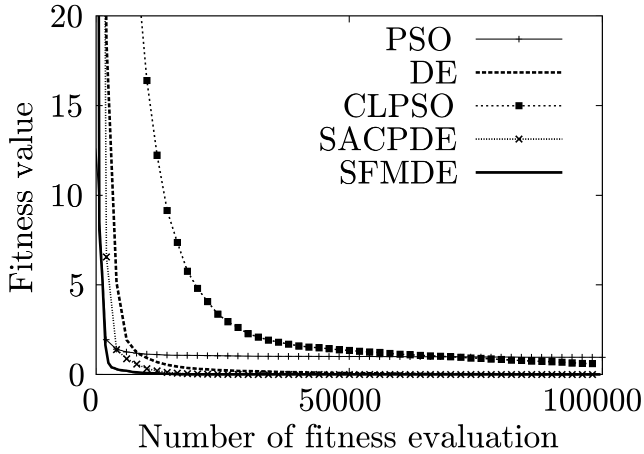
5.3.4 Memetic Differential Evolution Frameworks in Filter Design for Defect Detection in Paper Production

This publication studies and analyzes Memetic Differential Evolution (MDE) frameworks for the aforementioned filter design problem in paper production. This publication is the convergence point of practical studies, studying and coalescing the previous articles and further detailing the adaptation process and presenting larger comparative study of the methods previously introduced. This work can be considered as an overview of those algorithms.

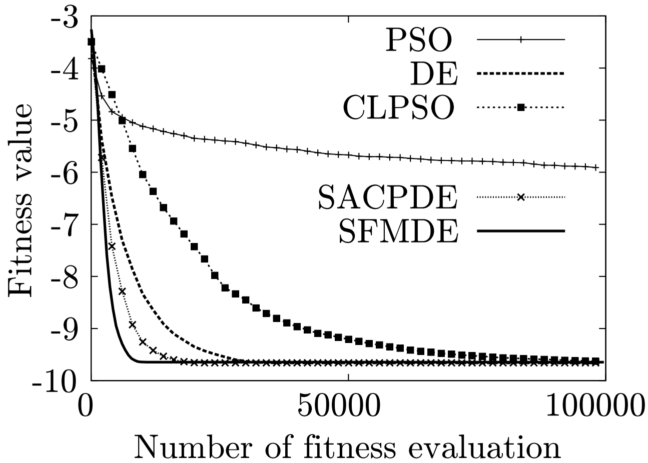
The introduced algorithms are compared to each other in the context of the filter design problem. The binomial explorative features of the DE framework in contraposition to the exploitative features of the local searcher are analyzed in detail in light of the stagnation prevention problem, typical for DE. Much emphasis is given to the various adaptation systems and to their applicability to this image processing problem.

The performance of the algorithms in vision problem is plotted in Figure 22. As can be seen from the results all memetic algorithms tested outperform the classical methods. As can be seen from the table, MDE and EMDE find almost exactly the same solution with EMDE attaining slightly higher precision, while the solutions for the rest of the algorithms are considerably different, though the fitness is nearly the same. It is apparent that SFMDE is faster than other algorithms, possibly due to more efficient local searchers. Filtered images corresponding to the best results of the various algorithms are displayed in Figures 23 (images within training set) and 24 (validation images).

The results indicate progress from unusable filters derived with SA and ES to nearly usable by DE into practically applicable filters derived by various *MDEs. This work concludes the filter design problem. The model used is likely to be exhausted by now and not capable of providing added improvements regardless of the optimization method applied.

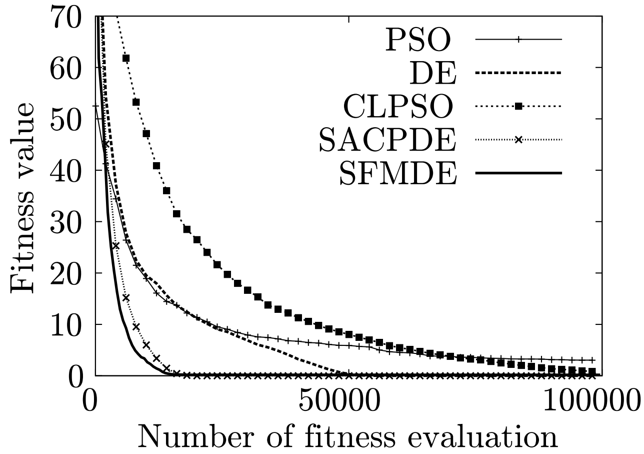


(a) Griewangk function

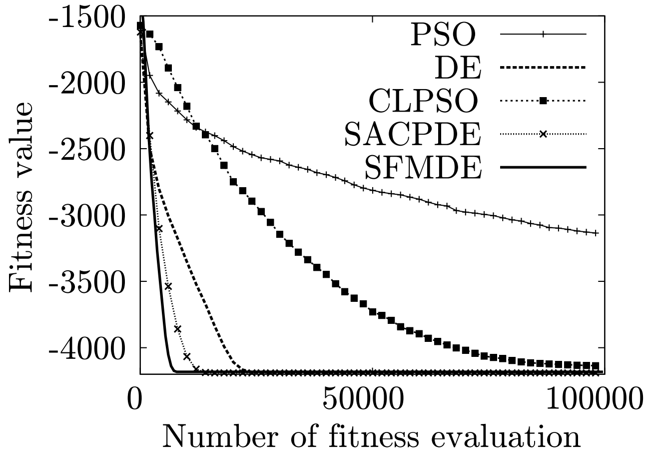


(b) Michalewicz function

FIGURE 20 Performance of SFMDE over the set of test problems



(a) Rastrigin function



(b) Schwefel function

FIGURE 20 (continued)

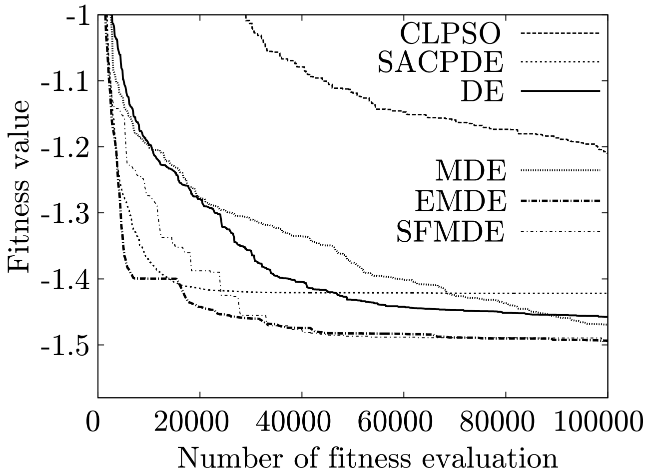


FIGURE 21 Performance of SFMDE against modern meta-heuristics in the vision problem

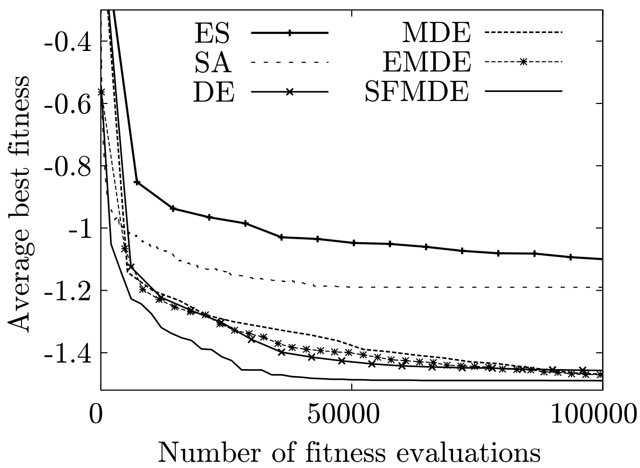
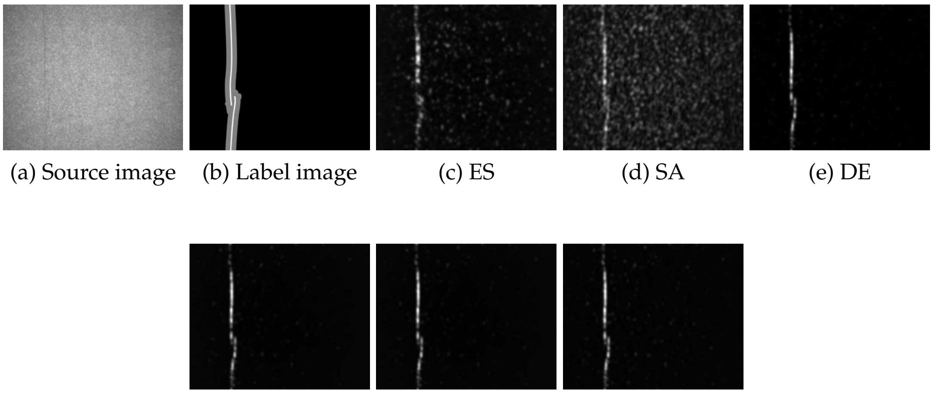


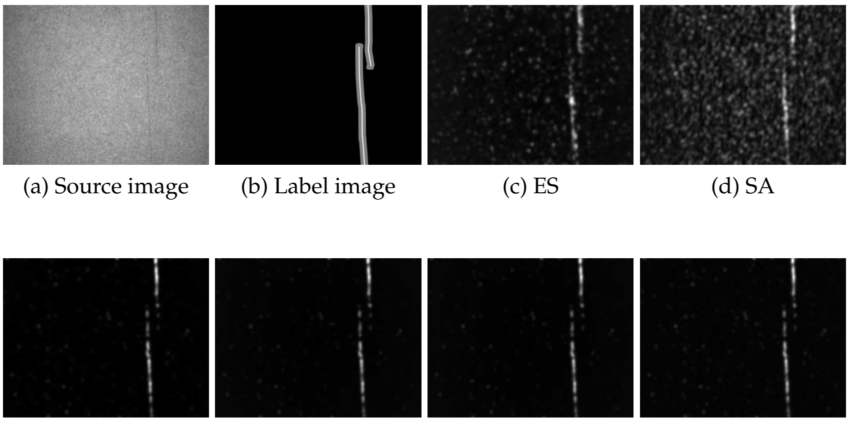
FIGURE 22 Algorithmic performance



(a) Source image (b) Label image (c) ES (d) SA (e) DE

(f) MDE (g) EMDE (h) SFMDE

FIGURE 23 Best results on a training set image



(a) Source image (b) Label image (c) ES (d) SA

(e) DE (f) MDE (g) EMDE (h) SFMDE

FIGURE 24 Best results on a validation image

5.3.5 On Memetic Differential evolution Frameworks: A Study of Advantages and Limitations in Hybridization

This paper aims to study the benefits and limitations in the hybridization of the DE with local search algorithms. It is of introductory quality by the way that it allows to conjecture about the efficiency of MDE with respect to DE and in a sense helps to justify the approach taken in the thesis.

In order to perform this study, the performance of all three memetic algorithms introduced in this thesis are compared against a plain DE with tuned parameter setting. The algorithms have been tested with common test functions (described in Section 5.1) which hopefully provide a wide range of landscapes.

The comparative analysis has been performed on a set of various test functions. Numerical results show that the MAs without any extensive parameter tuning are still competitive with the finely tuned plain DE.

The example performances of various algorithms are shown in Figure 25. It should also be stressed that empirical tuning of DE is not usually possible in real world problems. From these figures it can be deduced that neither DE nor any of the various *MDE schemes strictly dominate the other. In most cases algorithms have somewhat similar performances, which is a success for those *MDE schemes since they are competing against the finely tuned DE.

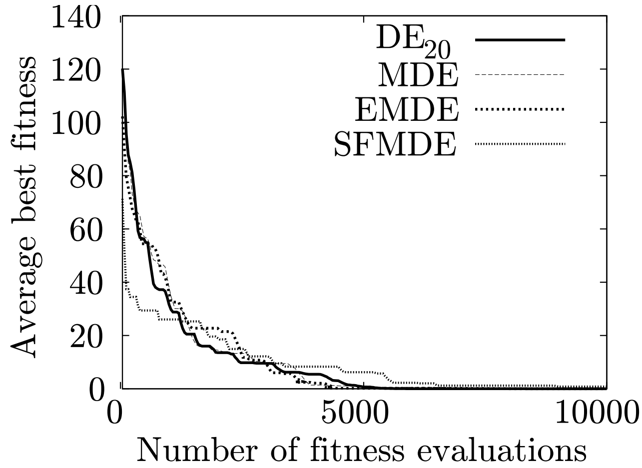
In simple cases, such as De Jong hyperbola, memetic algorithms win simply by starting the local search in the basin of attraction. In cases such as Ackley, which contains a strong global basin of attraction and numerous local minimae, it can be conjectured that local searchers mislead the search by focusing on the small details of the landscape. However, such focusing may be beneficial, since, for example in the case of the Rastrigin function, the DE is able, with a suitable selection of F, to simply move from one equidistant minimae to another. When a local search finds these minimae, the problem becomes in a sense unimodal as higher frequency changes in the fitness become irrelevant.

5.3.6 Fitness Diversity Adaptation in Multimeme Algorithms: A Comparative Study

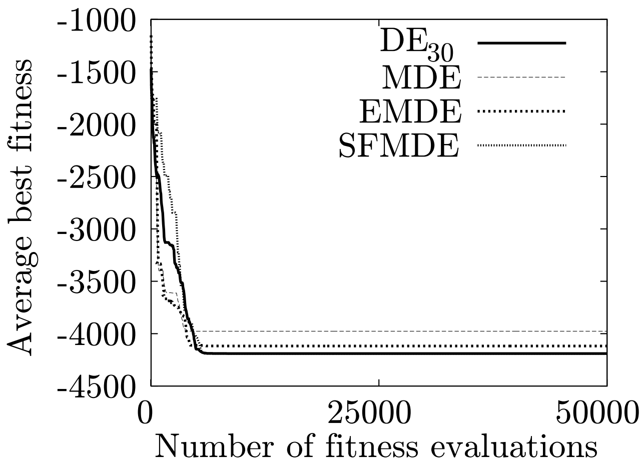
This publication provides the real starting point for Part II of the papers by tackling the central topic of fitness diversity by comparing three different fitness diversity adaptations introduced in Section 3.1 in Multimeme Algorithms. These diversity indexes have been integrated within a MmA in literature called Fast Adaptive Memetic Algorithm (FAMA) [CCN^{+}07a].

FAMA is a modified real-encoded genetic algorithm with a variable population size, uniform mutation, and local search. Population size, mutation rate, and activation of local search are controlled by a fitness diversity measurement. Different measures have been experimented on in the included article [PIX].

Empirically FAMA has a very fast convergence as its name indicates, most likely due to a high mutation and changing population size which culls bad results quickly from the population. Also, local search is apt at both exploiting

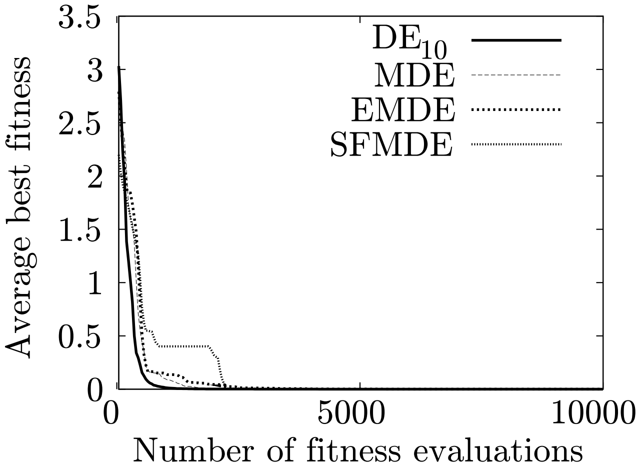


(a) *MDE Performance in Rastrigin function

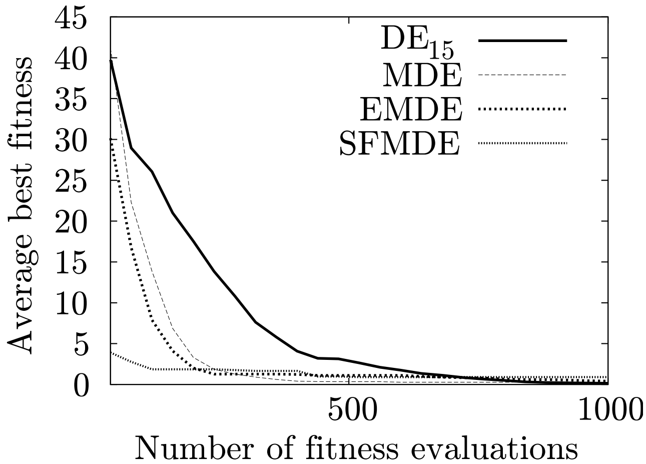


(b) *MDE Performance in Schwefel function

FIGURE 25 Behaviour of *MDEs on various test functions



(a) *MDE Performance in Ackley function



(b) *MDE Performance in Dejong function

FIGURE 25 (continued)

the found promising solutions as well as restoring the diversity that is lost by decreasing the size of a population.

Numerical results in this paper show that it is not possible to establish a superiority of examined adaptive schemes over the others, and the choice of a proper adaptation must be made by considering the features of the problem under study. More specifically, ψ measure outperforms the others in the presence of plateaus or limited range of variability in fitness values and in general appears as most robust. Adaptation by ν is more proper for landscapes having distant and strong basins of attraction. On the other hand ξ, despite of its mediocre average performance can occasionally lead to excellent results. Example plots of performance[2] can be seen in Figure 26, where different algorithms are compared in three distinct cases.

In the framework of the collected work this article provides for background on coordination schemes and has a natural continuation in the "Natura Non Facit Saltus" paper presented next, which further investigates the fitness diversity based coordination schemes.

5.3.7 The "Natura Non Facit Saltus" Principle in Memetic Computing

The nature doesn't jump. This paper can be seen as a loose continuation to the previous one. It continues the study of fitness diversity based adaptation and coordination of local search. For this reason it studies the employment of continuous probability functions instead of step functions for adaptive coordination of the local search in fitness diversity based MAs. This idea was first presented with the EMDE algorithm in article [PII].

Two probability distributions are considered in this study: the beta and exponential distributions. These probability distributions have been tested within two memetic frameworks, MDE and FAMA (see Section 5.3.6). To compare the systems, the activation functions plotted in Figures 27 and 28 were tested with test problems. In this experiment dimensions similar to those in static thresholds were used. However, since control parameter values are not uniformly distributed, it does not have the same number of local search activations.

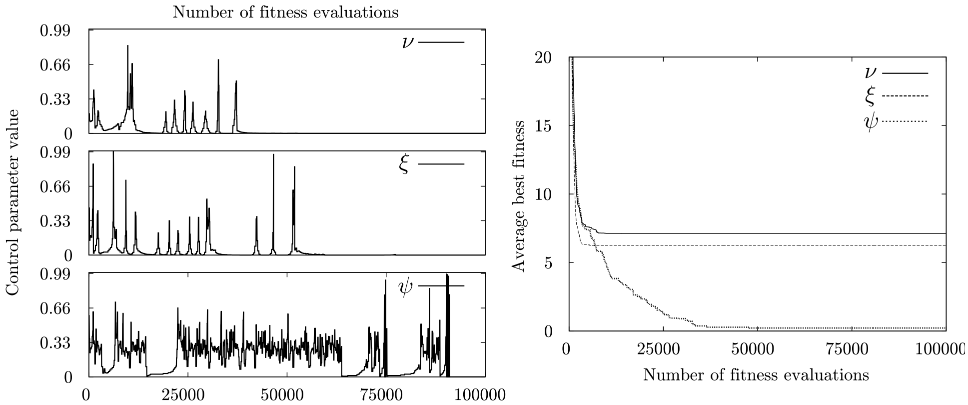
Numerical results in the paper show that employment of the probability distributions can be beneficial and improve the performance of the original Memetic Algorithms on a set of test functions without varying the balance between the evolutionary and local search components.

Also, a factor that needs to be emphasized is the benefit of using softer logic in algorithm design, making it easier and somewhat more flexible to design control schemes.

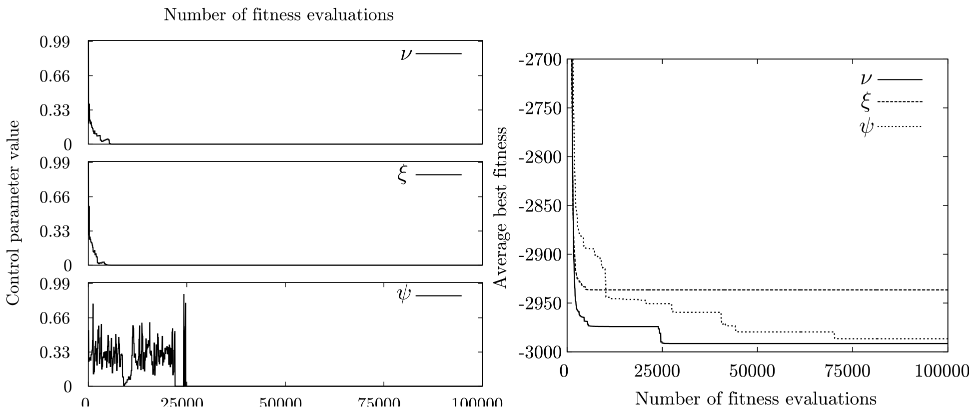
5.3.8 Integrating Fitness Diversity Self-Adaptation in Differential Evolution

This included chapter is the final work done on the topic of fitness diversity. Unlike the previous works in this thesis, it does not deal with a memetic algo-

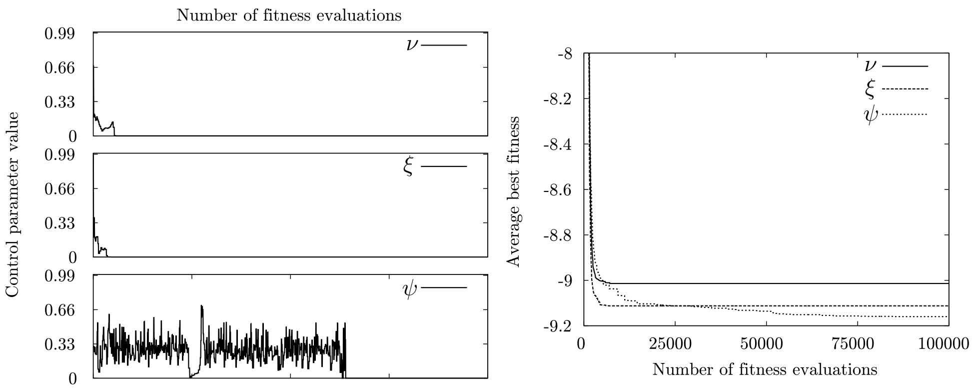
[2] See the original paper for the rest of the results



(a) Performance of different adaptations on Rastrigin function



(b) Performance of different adaptations on Schwefel function



(c) Performance of different adaptations on Michalewicz function

FIGURE 26 Behaviour of various adaptation schemes with FAMA algorithm

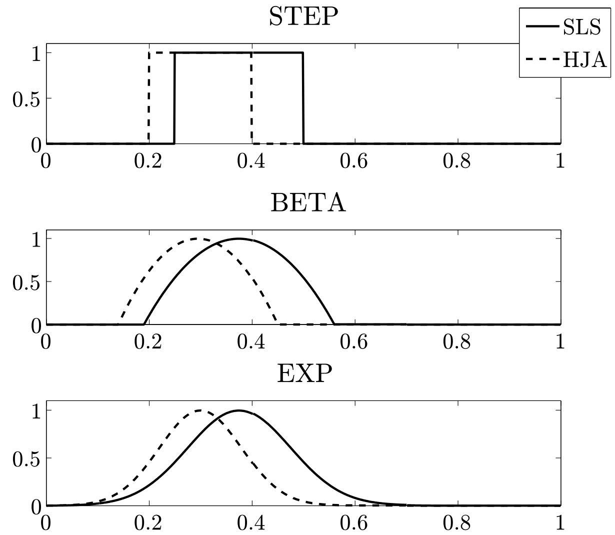


FIGURE 27 Activation functions for MDE

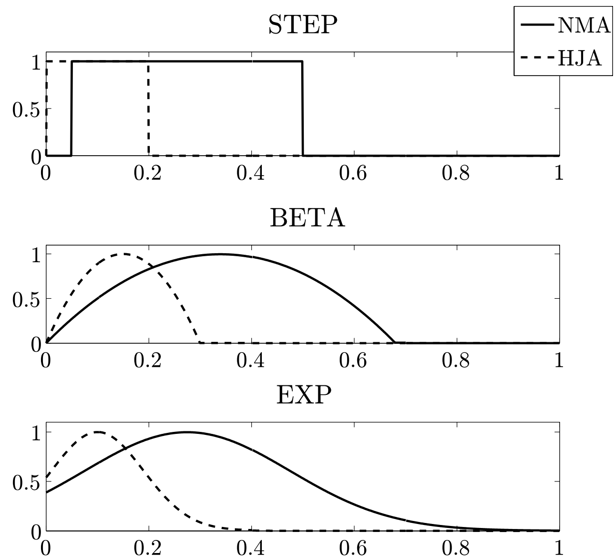


FIGURE 28 Activation functions for FAMA

rithm. Instead this work proposes the integration of fitness adaptation techniques within the parameter setting of DE. Parameters concerning explorative logic and crossover probability are encoded within each genotype and self-adaptively updated during the evolution by considering the fitness diversity of the entire population and ranking position of the solution under examination. This work holds it place in this thesis by providing further application and evidence of suitability of fitness diversity in the field of global optimization. Even though the algorithm proposed herein does not strictly fall in the category of MDE based algorithms it is of a close enough resemblance to them.

This work introduces an algorithm called Fitness Diversity Adaptive Differential Evolution (FDSADE). The algorithm applies ϕ-measured (See Section 3.1, equation (42)) fitness diversity in order to properly adapt F and Cr values to the problem at hand. The algorithm is based on the SACPDE model with two important changes. Firstly, the control parameter F_i for the i^{th} individual is updated according to the following schema:

$$F_i = \begin{cases} F_l + F_u u_1, & \text{if} \quad u_2 < K(1-\phi), \\ F_i, & \quad\quad \text{otherwise.} \end{cases} \tag{52}$$

Variables F_l and F_u are control parameters defining the upper and lower limits for F. A comparison of the performance has been conducted by considering a standard DE as well as modern DE based algorithms.

The Cr values for individuals are adapted with the same logic:

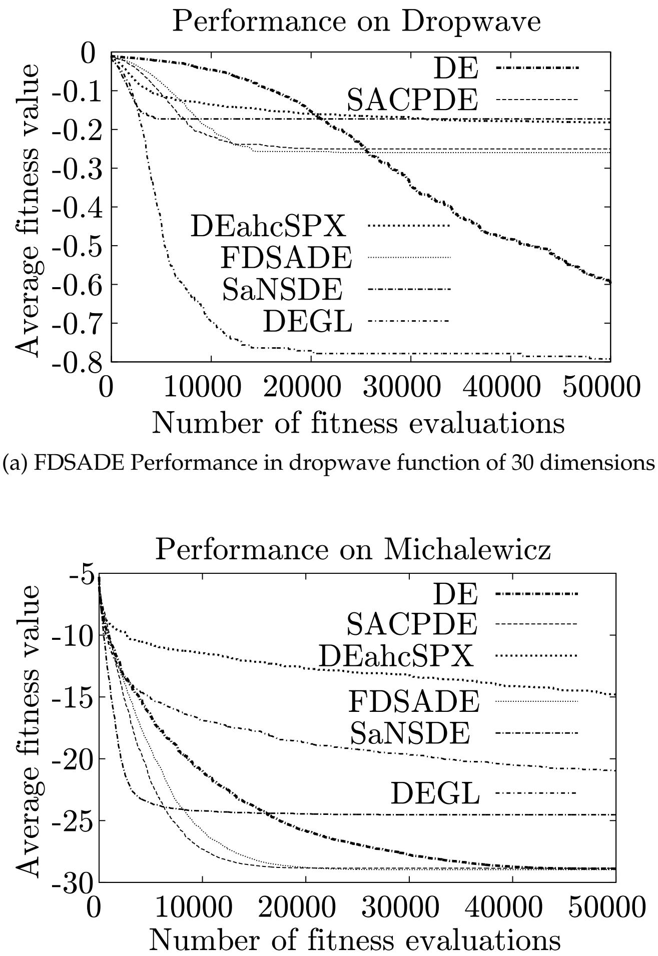
$$CR_i = \begin{cases} u_3, & \text{if} \quad u_4 < K(1-\phi) \\ CR_i, & \quad\quad \text{otherwise} \end{cases} \tag{53}$$

where K is the maximal update probability of the parameters. The algorithm works by the idea that in a high diversity condition ($\phi \approx 1$) one should try to exploit the current parameter setting while in a low diversity conditions the algorithm should try to find a new set of working parameters. This logic is similar to that expressed before in the context of local search activation with MDE style algorithms.

Numerical results achieved in this work (see Figures 29-30) show that the proposed approach seems promising for some fitness landscapes and still competitive with modern algorithms in other cases in both low (30) dimensional and relatively high (100) dimensional test cases. In most cases the proposed self-adaptation is helpful in terms of algorithmic performance.

5.3.9 A Fast Randomized Memetic Algorithm in Highly Multi-modal Problems

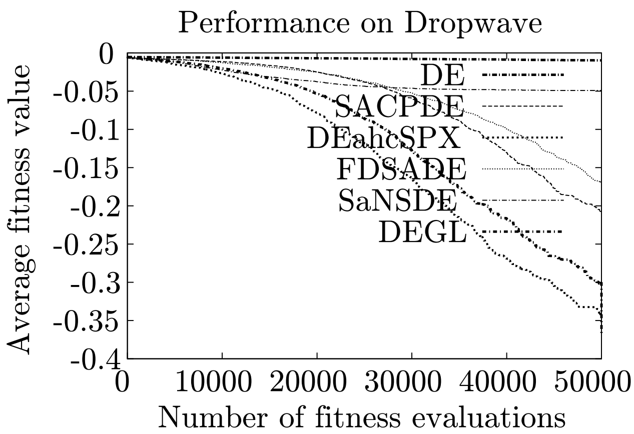
The last paper is a lighter work studying the ultimate end of stochasticity in fitness diversity based adaptation. Previous papers have gone from static coordination ([PI]) to stochastic one ([PII]) and as the last publication of the series this one studies the effects of applying a completely stochastic scheme without any diversity based or other kind of adaptation.



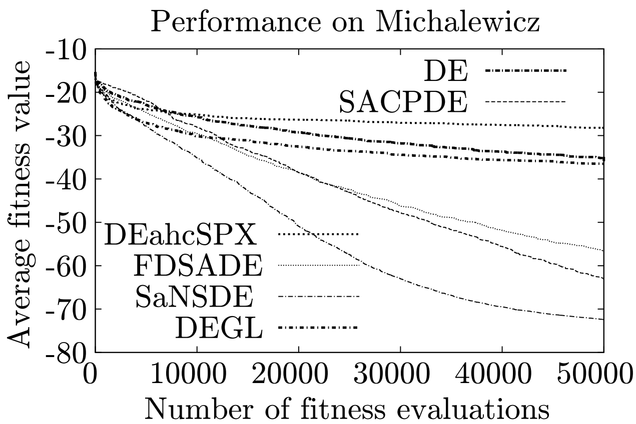
(a) FDSADE Performance in dropwave function of 30 dimensions

(b) FDSADE Performance in Michalewicz function of 30 dimensions

FIGURE 29 Behaviour of FDSADE on 30 dimensional test functions



(a) FDSADE Performance in dropwave function of 100 dimensions



(b) FDSADE Performance in Michalewicz function of 100 dimensions

FIGURE 30 Behaviour of FDSADE on 100 dimensional test functions

This paper proposes a Fast Randomized Memetic Algorithm (FRMA) for handling multi-modalities in fitness landscapes. The FRMA is derived from FAMA (introduced in [PV]), by removing the fitness diversity control scheme and replacing it with a pure random solution.

Numerical results show, for example, in the convergence plots included in Figure 31, that for a set of test functions the FRMA outperforms popular metaheuristics, often implemented for multi-modal problems. Comparison with adaptive memetic algorithms employing the same algorithmic components highlights the FRMA's high performance in terms of both convergence velocity and final solution detected. It can be conjectured that this is mostly due to the structure of FAMA and FRMA algorithms which incorporate changing population size and highly exploitative nature. When the control value fluctuates the algorithm discards a large part of the population, exploits the solutions that remain and yet again increases the population size in order to re-introduce diversity. It is possible that this scheme is non-optimal if the algorithm remains in one stage for too long, which can happen with a static control scheme utilized by various FAMA algorithms. Applying randomness in this process avoids this and gives additional empirical evidence towards the use of a stochastic control scheme.

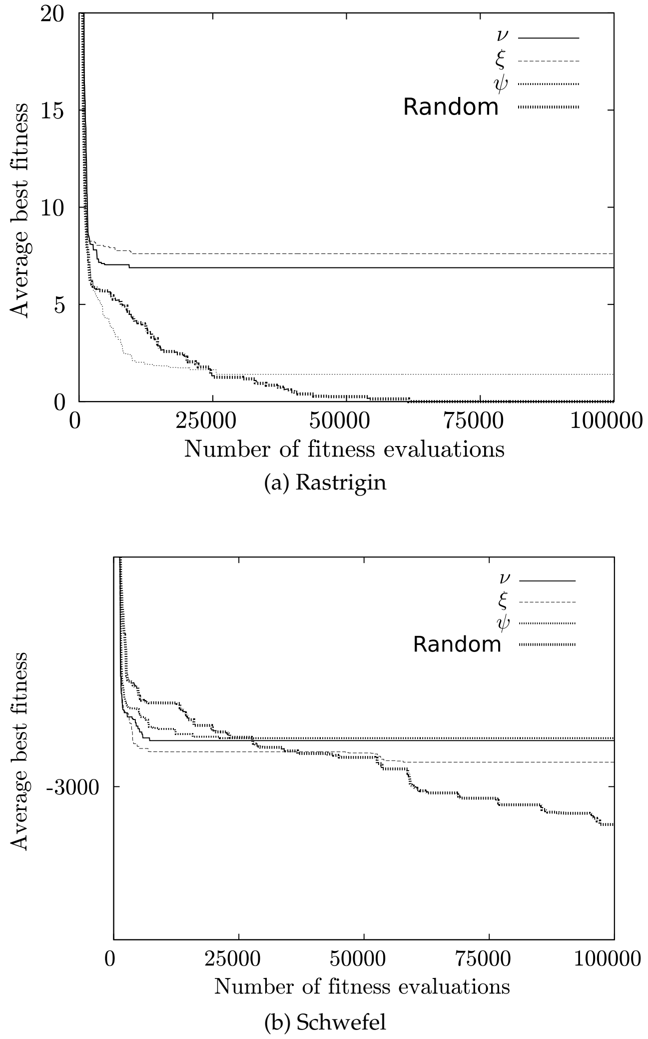
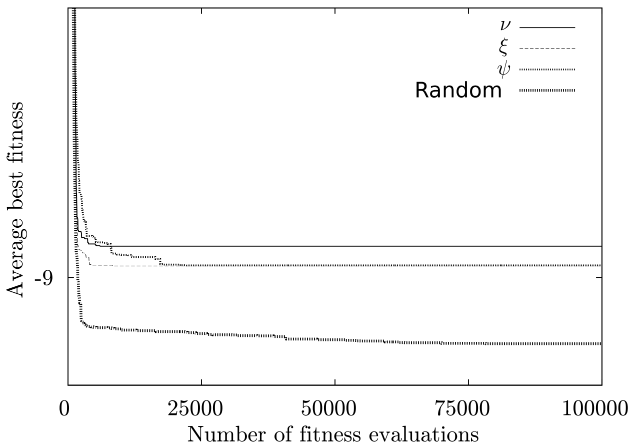


FIGURE 31 Comparison with $\xi-$, $\psi-$ and $\nu-$ FAMA



(a) Michalewicz

FIGURE 31 (continued)

6 CONCLUSIONS

In this work nine papers on subjects of memetic computation, adaptation and fitness diversity have been presented. The biggest contribution is the introduction of the memetic differential evolution -family of algorithms. In the included papers it has been verified as a good solution for various optimization problems. More importantly, its underlying ideas have been tested, and all the tests provide strong evidence for this style of algorithm design which hopefully will outlast the described methods. The machine vision problem described in this work has been mostly solved, but lacks rigorous testing within the industry. In the authors' view it provides the first steps of a future way of solving these problems – not by human engineering but by automatic processes.

Some general conjectures can be made in addition to those given in the articles about the performance of various algorithms introduced: DE is quite sensitive to parameter setting and performs in a non-optimal way when the dimensionality of the problems increases. Also, a strong dependency of parameter setting can be seen in the studies conducted. The parameter setting is an active problem area but it seems that the memetic approaches used alleviate both of these problems in a considerable degree.

The work will continue on the topic of optimization and focus more strictly on modern adaptive schemes and novel local searchers, replacing old methods used in these papers. The general philosophy derived from this work favours stochastic control schemes and randomness. This development will be critical when progressing to problems of high dimensions (more than 100 parameter problems) and possibly even in noisy cases. Possible future work will involve both problems of a large scale dimension and multi-objective adaptations of these algorithms.

YHTEENVETO (FINNISH SUMMARY)

Työ koostuu erilaisten memeettisten optimointimenetelmien soveltamisesta ja niiden suunnittelusta. Optimointimenetelmät, joita tässä työssä käytetään, koostuvat pääasiassa differentiaalievoluutiosta ja siitä edelleen kehitetyistä memeettisistä algoritmeista. Tällaiset menetelmät ovat saavuttaneet suosiota akateemisissa piireissä ja jalansijaa teollisuudessa. Ne ovat erityisen ajankohtaisia kasvavan laskentatehon ja hankalammiksi muuttuvien ongelmien piirissä.

Tämä työ käsittelee myös paperinvalmistukseen liittyvää vianilmaisuongelmaa. Vianilmaisu on haastava ongelma, jossa on erittäin tiukat reaaliaikavaatimukset. Tästä johtuen työssä ei voida soveltaa monimutkaisia ja laskennallisesti raskaita malleja, vaan sen sijaan ongelma on muotoiltu perinteisen suodinmallin kautta optimointitehtäväksi, joka ratkaistaan tehokkailla laskennallisesti älykkäillä menetelmillä, kuten memeettisillä algoritmeilla. Työ on tuottanut lupaavia tuloksia sekä yksittäisessä konenäköongelmassa että optimointimenetelmien kehityksessä.

ERRATA

- Gabor filter formula in papers ([PI, PIV]) contains an error. See Section 4.4, equation (49), for correction.
- Super-fit Control Adaptation, p. 6, "The procedure is repeated until a budget condition is not exceeded", "The procedure is repeated until a budget condition is exceeded".
- In article [PVII] the bounds for local searcher activation in text do not match ones in the picture. (Text is wrong). This is carried over from a article [PI] which also has this flaw.
- In [PIII], pseudo-code in Figure 6 lacks normalization for the function describing probability of LS activation. Refer to text in [PIII] for clarification.
- Paper [PII] contradicts itself by stating "can be used in real-time conditions without cost of extra hardware components." This is intended to mean extra hardware components *in addition to* specialized detection hardware described earlier.

REFERENCES

[Ack89] R. L. Ackoff. From data to wisdom. *Journal of Applied Systems Analysis*, 16(1):3–9, 1989.

[AK94] H. K. Aghajan and T. Kailath. *Method of edge detection in optical images using neural network classifier*. US Patent 5,311,600, 1994.

[BA94] K. Bowyer and S. Astley. *State of the Art in Digital Mammographic Image Analysis*. World Scientific, 1994.

[BGB+06] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10(6):646–657, 2006.

[BGK04] E. K. Burke, S. Gustafson, and G. Kendall. Diversity in genetic programming: an analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation*, 8(1):47–62, 2004.

[BHS91] T. Back, F. Hoffmeister, and H.P. Schwefel. A survey of evolution strategies. *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 2—9, 1991.

[BLE+04] A. Blum, H. Luedtke, U. Ellwanger, R. Schwabe, G. Rassner, and C. Garbe. Digital image analysis for diagnosis of cutaneous melanoma. Development of a highly effective computer algorithm based on analysis of 837 melanocytic lesions. *British Journal of Dermatology*, 151(5):1029–1038, 2004.

[Bov05] A. C. Bovik. *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. Academic Press, Inc. Orlando, FL, USA, 2005.

[Bra65] R Bracewell. *"Convolution" and "Two-Dimensional Convolution."*, pages 25–50, 243–244. McGraw-Hill, 1965.

[BSdJ+99] J. A. Belien, S. Somi, J. S. de Jong, P. J. van Diest, and J. P. Baak. Fully automated microvessel counting and hot spot selection by image processing of whole tumour sections in invasive breast cancer. *British Medical Journal*, 52(3):184, 1999.

[BZM06] J. Brest, V. Zumer, and M.S. Maucec. Self-adaptive differential evolution algorithm in constrained real-parameter optimization. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 215–222, 2006.

[CC06] Fang-Hsuan Cheng and Yu-Liang Chen. Real time multiple objects tracking and identification based on discrete wavelet transform. *Pattern Recognition*, 39(6):1126–1139, June 2006.

[CCK02] Y. R. Chen, K. Chao, and M. S. Kim. Machine vision technology for agricultural applications. *Computers and Electronics in Agriculture*, 36(2-3):173–191, 2002.

[CCN+07a] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumner. A fast adaptive memetic algorithm for on-line and off-line control design of pmsm drives. *IEEE Transactions On Systems, Man and Cybernetics - Part B, special issue on Memetic Algorithms*, 37:28–41, 2007.

[CCN+07b] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumner. A fast adaptive memetic algorithm for online and offline control design of pmsm drives. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 37(1):28–41, 2007.

[CDK06] U. K. Chakraborty, S. Das, and A. Konar. Differential evolution with local neighborhood. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 2042–2049, 2006.

[Cha36] C. Chaplin. Modern times, 1936.

[CHK+02] R. Carr, W. Hart, N. Krasnogor, J. Hirst, E. K. Burke, and J. Smith. Alignment of protein structures with a memetic evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference table of contents*, pages 1027–1034. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2002.

[CJS93] Z. H. Cho, J. P. Jones, and M. Singh. *Foundations of Medical Imaging*. Wiley-Interscience, 1993.

[CKS01] P. Cowling, G. Kendall, and E. Soubeiga. A hyperheuristic approach to scheduling a sales summit. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 176–190, 2001.

[CN] A. Caponio and F. Neri. Integrating cross-dominance adaptation in multi-objective memetic algorithms. volume 171 of *Studies in Computational Intelligence*. Springer. to appear.

[Coe02] C. A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, 2002.

[CP00] C. H. Chan and G. K. H. Pang. Fabric defect detection by fourier analysis. *IEEE Transactions on Industry Applications*, 36(5):1267–1276, 2000.

[Dau85] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Optical Society of America, Journal, A: Optics and Image Science*, 2:1160–1169, 1985.

[Dau03] J. Daugman. The importance of being random: statistical principles of iris recognition. *Pattern Recognition*, 36(2):279–291, February 2003.

[DC99] M. Dorigo and G. Di Caro. Ant colony optimization: a new meta-heuristic. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 2, page 1477 Vol. 2, 1999.

[DH95] D. Dunn and W. E. Higgins. Optimal gabor filters for texture segmentation. *IEEE Transactions on Image Processing*, 4(7):947–964, 1995.

[DMC96] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1):29–41, 1996.

[EK95] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, pages 39–43, 1995.

[ES98] A. E. Eiben and C. A. Schippers. On evolutionary exploration and exploitation. *Fundamenta Informaticae*, 35(1-4):35–50, 1998.

[ES03] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.

[FS89] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *Biological Cybernetics*, 61(2):103–113, 1989.

[Gab46] D. Gabor. Theory of communication. *X*, 93(3):429–457, 1946.

[Glo89] F. Glover. Tabu search-part i. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[Glo90] F. Glover. Tabu search-part ii. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[GMK02] R. Gamperle, S. D Muller, and P. Koumoutsakos. A parameter study for differential evolution. *Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, pages 293–298, 2002.

[GMW82] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, 1982.

[Gol89] D. E. Goldberg. Genetic algorithms in search. *Optimization, and Machine Learning. Addison-Wesley*, 432, 1989.

[Gra78] G. H. Granlund. In search of a general picture processing operator. *Computer Graphics and Image Processing*, 8(2):155–178, 1978.

[GS97] D. K. Giles and D. C. Slaughter. Precision band spraying with machine-vision guidance and adjustable yaw nozzles. *Transactions of the ASAE*, 40(1):29–36, 1997.

[GSZ04] G. Gilboa, N. Sochen, and Y. Y. Zeevi. Image enhancement and denoising by complex diffusion processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1020–1036, 2004.

[GW07] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2007.

[HJ61] R. Hooke and T. A. Jeeves. Direct search solution of numerical and-statistical problems. *J. Assoc. Comput. Mach*, 8(2):212–229, 1961.

[HKG00] A.D. Hoover, V. Kouznetsova, and M. Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging*, 19(3):203–210, 2000.

[Hor86] B. K. P. Horn. *Robot Vision*. MIT Press, 1986.

[HS05] H. H. Hoos and T. Stützle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, 2005.

[IHR+00] J. Iivarinen, K. Heikkinen, J. Rauhamaa, P. Vuorimaa, and A. Visa. Defect detection scheme for web surface inspection. *International journal of Pattern Recognition and Artificial Iintelligence*, 14(6):735–755, 2000.

[IRPR04] J. Iivarinen, R. Rautkorpi, J. Pakkanen, and J. Rauhamaa. Content-based retrieval of surface defect images with picsom. *International Journal of Fuzzy Systems*, 6(3):160, 2004.

[IYM03] H. Ishibuchi, T. Yoshida, and T. Murata. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, 7(2):204–223, 2003.

[Jah00] B. Jahne. *Computer Vision and Applications: A Guide for Students and Practitioners*. Academic Press, 1st edition, May 2000.

[Jai89] A. K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1989.

[JKW85] P.K Sahoo J.N Kapur and A.K.C Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, 29(3):273–285, 1985.

[JRL97] A. K. Jain, N. K. Ratha, and S. Lakshmanan. Object detection using gabor filters. *Pattern Recognition*, 30(2):295–309, 1997.

[KÖ3] J. K. Kämäräinen. *Feature extraction using Gabor filters*. Lappeenrannan teknillinen yliopisto, Lappeenranta, 2003.

[KGV83] S. Kirkpatric, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[KMM01] T. Karkkainen, K. Majava, and M. M. Makela. Comparison of formulations and solution methods for image restoration problems. *Inverse Problems*, 17(6):1977–1995, 2001.

[KP02] A. Kumar and G. K. H. Pang. Defect detection in textured materials using gabor filters. *IEEE Transactions on Industry Applications*, 38(2):425–440, 2002.

[KR79] R. L. Kirby and A. Rosenfeld. A note on the use of (gray level, local average gray level) space as an aid in threshold selection. 1979.

[Kra04] N. Krasnogor. Towards robust memetic algorithms. *Recent Advances in Memetic Algorithms, Springer, Berlin, Germany*, 2004.

[Lee06] G. Lee. The wall street journal: Will all of us get our 15 minutes on a youtube video? http://online.wsj.com/public/article/SB115689298168048904-5wWyrSwyn6RfVfz9NwLk774VUWc_20070829.html, August 2006.

[LHKM04] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina. Real-coded memetic algorithms with crossover hill-climbing. *Evolutionary Computation*, 12(3):273–302, 2004.

[LHN89] F. Luk, V. Huynh, and W. North. Measurement of surface roughness by a machine vision system. *Journal of Physics E: Scientific Instruments*, 22:977–980, 1989.

[LKCL00] J. J. J. Lien, T. Kanade, J. F. Cohn, and C. C. Li. Detection, tracking, and classification of subtle changes in facial expression. *Journal of Robotics and Autonomous Systems*, 31(3):131–146, 2000.

[LL02] J. Liu and J. Lampinen. On setting the control parameter of the differential evolution method. *Proceedings of the 8th International Conference on Soft Computing (MENDEL 2002)*, page 11?18, 2002.

[LL03] J. Liu and J. Lampinen. Population size adaptation for differential evolution algorithm using fuzzy logic. *Intelligent Systems Design and Applications*, 2003.

[LL05] J. Liu and J. Lampinen. A fuzzy adaptive differential evolution algorithm. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 9(6):448–462, 2005.

[LMD98] V. Leemans, H. Magein, and M. F. Destain. Defects segmentation on golden delicious apples by using colour machine vision. *Computers and Electronics in Agriculture*, 20(2):117–130, 1998.

[Loc00] M. Locatelli. Simulated annealing algorithms for continuous global optimization: Convergence conditions. *Journal of Optimization Theory and Applications*, 104(1):121–133, 2000.

[LQSB06] J.J. Liang, A.K. Qin, P.N. Suganthan, and S. Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, 10(3):281–295, 2006.

[LSDJ06] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 2(1):1–19, 2006.

[LW02] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher lineardiscriminant model for face recognition. *IEEE Transactions on Image Processing*, 11(4):467–476, 2002.

[LZ00] J. Lampinen and I. Zelinka. On stagnation of the differential evolution algorithm. *Proceedings of MENDEL*, pages 76–83, 2000.

[Maa04] H. Maaranen. *On heuristic Hybrid Methods and Structured Point Sets in Global Continous Optimization*. Jyväskylä Studies in Computing. Jyväskylä University Printing House, 2004.

[Mic95] Z. Michalewicz. A survey of constraint handling techniques in evolutionary computation methods. *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, 1995.

[MM00] W. Y. Ma and B. S. Manjunath. Edgeflow: a technique for boundary detection and image segmentation. *IEEE Transactions on Image Processing*, 9(8):1375–1388, 2000.

[MN89] P. Moscato and M. Norman. A competitive and cooperative approach to complex combinatorial search. Technical Report 790, 1989.

[MNR92] R. Mehrotra, K. R. Namuduri, and N. Ranganathan. Gabor filter-based edge detection. *Pattern Recognition*, 25(12):1479–1494, 1992.

[Ner07] F. Neri. *Fitness Diversity Adaptation in Memetic Algorithms*, volume 81 of *Jyväskylä Studies in Computing*. Jyväskylä University Printing House, 2007.

[NI08] N. Noman and H. Iba. Accelerating differential evolution using an adaptive local search. *IEEE Transactions on Evolutionary Computation*, 12(1):107–125, 2008.

[Nib85] W. Niblack. *An introduction to digital image processing*. Strandberg Publishing Company Birkeroed, Denmark, Denmark, 1985.

[NKV08] F. Neri, N. Kotilainen, and M. Vapa. A memetic-neural approach to discover resources in p2p networks. Studies in Computational Intelligence, pages 119–136. Springer, 2008.

[NM65] J. A. Nelder and R. Mead. A simplex method for function minimisation. *Computer J.*, 7:308–313, 1965.

[NM07] F. Neri and R. A. E. Mäkinen. Hierarchical evolutionary algorithms and noise compensation via adaptation. Studies in Computational Intelligence, pages 345–369. Springer, 2007.

[NOL08] Q. H. Nguyen, Y. S. Ong, and M. H. Lim. Non-genetic transmission of memes by diffusion. In *Proceedings of the 10th annual conference on genetic and evolutionary computation*, pages 1017–1024. ACM New York, NY, USA, 2008.

[NTCO07] F. Neri, J. Toivanen, G. L. Cascella, and Y. S. Ong. An adaptive multimeme algorithm for designing hiv multidrug therapies. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, Special Issue on Computational Intelligence Approaches in Computational Biology and Bioinformatics*, 4(2):264–278, 2007.

[NTM07] F. Neri, J. Toivanen, and R. A. E Mäkinen. An adaptive evolutionary algorithm with intelligent mutation local searchers for designing multidrug therapies for hiv. *Applied Intelligence, Springer, Special Issue on Computational Intelligence in Medicine and Biology*, 27(3):219–235, December 2007.

[NW00] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, April 2000.

[OK04] Y. S. Ong and A. J. Keane. Meta-lamarckian in memetic algorithm. *IEEE Transactions on Evolutionary Computation*, 8(2):99–110, 2004.

[OLZW06] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong. Classification of adaptive memetic algorithms: a comparative study. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 36(1):141–152, 2006.

[OPM02] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 971–987, 2002.

[Ote07] George Otes. Holy moly! « flickr blog. http://blog.flickr.net/en/2007/11/13/holy-moly/, November 2007.

[otPcS95] Technical Association of the Pulp, Paper Industry Winding committee, and R. Duane Smith. *Roll and Web Defect Terminology*. Tappi, 1st edition, May 1995.

[Ots79] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[PC02] J. R. Parker and S. Chan. Dirt counting in pulp: An approach using image analysis methods. *Signal and Image Processing (SIP 2002)*, 2002.

[PSL05] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution: A Practical Approach To Global Optimization*. Springer, 2005.

[QS05] A. K. Qin and P. N. Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 2, 2005.

[Rec73] I. Rechenberg. Evolutionsstrategie. *Frommann-Holzboog, Stuttgart*, 1973.

[Rey87] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4):25–34, 1987.

[RN95] S. J. Russell and P. Norvig. *Artificial Intelligence: Modern Approach*. Prentice Hall, 1st edition, 1995.

[Ros60] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.

[RSC$^+$07] P. Rogata, E. Di Sotto, F. Câmara, A. Caramagno, J.M. Rebordao, B. Correia, P. Duarte, and S. Mancuso. Design and performance assessment of hazard avoidance techniques for vision-based landing. *Acta Astronautica*, 61(1-6):63–77, 2007.

[RY05] T. P. Runarsson and X. Yao. Search biases in constrained evolutionary optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 35(2):233–243, 2005.

[SB97] S. M. Smith and J. M. Brady. Susan—a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, May 1997.

[SBM05] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection using evolutionary gabor filter optimization. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):125–137, 2005.

[SBM06] Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *IEEE Transactions on Pattern Analysis and Machine Iintelligence*, pages 694–711, 2006.

[SBO94] V. Srinivasan, P. Bhatia, and S. H. Ong. Edge detection using a neural network. *Pattern Recognition*, 27(12):1653–1662, 1994.

[Sch77] H. P. Schwefel. Numerische optimierung von computer-modellen mittels der evolutionsstrategie, volume 26 of interdisciplinary systems research. *Birhauser, Basel*, 1977.

[Sez90] M. I. Sezan. A peak detection algorithm and its application to histogram-based image data reduction. *Computer Vision, Graphics, and Image Processing*, 49(1):36–51, 1990.

[SHB98] M. Sonka, V. Hlavac, and R. Boyle. Image processing, analysis, and machine vision second edition. *Brooks/Cole, Thomson Asia Pte Led, United State of America*, pages 123–133, 1998.

[SLCJ+05] J. V. B. Soares, J. J. G. Leandro, R. M. Cesar-Jr, H. F. Jelinek, and M. J. Cree. Retinal vessel segmentation using the 2-d morlet wavelet and supervised classification. *Arxiv preprint cs.CV/0510001*, 2005.

[Smi07] J. E. Smith. Coevolving memetic algorithms: A review and progress report. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 37(1):6–17, 2007.

[Sob05] J. L. Sobral. Optimised filters for texture defect detection. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 3, 2005.

[SP95] R. Storn and K. Price. Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces": Technical report. *International Computer Science Institute, Berkley*, 1995.

[SP97] R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[SPJ+04] X. Song, B. W. Pogue, S. Jiang, M. M. Doyley, H. Dehghani, T. D. Tosteson, and K. D. Paulsen. Automated region detection based on the contrast-to-noise ratio in near-infrared tomography. *Applied Optics*, 43(5):1053–1062, 2004.

[SS04] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13:146, 2004.

[SZ05] N. G. Shankar and Z. W. Zhong. Defect detection on semiconductor wafer surfaces. *Microelectronic Engineering*, 77(3-4):337–346, 2005.

[TPS+03] M. Turtinen, M. Pietikäinen, O. Silvén, T. Maenpää, and M. Niskanen. Paper characterisation by texture using visualisation-based training. *International Journal of Advanced Manufacturing Technology*, 22(11/12):890–898, December 2003.

[Tsa95] W-H. Tsai. *Moment-preserving thresholding: a new approach*, pages 44–60. IEEE Computer Society Press, 1995.

[VT04] J. Vesterstrom and R. Thomsen. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1980–1987 Vol.2, 2004.

[Web88] S. Webb. *The Physics of Medical Imaging*. Inst of Physics Pub Inc, 1988.

[WMCJ97] D. H. Wolpert, W. G. Macready, IBMAR Center, and C. A. San Jose. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[WR83] J. M. White and G. D. Rohrer. Image thresholding for optical character recognition and other applications requiring character image extraction. *IBM Journal of Research and Development*, 27(4):400–411, 1983.

[Wri96] M. H Wright. Direct search methods: Once scorned, now respectable. *In Numerical Analysis 1995 (Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis*, pages 191—208, 1996.

[XLF05] F. Xu, X. Liu, and K. Fujimura. Pedestrian detection and tracking with night vision. *IEEE Transactions on Intelligent Transportation Systems*, 6(1):63–71, 2005.

[YTY08] Z. Yang, K. Tang, and X. Yao. Self-adaptive differential evolution with neighborhood search. In *Proceedings of the 2008 Congress on Evolutionary Computation*, 2008.

[YYH08] Z. Yang, X. Yao, and J. He. *Making a Difference to Differential Evolution*, pages 397–414. Springer, 2008.

[ZWLK06] K. Zielinski, P. Weitkemper, R. Laur, and K. D Kammeyer. Parameter study for differential evolution using a power allocation problem including interference cancellation. *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1857–1864, 2006.

[ÖBK08] E. Özcan, B. Bilgin, and E. E. Korkmaz. A comprehensive analysis of hyper-heuristics. *Intelligent Data Analysis*, 12(1):3–23, 2008.

ORIGINAL PAPERS

PI

A MEMETIC DIFFERENTIAL EVOLUTION IN FILTER DESIGN FOR DEFECT DETECTION IN PAPER PRODUCTION

by

V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava, T. Rossi 2007

Applications of Evolutionary Computing, Lectures Notes in Computer Science, Volume 4448, pp. 320–329

https://doi.org/10.1007/978-3-540-71805-5_35

PII

AN ENHANCED MEMETIC DIFFERENTIAL EVOLUTION IN FILTER DESIGN FOR DEFECT DETECTION IN PAPER PRODUCTION

by

V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava, T. Rossi 2008

Evolutionary Computation Journal, MIT Press, Volume 16, no 4

https://doi.org/10.1162/evco.2008.16.4.529

PIII

SUPER-FIT CONTROL ADAPTATION IN MEMETIC DIFFERENTIAL EVOLUTION FRAMEWORKS

by

A. Caponio, F. Neri, V. Tirronen 2008

Soft Computing-A Fusion of Foundations, Methodologies and Applications, to appear

https://doi.org/10.1007/s00500-008-0357-1

PIV

MEMETIC DIFFERENTIAL EVOLUTION FRAMEWORKS IN FILTER DESIGN FOR DEFECT DETECTION IN PAPER PRODUCTION

by

F. Neri, V. Tirronen 2008

Evolutionary Computer Vision; Studies in Computational Intelligence, to appear

https://doi.org/10.1007/978-3-642-01636-3_7

PV

FITNESS DIVERSITY BASED ADAPTATION IN MULTIMEME ALGORITHMS: A COMPARATIVE STUDY

by

F. Neri, V. Tirronen, T. Kärkkäinen, T. Rossi 2007

Proceedings of the IEEE Congress on Evolutionary Computation, pp. 2374–2381

http://dx.doi.org/10.1109/CEC.2007.4424768

PVI

ON MEMETIC DIFFERENTIAL EVOLUTION FRAMEWORKS: A STUDY OF ADVANTAGES AND LIMITATIONS IN HYBRIDIZATION

by

F. Neri, V. Tirronen 2008

Proceedings of the IEEE World Congress on Computational Intelligence, pp. 2135–2142

https://doi.org/10.1109/CEC.2008.4631082

PVII

THE "NATURA NON FACIT SALTUS" PRINCIPLE IN MEMETIC COMPUTING

by

V. Tirronen, F. Neri, K. Majava, T. Kärkkäinen 2008

Proceedings of the IEEE World Congress on Computational Intelligence, pp. 3882–3889

https://doi.org/10.1109/CEC.2008.4631325

PVIII

DIFFERENTIAL EVOLUTION WITH FITNESS DIVERSITY SELF-ADAPTATION

by

V. Tirronen, F. Neri 2009

Nature-Inspired Algorithms for Optimisation, Studies in Computational Intelligence, to appear

https://doi.org/10.1007/978-3-642-00267-0_7

PIX

A FAST RANDOMIZED MEMETIC ALGORITHM IN HIGHLY MULTI-MODAL PROBLEMS

by

V. Tirronen, F. Neri 2007

Proceedings of EuroGEN 2007