Online Parameter Adaptation Methods for Population-Based Metaheuristics

A Dissertation

submitted to the designated by the General Assembly
of the Department of Computer Science and Engineering
Examination Committee

by

Vasileios A. Tatsis

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

University of Ioannina

March 2019

Advisory Committee:

- **Konstantinos E. Parsopoulos**, Associate Professor, Department of Computer Science & Engineering, University of Ioannina, Greece (advisor)
- **Thomas Bartz-Beielstein**, Professor, Faculty of Computer Science and Engineering, Institute for Data Science, Engineering and Analytics, TH Köln, Germany
- **Ilias S. Kotsireas**, Professor, Department of Physics and Computer Science, Wilfrid Laurier University, Canada

Examining Committee:

- **Konstantinos E. Parsopoulos**, Associate Professor, Department of Computer Science & Engineering, University of Ioannina, Greece
- **Thomas Bartz-Beielstein**, Professor, Faculty of Computer Science and Engineering, Institute for Data Science, Engineering and Analytics, TH Köln, Germany
- **Ilias S. Kotsireas**, Professor, Department of Physics and Computer Science, Wilfrid Laurier University, Canada
- **Isaac E. Lagaris**, Professor, Department of Computer Science & Engineering, University of Ioannina, Greece
- **Konstantinos D. Blekas**, Associate Professor, Department of Computer Science & Engineering, University of Ioannina, Greece
- **Dimitris G. Papageorgiou**, Associate Professor, Deptartment of Materials Science and Engineering, University of Ioannina, Greece
- **Konstantina Skouri**, Associate Professor, Department of Mathematics, University of Ioannina, Greece

Dedication

To my parents Vasiliki and Antonis

Acknowledgements

First and foremost I would like to express my sincere and deepest gratitude to my advisor, Professor Konstantinos E. Parsopoulos, for his valuable guidance and continuous support from the beginning of my PhD studies, his patience, motivation, enthusiasm, and immense knowledge. His guidance was crucial, in my research and writing of this thesis. Our excellent collaboration has given me the opportunity to expand my research knowledge and skills that helped me during my research endeavor, as well as my mind perception in general. Our long discussions have been helpful and motivating, offering valuable, life lessons. I could not have imagined having a better advisor, friend and especially mentor for my PhD study.

Besides my advisor, I would like to express my sincere thanks to the members of the Advisory Committee, Professor Thomas Bartz-Beielstein and Professor Ilias Kotsireas, for their encouragement and insightful comments. Their advices and constructive comments during my studies have been of great importance.

My thanks also goes to the members of the Examination Committee, Professor Isaac Lagaris, Professor Konstantinos Blekas, Professor Dimitris Papageorgiou and Professor Konstantina Skouri for their constructive criticism on my work.

I would like to also express my thanks to the High Performance Intelligent Computing and Signal Processing lab (HICASP), as well as in my labmate and friend Dimitris Souravlias for our long stimulating discussions, and for the pleasant atmosphere in the office.

Moreover, after all these years of studies and work, I couldn't possibly come up with a complete list of all the people who had truly supported me. However, I want to thank from my heart all my friends and colleagues, for their continuous support in this great journey, as well as my life-partner for her unconditional support through the years of my PhD studies.

Last but not least, I would like to deeply express my gratitude to the most valuable

persons, my parents, for their unique love, guidance and continuous spiritual support during writing this thesis and all my life pursuits so far.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AOV	Average Objective Value
CEC-2013	Special Session on Real-Parameter Single-Objective Optimization
DE	Differential Evolution
DEGPA	Differential Evolution with Grid-Based Adaptation
DEGPOA	DE with Grid-based Parameter and Operator Adaptation
eDEGPA	Enhanced Differential Evolution with Grid-Based Adaptation
eDEGPOA	Enhanced DE with Grid-based Parameter and Operator Adaptation
GPALS	Gradient-based Parameter Adaptation with Line Search
GPALS-DE	Gradient-based Parameter Adaptation with Line Search on Differential Evolution
GPALS-PSO	Gradient-based Parameter Adaptation with Line Search on Particle Swarm Optimization
GPAM	Grid-based Parameter Adaptation Method (Scalar Parameters)
GPAM*	Grid-based Parameter Adaptation Method (Mixed Parameters)
IQR	Interquartile Range
OVSD	Objective Value Standard Deviation
PSO	Particle Swarm Optimization
PSOPNA	Particle Swarm Optimization with Parameter and Neighborhood Adaptation
SOCO	Test suite from Special Issue on Large-Scale Continuous Optimization Problems of the Soft Computing Journal

Abstract

Vasileios A. Tatsis, Ph.D., Department of Computer Science and Engineering, University of Ioannina, Greece, March 2019.
Online Parameter Adaptation Methods for Population-Based Metaheuristics.
Advisor: Konstantinos E. Parsopoulos, Associate Professor.

Optimization problems lie in the core of scientific and technological development. They appear in almost every decision-making process, under various types and forms. A multitude of algorithms have been proposed in relevant literature to solve optimization problems. However, theoretical evidence suggests that the development of an overall optimal algorithm is impossible. For this reason, problem-specific optimization algorithms have been developed, incorporating a variety of features and ad hoc operations that exploit specific properties of the corresponding optimization problem.

Typically, optimization algorithms have control parameters that adjust their dynamic with critical impact on their performance. Thus, proper parameter tuning becomes the cornerstone of efficient problem solving. There is a continuous line of research on parameter tuning methods since the early development of optimization algorithms. The majority of these methods addresses the tuning problem offline, i.e., prior to the algorithm's execution. Established offline methods are based on statistical methodologies to identify promising parameter configurations, and their results may be reusable in problems of similar type. However, they neglect the algorithm's feedback and performance fluctuations during its run. The alternative approach is the use of online methods that dynamically adapt the parameters during the algorithm's run. These methods exploit real-time performance data and, hence, they can make informative decisions on the parameter adaptation. This usually comes at the cost of non-reusable decisions.

The main goal of the present thesis is the development of new online parameter adaptation methods that can be particularly useful for the class of metaheuristic

optimization algorithms. The first part of the dissertation comprises the necessary background information on the current state-of-the-art and the optimization algorithms that will be used for demonstration purpose. In the second part of the thesis, two new online parameter adaptation methods are proposed. The first method, called Grid-based Parameter Adaptation Method, is based on grid search in the parameter space. The proposed method can be used on any algorithm and tackles both scalar and discrete parameters (including categorical ones). The new method is demonstrated on two state-of-the-art metaheuristics. For this purpose, two established benchmark suites are also considered. The second proposed method, called Gradient-based Parameter Adaptation Method with Line Search, replaces the grid search with approximate gradient search in the parameter space. The search procedure is further equipped with a recently proposed gradient-free line search technique. These modifications offer additional performance improvement with respect to the grid-based method, as revealed by the relevant performance assessment.

Εκτεταμενη Περιληψη

Βασίλειος Τάτσης, Δ.Δ., Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Μάρτιος 2019.
Online Μέθοδοι Προσαρμογής Παραμέτρων σε Πληθυσμιακούς Μεταευρετικούς αλγορίθμους.
Επιβλέπων: Κωνσταντίνος Ε. Παρσόπουλος, Αναπληρωτής Καθηγητής.

Τα προβλήματα βελτιστοποίησης βρίσκονται στον πυρήνα της επιστημονικής και τεχνολογικής έρευνας. Εμφανίζονται σχεδόν σε κάθε διαδικασία λήψης αποφάσεων, υπό διάφορους τύπους και μορφές. Για την επίλυση προβλημάτων βελτιστοποίησης έχουν προταθεί πολλοί αλγόριθμοι στη σχετική βιβλιογραφία. Ωστόσο, θεωρητικές μελέτες έδειξαν ότι είναι αδύνατη η ανάπτυξη ενός καθολικά βέλτιστου αλγορίθμου. Για το λόγο αυτό, η έρευνα επικεντρώνεται στην ανάπτυξη αλγορίθμων βελτιστοποίησης για συγκεκριμένα προβλήματα, οι οποίοι ενσωματώνουν ποικίλα χαρακτηριστικά και ad hoc λειτουργίες που εκμεταλλεύονται συγκεκριμένες ιδιότητες του αντίστοιχου προβλήματος βελτιστοποίησης.

Τυπικά, οι αλγόριθμοι βελτιστοποίησης έχουν παραμέτρους ελέγχου που προσαρμόζουν τη δυναμική τους με κρίσιμο αντίκτυπο στην απόδοσή τους. Έτσι, η σωστή προσαρμογή παραμέτρων αποτελεί ακρογωνιαίο λίθο για την αποτελεσματική επίλυση προβλημάτων. Για το λόγο αυτό, υπάρχει συνεχές και αυξανόμενο ερευνητικό ενδιαφέρον για τις μεθόδους προσαρμογής παραμέτρων. Η πλειονότητα αυτών των μεθόδων αντιμετωπίζει το πρόβλημα προσαρμογής παραμέτρων offline, δηλαδή πριν από την εκτέλεση του αλγορίθμου. Καθιερωμένες μέθοδοι αυτού του τύπου βασίζονται σε στατιστικές μεθοδολογίες και τα αποτελέσματά τους δύνανται να επαναχρησιμοποιηθούν σε παρόμοια προβλήματα. Ωστόσο, δεν λαμβάνουν υπόψη δεδομένα που προκύπτουν κατά την εκτέλεση του αλγορίθμου, καθώς και πιθανές διακυμάνσεις στην απόδοσή του. Η εναλλακτική προσέγγιση είναι η χρήση online μεθόδων που προσαρμόζουν δυναμικά τις παραμέτρους κατά την εκτέλεση του αλ-

γορίθμου. Αυτές οι μέθοδοι εκμεταλλεύονται δεδομένα απόδοσης του αλγορίθμου που προκύπτουν σε πραγματικό χρόνο και, ως εκ τούτου, μπορούν να ενημερώνουν άμεσα τις παραμέτρους. Ωστόσο, τα αποτελέσματα αυτών των μεθόδων συνήθως δεν είναι επαναχρησιμοποιήσιμα σε παρόμοια προβλήματα.

Ο κύριος στόχος της παρούσας διατριβής είναι η ανάπτυξη νέων online μεθόδων προσαρμογής παραμέτρων, με ιδιαίτερη στόχευση στις μεταευρετικές μεθόδους βελτιστοποίησης. Το πρώτο μέρος της διατριβής περιλαμβάνει τις απαραίτητες βασικές πληροφορίες σχετικά με το τρέχον state-of-the-art και τους αλγορίθμους βελτιστοποίησης που θα χρησιμοποιηθούν για την επίδειξη των νέων μεθόδων. Στο δεύτερο μέρος της διατριβής προτείνονται δύο νέες μέθοδοι προσαρμογής παραμέτρων. Η πρώτη μέθοδος, που ονομάζεται Grid-based Parameter Adaptation Method, βασίζεται στην αναζήτηση πλέγματος στο χώρο των παραμέτρων. Η προτεινόμενη μέθοδος μπορεί να χρησιμοποιηθεί σε οποιονδήποτε αλγόριθμο και αντιμετωπίζει τόσο τις πραγματικές όσο και τις διακριτές παραμέτρους (συμπεριλαμβανομένων των κατηγορικών παραμέτρων). Η νέα μέθοδος εφαρμόζεται σε δύο δημοφιλείς μεταευρετικούς αλγορίθμους. Για το σκοπό αυτό, χρησιμοποιούνται δύο βασικές σουίτες δοκιμαστικών προβλημάτων. Η δεύτερη προτεινόμενη μέθοδος, η οποία ονομάζεται Gradient-based Parameter Adaptation Method with Line Search, αντικαθιστά την αναζήτηση πλέγματος με προσεγγιστική αναζήτηση παραγώγων στο χώρο των παραμέτρων. Η διαδικασία αναζήτησης είναι επιπλέον εφοδιασμένη με μια πρόσφατη τεχνική ευθύγραμμης αναζήτησης χωρίς παραγώγους. Οι παραπάνω τροποποιήσεις προσφέρουν πρόσθετη βελτίωση απόδοσης σε σχέση με τη μέθοδο πλέγματος, όπως αποκαλύπτεται από τη σχετική πειραματική αξιολόγηση.

Chapter 1

Introduction

1.1 Overview

The continuous technological evolution has created the need for more enhanced algorithmic tools in every aspect of scientific research. During the past decade, Optimization has been placed in the center of scientific research. Optimization problems are met literally everywhere, requiring diverse optimization algorithms to tackle them. Theoretical results such as the No Free Lunch theorem [1] suggest that there is no universal algorithm that can tackle all problems equally well.

The parameter tuning problem has been a central research topic for many years, resulting in a variety of tuning methods distinguished in two categories: offline and online methods. In early approaches, offline parameter tuning was applied, prior to the algorithm's execution on the studied problem. This approach requires deep knowledge of the studied problem as well as experience from the practiotioner's side. The good performance of such approaches is strongly connected to resource-intensive preprocessing, based on trial-and-error experimentation. Usually, this requires the

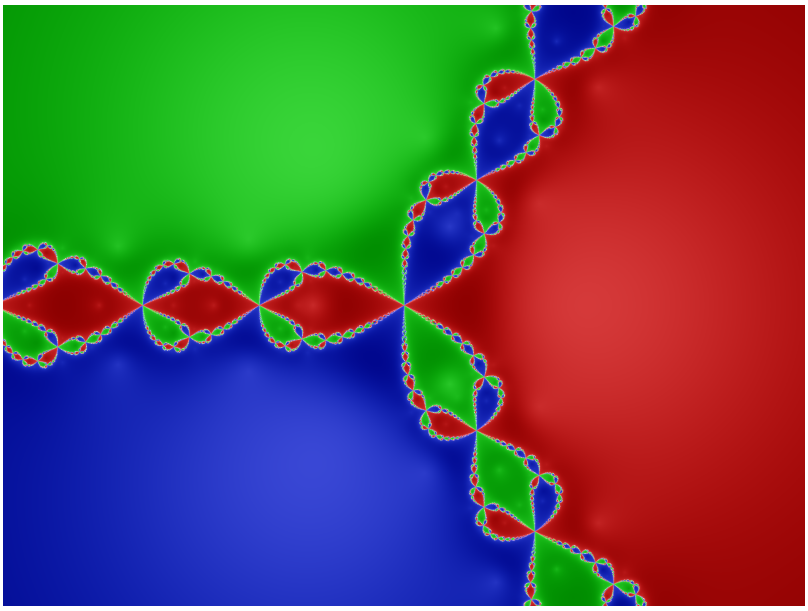


Figure 1.1: Convergence regions of the Newton-Raphson method for the problem $z^3 - 1 = 0$ in the complex plane.

execution of large number of algorithm variants under alternative parameters, and performance comparisons based on statistical methodologies. Very often, this procedure requires more time than the solution of the studied problem itself. However, it can produce results that may be reusable in problem of similar type.

On the other hand, online parameter adaptation, also called parameter control, does not require preprocessing although at the cost of hardly reusable results. Various ad-hoc procedures have been proposed for this purpose in the literature. The present thesis proposes two general-purpose online parameter adaptation methods, which are also algorithm-independent.

Metaheuristics have been frequently used to tackle optimization problems where good (sub-)optimal solutions are needed in reasonable time. However, they have proved to be rather sensitive on their parameter settings. For this reason, they constitute an excellent testbed for the developed methods that are presented in the present thesis.

1.2 Motivation

Early motivation for the developments presented in the present thesis comes from deterministic optimization and, specifically, from the Newton-Raphson convergence

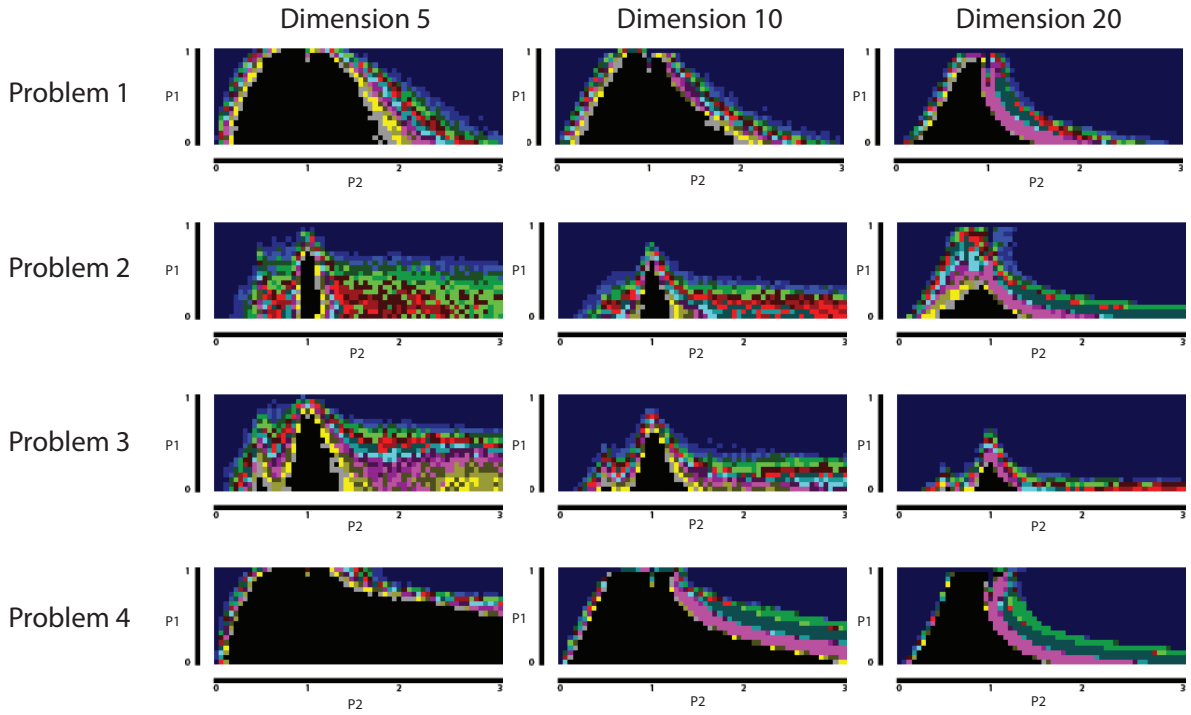


Figure 1.2: Convergence regions of the DE algorithm for various parameter values.

fractal depicted in Fig. (1.1). This image refers to the covergence regions of Newton-Raphson applied on the problem $z^3 - 1 = 0$ in the complex plane. The three main colors, (red, green, blue) are associated with the detected roots respectively. We can clearly distinguish solid regions in which, initialization of the algorithm leads to the closest root, as well as fractal-like regions where convergence is unpredictable. Clearly, these regions show that the choice of the starting point is very crucial.

Simiraly, the performance of metaheuristics is also strongly connected to their parameterization, as it is illustrated in Fig. (1.2). This figure illustrates the performance of the Differential Evolution (DE) algorithm in its scalar parameter space, for various problems and dimensions. Specifically, for each pair of parameter values $(P1, P2)$, DE was executed 100 times and its performance in terms of solution quality was recorded. Each color refers to a specific solution precision, with black corresponding to the best one. The clear appearance of solid regions containing promising parameters set was the main motivation for the development of the online parameter adaptation methods presented in the thesis.

1.3 Thesis Contribution

In the first part of the dissertation, motivation for the use of parameter adaptation methods is provided. Then, in Chapter 3, a novel online parameter adaptation method for population-based metaheuristics is proposed. The core idea is based on grid search in the parameter domain, aiming at discovering promising regions of parameter values that are associated with enhanced performance. For this purpose, different quality criteria are considered in terms of solution quality and diversity. Different variants are proposed for handling continuous and discrete parameters in a single-objective or multi-objective manner. The derived methods are demonstrated on two popular metaheuristics, using two state-of-the-art test suites that include high-dimensional and low-dimensional problems. Experimental evidence reveals the effectiveness of the proposed approach and its competitiveness against other state-of-the-art adaptive algorithms. The following contributions are achieved by the proposed method:

- It is completely autonomous from the user during the algorithm's execution.
- Three quality criteria are considered.
- It can handle both continuous and discrete (including categorical) parameters.
- It does not require preprocessing, sparing significant amount of computational time and resources.
- It can ameliorate the performance of the algorithm even when unsuitable initial parameter values are selected by the user.
- It has sound performance on a large number of test problems.
- It has competitive performance against other state-of-the-art adaptive or self-adaptive algorithms.
- It does not depend on the algorithm, but offers a rather generic parameter adaptation scheme.

In Chapter 4, a more sophisticated method is proposed. Its core mechanism is based on performance gradient estimations while it is also equipped with line search. The use of the gradient-based approach for parameter adaptation results in significant improvement in convergence speed. Similarly to its predecessor, the gradient-based

method is highly autonomous. Following the same experimental configuration as for the grid-based approach, the method is applied on the two test suites, revealing its performance superiority against other state-of-the-art algorithms. The following contributions are obtained from the the proposed approach:

- It is completely autonomous from the user during its application.
- It does not depend on the algorithm.
- It does not require preprocessing, thereby sparing significant amount of computational time and resources.
- It is less sensitive to initial parameter values selected by the user.
- It has better convergence properties than the grid-based predecessor.
- It promotes more fine-grained parameter adaptation.
- It is favorably compared to other state-of-the-art adaptive or self-adaptive algorithms.

1.4 Thesis Layout

The thesis is organized as follows: Chapter 2 provides a brief description of parameter adaptation methods as well as the necessary background in metaheuristics. Chapter 3 presents a novel online parameter adaptation method based on grid-search estimations, demonstrated on two metaheuristics and benchmarked on two established test suites. Chapter 4 presents an enhanced method equipped with gradient based estimations and line search. Finally, Chapter 5 concludes the dissertation and outlines directions for future work.

Chapter 2

Background Information

2.1 Introduction

Metaheuristics are widely acknowledged as essential tools for solving difficult optimization problems in diverse scientific fields [2]. Albeit solution optimality is not guaranteed, they can provide (sub-)optimal solutions of real-world problems in reasonable time. This renders metaheuristics a valuable alternative especially in cases where traditional optimization tools or analytical approaches fail.

The performance of metaheuristics typically depends on their control parameters as well as on the particular problem instance [3–5]. The calibration and fine-tuning of the control parameters is a major issue in metaheuristics design.

2.2 Population-based Metaheuristics

Without loss of generality, consider the general form of the n-dimensional bound-constrained continuous optimization problem,

$$\min_{x \in X \subset \mathbb{R}^n} f(x), \tag{2.1}$$

where the search space X is defined as a hypercube,

$$X = [l_1, u_1] \times \cdots \times [l_n, u_n],$$

with l_i and u_i denoting the lower bound and the upper bound of the i-th direction component, respectively. Let also the sets of indices

$$D \triangleq \{1, 2, \ldots, n\}, \qquad I \triangleq \{1, 2, \ldots, N\}, \tag{2.2}$$

where D refers to the set of direction components and I refers to the indices of population members, respectively. The objective function value of a vector $x_i \in X$, $i \in I$, will be simply denoted also as

$$f_i = f(x_i).$$

Finally, the function `rand( )` denotes the pseudo-random number generator that produces uniformly distributed real numbers in the range $[0, 1]$.

2.2.1 Differential Evolution

Differential Evolution (DE) was introduced by R. Storn and K. Price [6] as a population-based, stochastic optimization algorithm for numerical optimization problems. Althought DE has flexible search operators, which are very convenient for the user, it is also characterized by sensitive dynamics with respect to its control parameters [7]. Nevertheless, its adaptability, simplicity, and efficiency has placed it among the most popular metaheuristics, counting a significant number of relevant works [8]. DE employs a different mechanism for producing new candidate solutions than the dominant probabilistic mechanisms of Evolutionary Algorithms (EAs). Specifically, it uses differences of the population's members to perturb existing candidate solutions.

Algorithm 2.1 Pseudocode of the DE algorithm.

INPUT: Population P; Population size N; Parameters F, CR; Maximum iterations t_{max}
$initialize(P)$
while $t < t_{max}$ **do**
  for $i = 1 : N$ **do**
    Choose mutually different indices r_s, $2 \leqslant s \leqslant 5$
    $u_i^{(t+1)} \leftarrow mutation\left(x_{r_s}^{(t)}, F\right)$ */* Use Eqs. (2.3)-(2.7) */*
    $v_i^{(t+1)} \leftarrow crossover\left(x_i^{(t)}, u_i^{(t+1)}, CR\right)$ */* Use Eq. (2.8) */*
    $evaluate\left(v_i^{(t+1)}\right)$
    $x_i^{(t+1)} \leftarrow selection\left(x_i^{(t)}, v_i^{(t+1)}\right)$ */* Use Eq. (2.9) */*
  end for
  Update index g of best individual
  $t \leftarrow t + 1$
end while

Similarly to EAs, mutation, crossover, and selection operators are applied to evolve the population.

The standard DE algorithm assumes a fixed-size population of size N,

$$P = \{x_1, x_2, \ldots, x_N\},$$

to probe the search space. Each individual x_i is an n-dimensional vector,

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^\top \in X, \quad i \in I,$$

and constitutes a candidate solution of the problem at hand. The population is randomly initialized in X, typically following a uniform distribution.

The population P is iteratively evolved by applying two operators, namely *mutation* and *crossover*, on each individual. Then, a *selection* phase takes place, where each individual of the new population competes with its corresponding original individual, and the best one passes to the next iteration. These operators are iteratively applied until a termination condition is satisfied.

The steps of DE are outlined in Algorithm 2.1. Let t denote the iteration counter. According to the mutation operator, a new vector $u_i^{(t+1)}$ is derived for each individual $x_i^{(t)}$ at iteration t, through combinations of existing individuals. A variety of mutation operators have been proposed in the relevant literature. The most common ones are

the following:

DE/Best/1

$$u_i^{(t+1)} = x_g^{(t)} + F\left(x_{r_1}^{(t)} - x_{r_2}^{(t)}\right), \tag{2.3}$$

DE/Rand/1

$$u_i^{(t+1)} = x_{r_1}^{(t)} + F\left(x_{r_2}^{(t)} - x_{r_3}^{(t)}\right), \tag{2.4}$$

DE/Current-to-Best/2

$$u_i^{(t+1)} = x_i^{(t)} + F\left(x_g^{(t)} - x_i^{(t)} + x_{r_1}^{(t)} - x_{r_2}^{(t)}\right), \tag{2.5}$$

DE/Best/2

$$u_i^{(t+1)} = x_g^{(t)} + F\left(x_{r_1}^{(t)} - x_{r_2}^{(t)} + x_{r_3}^{(t)} - x_{r_4}^{(t)}\right), \tag{2.6}$$

DE/Rand/2

$$u_i^{(t+1)} = x_{r_1}^{(t)} + F\left(x_{r_2}^{(t)} - x_{r_3}^{(t)} + x_{r_4}^{(t)} - x_{r_5}^{(t)}\right), \tag{2.7}$$

where $F \in (0, 1]$ is a fixed, user-defined parameter also called the *scale factor* [8]; g denotes the index of the best individual, i.e., the one with the lowest function value; and r_s are mutually different integers selected from the set I, also different than i.

After mutation, crossover is applied to produce a *trial vector*,

$$v_i = (v_{i1}, v_{i2}, \ldots, v_{in})^\top, \quad i \in I,$$

for each individual x_i. There are two main types of crossover operator. The most popular one is the *binomial crossover* where,

$$v_{ij}^{(t+1)} = \begin{cases} u_{ij}^{(t+1)}, & \text{if } \mathcal{R} \leqslant CR \text{ or } j = RN(n), \\ x_{ij}^{(t)}, & \text{otherwise,} \end{cases} \tag{2.8}$$

where $j \in D$; $\mathcal{R}$ is the realization of a uniformly distributed random variable in the range $[0, 1]$; $CR \in (0, 1]$ is the second control parameter of the algorithm, called the *crossover rate*; and $RN(n)$ is an integer randomly selected from the set D. The two conditions in the first branch of Eq. (2.8) ensure that at least one component of the mutated vector $u_i^{(t+1)}$ is inherited to the trial vector.

The alternative *exponential crossover* operator initially copies $x_i^{(t)}$ into the trial vector $v_i^{(t+1)}$. Subsequently, it randomly selects a component index $k \in D$ and sets the cor-

Algorithm 2.2 Pseudocode of the PSO algorithm.

INPUT: Swarm S; Population size N; Parameters c_1, c_2, w; Maximum iterations t_{max}
$initialize(S)$
while $t < t_{max}$ **do**
  for $i = 1 : N$ **do**
    Update index g of best particle
    $u_i^{(t+1)} \leftarrow update\_velocity\left(p_i^{(t)}, u_i^{(t)}, x_i^{(t)}, c_1, c_2, w\right)$ /* Use Eq. (2.10) */
    $x_i^{(t+1)} \leftarrow update\_position\left(x_i^{(t)}, u_i^{(t+1)}\right)$ /* Use Eq. (2.11) */
    $evaluate\left(x_i^{(t+1)}\right)$
    $p_i^{(t+1)} \leftarrow update\_best\_position\left(x_i^{(t+1)}, p_i^{(t)}\right)$ /* Use Eq. (2.12) */
  end for
  $t \leftarrow t + 1$
end while

responding component $v_{ik}^{(t+1)} = u_{ik}^{(t+1)}$. Then, starting from the index $k+1$, a number of components of $v_i^{(t+1)}$ are assigned the corresponding component values of $u_i^{(t+1)}$, according to a stochastic condition. After the first failure of the condition, the rest of the components retain the values initially copied from $x_i^{(t)}$ [9].

Finally, *selection* takes place where the trial vector $v_i^{(t+1)}$ competes with $x_i^{(t)}$ and the new individual for the next iteration of the algorithm is selected as follows,

$$x_i^{(t+1)} = \begin{cases} v_i^{(t+1)}, & \text{if } f\left(v_i^{(t+1)}\right) \leqslant f_i^{(t)}, \\ x_i^{(t)}, & \text{otherwise.} \end{cases} \tag{2.9}$$

The algorithm iteratively applies the same procedure until a termination condition is reached. Eventually, the individual x_g is reported as the best detected solution.

The parameters of DE (including its crossover operator) have been shown to be crucial for its convergence [10]. While proper parameter values can render DE a very efficient algorithm, mild perturbations may result in significantly inferior performance. Taking into consideration that proper parameter values are typically problem-dependent, it is reasonable to expect that parameter tuning of DE can be a laborious task.

2.2.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a popular metaheuristic proposed by J. Kennedy

and R. Eberhart [11]. Its robustness and efficiency in a variety of complex optimization problems, as well as its easy implementation and minor requirements on the problem's model, has placed it among the most popular algorithms.

The main concept of PSO is based on a *swarm* of search points called *particles*. The particles cooperatively probe the search space through adaptable position shifts, while retaining in *memory* their best visited positions. PSO employs a swarm of search points,

$$S = \{x_1, x_2, \ldots, x_N\},$$

which is randomly and uniformly initialized in the search space X. Each particle,

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \in X, \qquad i \in I,$$

represents a candidate solution of the problem. Its objective value will be simply denoted as $f_i = f(x_i)$. The particle iteratively moves in X according to an adaptable position shift, called the *velocity*, which is denoted as,

$$v_i = \{v_{i1}, v_{i2}, \ldots, v_{in}\},$$

and its best visited position stored in memory is denoted as,

$$p_i = \{p_{i1}, p_{i2}, \ldots, p_{in}\} \in X.$$

The dynamic of PSO is strongly dependent on exchange of information among the particles. Specifically, the particles communicate their best positions to other particles through communication channels that define their *neighborhoods*. This socially shared information is then used to guide their move.

There are two major PSO models with respect to the extent of information sharing. In the global (*gbest*) model each particle is aware of the overall best position of the whole swarm. In the local (*lbest*) model, information is shared only among a restricted number of predefined particles. The communication channels among the particles are determined by the employed *neighborhood topology*. The most popular neighborhood topology is the *ring*, where the i-th particle takes into account the findings of particles with indices belonging in a set,

$$NB_i = \{i - m, \ldots, i - 1, i, i + 1, \ldots, i + m\}.$$

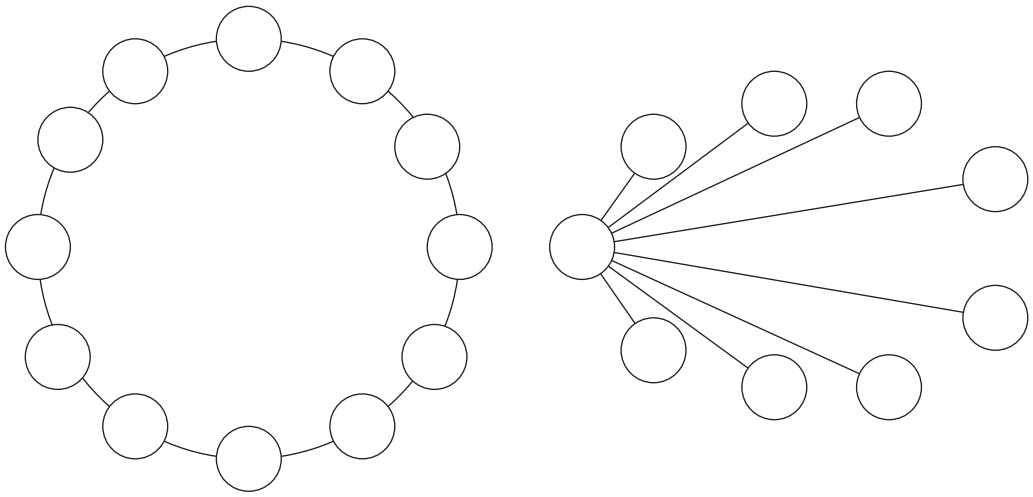


Figure 2.1: Popular neighborhood topologies: ring (left) and star (right).

This neighborhood topology can be depicted as a ring-shaped graph (hence the name) where particles lie on the nodes and each one is connected with its two immediate neighbors only. The parameter m defines the size of the neighborhood and it is called the *neighborhood's radius*. The *ring* and *star* topologies are graphically illustrated in Figure 2.1, in which the nodes denote the particles and the edges denote the communication channels.

Let the index g_i denote the best particle in NB_i, i.e.,

$$g_i = \arg\min_{k \in NB_i} f(p_k),$$

and let t denote the iteration counter. Then, the update equations of plain PSO are given as follows [12]:

$$v_{ij}^{(t+1)} = w\, v_{ij}^{(t)} + c_1\, \texttt{rand}()\, \left(p_{ij}^{(t)} - x_{ij}^{(t)}\right) + c_2\, \texttt{rand}()\, \left(p_{g_i j}^{(t)} - x_{ij}^{(t)}\right), \tag{2.10}$$

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)}, \tag{2.11}$$

where $i \in I$ and $j \in D$; w is a velocity clamping parameter called *inertia weight*; and c_1, c_2, are two scalars called the *cognitive parameter* and *social parameter*, respectively.

Finally, each particle updates its best position at every iteration as follows,

$$p_i^{(t+1)} = \begin{cases} x_i^{(t+1)}, & \text{if } \; f_i^{(t+1)} \leqslant f\left(p_i^{(t)}\right), \\ p_i^{(t)}, & \text{otherwise.} \end{cases} \tag{2.12}$$

The algorithm iterates until a user-defined termination condition is satisfied. The steps of PSO are outlined in Algorithm 2.2, while the reader can find thorough presentation of PSO and its variants in [12, 13].

2.3 General Population-based Algorithm Model

A general population-based algorithm model will be henceforth considered to describe the proposed online parameter adaptation algorithms in the following chapters. According to the presented population-based algorithms in the previous sections, the general model assumes a fixed-size population,

$$P = \{x_1, x_2, \ldots, x_N\},$$

of size N to probe the search space. Each individual x_i is an n-dimensional vector,

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^\top \in X, \quad i \in I,$$

and constitutes a candidate solution of the problem. The population is randomly initialized in the search sapce X, following a uniform distribution. Also, the scalar parameters of the algorithm will be henceforth denoted as $\rho_1, \rho_2, \ldots, \rho_{np}$.

2.4 Parameter Adaptation Methods

There are two major categories of parameter setting methods that dominate the relevant literature, namely offline and online methods [3, 4].

2.4.1 Offline

Offline parameter tuning is based on existing performance data of the algorithm collected from previous applications on similar problems. If such data is unavailable, it is collected through a preprocessing phase based on preliminary experimentation with different parameter settings on the problem at hand. The best-performing parameter values are then adopted in the algorithm. Typically, this approach requires significant computational effort. The current experience with numerous metaheuristics suggests

that this effort can be comparable or even higher to the effort needed for solving the problem itself. Most offline parameter tuning approaches are based on statistical methodologies. Typical examples are the Design of Experiments [14], which provides a sound statistical methodology to analyze, compare and learn from experiments, F-Race [5], which is a racing algorithm based on statistical comparisons for selecting the best configuration out of a predefined set of candidate configurations; the Sequential Model-Based Optimization [15], which uses explicit regression models to describe the dependence of target algorithm performance on parameter settings; and ParamILS [16], which conducts an iterated local search in parameter configuration space. Such approaches can offer promising results at the cost of additional implementation and experimentation effort. Nevertheless, their outcome is often reusable in a wide range of similar problems. Relative works can be found also in [17,18].

2.4.2 Online

In contrast to the offline tuning approaches, online methods aim at dynamically adapting the parameters of the algorithm, based on feedback during its run [3,16]. The popularity of such methods can be attributed to the lack of preliminary experimentation and the limited user intervention. On the other hand, online methods often exhibit two major weaknesses:

1. *Overspecialization*: they are usually based on *ad hoc* procedures developed for a specific algorithm or problem type.

2. *New control parameters*: the number of new parameters introduced by the tuning method may significantly expand the parameter domain.

The overspecialization issue renders the outcome of online methods hardly reusable even in different runs of the same algorithm on a specific problem. Moreover, increasing significantly the number of parameters by introducing new ones may increase the sensitivity of the algorithm and, concurrently, impose the necessity for tuning the tuning procedure itself. These weaknesses have offered motivation for the development of new online methods during the past decades.

2.5 Literature review

There is a number of adaptive online approaches with respect to the two meta-heuristics described in the previous section, namely DE and PSO. In [19] Brest *et al.* proposed a self-adaptive algorithm that probabilistically assigns control parameter values. Zhao *et al.* [20] proposed a self-adaptive scheme with multi-trajectory search. Also, a distributed adaptive scheme with scale factor inheritance was proposed in [21] by Weber *et al.*, where subpopulations are connected in a ring topology, each one having its own scale factor. Moreover, a self-adaptive algorithm learning from previous experiences was proposed in [22], while in [23] a parallel multi-population model with random connection topology was proposed. A survey paper revising the terminology, as well as the classification of control mechanisms, was provided in [24].

Polakova *et al.* [25] introduced a controlled restart DE in which the restarting conditions are derived from the difference of extremal values of the objective function and the estimated maximum distance among the points in the current population. Moreover, the jDElscop variant was introduced in [26], combining three strategies and a population-reduction mechanism. A generalized adaptive scheme that is based on probability distribution was proposed in [27].

Zhang and Sanderson in JADE [28] algorithm use an external archive to adapt the control parameters. In the same vein, in [29] Tanabe and Fukunaga proposed a new parameter adaptation technique (called SHADE), which uses historical memory of successful control parameter settings to guide the selection of future values. Moreover, in [30] Tanabe *et al.* introduced an enhanced version of SHADE, called L-SHADE, which incorporates success-history based parameter adaptation.

Tvrdík *et al.* also adapt the control parameters of the DE in [31] through competition within the algorithm, while a new adaptive variant with twelve competing strategies was proposed in [32]. A comparative analysis of the binomial and exponential crossover variants was conducted in [33] and [34], providing also theoretical results. Brest *et al.* in [35] proposed a self-adaptive DE with small, varying population size.

LaTorre *et al.* in [36] have analyzed the behavior of a hybrid algorithm that combines two heuristics. An adaptive memetic DE with global and local neighborhood-based mutation operators was proposed by Piotrowski in [37]. Finally, in [38] Segura *et al.* studied the relation between exploration and exploitation with respect to the

Table 2.1: Summary of the SOCO test problems

No.	Functions	Range	f^*
1	Shifted Sphere Function	$[-100, 100]^n$	-450
2	Shifted Schwfel's Problem 2.21	$[-100, 100]^n$	-450
3	Shifted Rosenbrock's Function	$[-100, 100]^n$	-390
4	Shifted Rastrigin's Function	$[-5, 5]^n$	-330
5	Shifted Griewank's Function	$[-600, 600]^n$	-180
6	Shifted Ackley's Function	$[-32, 32]^n$	-140
7	Schwefel's Problem 2.22	$[-10, 10]^n$	0
8	Schwefel's Problem 1.2	$[-65.536, 65.536]^n$	0
9	Extended f_10	$[-100, 100]^n$	0
10	Bohachevsky	$[-15, 15]^n$	0
11	Schaffer	$[-100, 100]^n$	0
12	Hybrid Composition Function	$[-100, 100]^n$	0
13	Hybrid Composition Function	$[-100, 100]^n$	0
14	Hybrid Composition Function	$[-5, 5]^n$	0
15	Hybrid Composition Function	$[-10, 10]^n$	0
16	Hybrid Composition Function	$[-100, 100]^n$	0
17	Hybrid Composition Function	$[-100, 100]^n$	0
18	Hybrid Composition Function	$[-5, 5]^n$	0
19	Hybrid Composition Function	$[-10, 10]^n$	0

scale factor in DE.

In the same vein, a number of PSO variants with dynamically adjusted parameters have been proposed in literature. In [39] the inertia weight of PSO is automatically controlled based on the swarm's distribution and particles' fitness values. Moreover, in [40] an adaptable PSO algorithm was proposed based on a stability criterion, while in [41] a fuzzy system was employed for the same purpose.

2.6 Benchmark Suites

It is reasonable to expect that parameter adaptation methods, such as the ones proposed in the present thesis, can be beneficial for an algorithm particularly in cases of demanding optimization problems. Undoubtedly, large-scale problems consisting of hundreds of decision variables constitute an appropriate testbed for investigating the potential benefits. Nevertheless, problems of lower dimension can verify the wide applicability of the method. The solution quality criterion for all the considered problems is the objective value error defined as,

$$\varepsilon^* = f\left(x^*\right) - f\left(x_{\text{opt}}\right), \tag{2.13}$$

Table 2.2: Summary of the CEC-2013 test problems

No.	Functions	Range	f^*
1	Sphere Function	$[-100, 100]^n$	-1400
2	Rotated High Conditioned Elliptic Function	$[-100, 100]^n$	-1300
3	Rotated Bent Cigar Function	$[-100, 100]^n$	-1200
4	Rotated Discus Function	$[-100, 100]^n$	-1100
5	Different Powers Function	$[-100, 100]^n$	-1000
6	Rotated Rosenbrock's Function	$[-100, 100]^n$	-900
7	Rotated Schaffers Γ7 Function	$[-100, 100]^n$	-800
8	Rotated Ackley's Function	$[-100, 100]^n$	-700
9	Rotated Weierstrass Function	$[-100, 100]^n$	-600
10	Rotated Griewank's Function	$[-100, 100]^n$	-500
11	Rastrigin's Function	$[-100, 100]^n$	-400
12	Rotated Rastrigin's Function	$[-100, 100]^n$	-300
13	Non-Continuous Rotated Rastrigin's Function	$[-100, 100]^n$	-200
14	Schwefel's Function	$[-100, 100]^n$	-100
15	Rotated Schwefel's Function	$[-100, 100]^n$	100
16	Rotated Katsuura Function	$[-100, 100]^n$	200
17	Lunacek Bi-Rastrigin Function	$[-100, 100]^n$	300
18	Rotated Lunacek Bi-Rastrigin Function	$[-100, 100]^n$	400
19	Expanded Griewank's plus Rosenbrock's Function	$[-100, 100]^n$	500
20	Expanded Scaffer's Γ6 Function	$[-100, 100]^n$	600
21	Composition Function 1 $(n = 5, Rotated)$	$[-100, 100]^n$	700
22	Composition Function 2 $(n = 3, Unrotated)$	$[-100, 100]^n$	800
23	Composition Function 3 $(n = 3, Rotated)$	$[-100, 100]^n$	900
24	Composition Function 4 $(n = 3, Rotated)$	$[-100, 100]^n$	1000
25	Composition Function 5 $(n = 3, Rotated)$	$[-100, 100]^n$	1100
26	Composition Function 6 $(n = 5, Rotated)$	$[-100, 100]^n$	1200
27	Composition Function 7 $(n = 5, Rotated)$	$[-100, 100]^n$	1300
28	Composition Function 8 $(n = 5, Rotated)$	$[-100, 100]^n$	1400

where x^* is the solution achieved by the algorithm and x_{opt} is the known globally optimal solution of the problem.

2.6.1 SOCO Test Suite

For large-scale problems, the test suite provided in the *special issue on large-scale continuous optimization problems* of the Soft Computing journal [42] (henceforth denoted as SOCO) is considered. This test suite consist of 19 large-scale continuous optimization problems, henceforth denoted as f_1-f_{19}, of dimension $n = 50, 100, 200, 500$. Among them, f_1-f_6 come from the CEC-2008 test suite [43], accompanied by 13 shifted and hybrid test problems of high complexity, including separable and non-separable problems, which are reported in Table 2.1. The main goal determined by the test suite is the detection of the known global minimizers of the test problems within the tight limit of $q_{\max} = 5000 \times n$ function evaluations (FEs). Also, a number of 25 experiments

is required for the derivation of statistical results.

The test suite is provided on the internet[1] along with complementary material. This includes complete results for three base algorithms, namely DE with exponential crossover, CHC [44], and GCMAES [45], along with the average solution values for 13 additional algorithms, namely SOUPDE [46], DE-D^{40}+M^{m} [47], GaDE [48], jDElscop [26], SaDE-MMTS [20], MOS [49], MA-SSW-Chains [50], RPSO-vm [51], Tuned IPSOLS [52], EvoPROpt [53], EM323 [54], VXQR1 [55], and GODE [56]. Note that adaptive and self-adaptive algorithms are included among them. Beside the exponential DE, the most popular binomial DE is additionally considered as a base algorithm in the experiments presented later in the present thesis. The source code of the test problems is available online [57].

2.6.2 CEC-2013 Test Suite

In addition to the large-scale problems, the mainstream *CEC-2013* test suite from the special session on real-parameter single-objective optimization [58] is also considered. It consists of 28 benchmark problems denoted as f_1-f_{28}, including unimodal, multimodal, and composite functions, all reported in Table 2.2. Only the dimensions $n = 30$ and $n = 50$ are considered, since problems of lower dimension can hardly offer useful information for approaches such as the proposed ones. According to this test suite, the search space is equal to $[-100, 100]^n$ for all test problems. Also, the maximum number of function evaluations is equal to $q_{\max} = 10000 \times n$. This is in contrast to the SOCO test suite where non-symmetrical search spaces and half the number of function evaluations are considered for problems of larger dimension. The optimal solutions for all test problems are known, and the suggested number of runs per problem is 51 as dictated by the test suite.

Complementary material provided by the test suite is also available on the internet [59]. This includes complete results for one of the most competitive DE variants, namely L-SHADE [30] and SHADE [29], as well as for DEcfbLS [60], jande [61], DE_APC [62], and PVADE [63]. Note that all the provided DE algorithms are adaptive and self-adaptive.

[1] http://sci2s.ugr.es

2.6.3 Further Implementation Details

At this point, it shall be underlined that the algorithms used for comparisons with the proposed methods of this thesis were already tuned for the corresponding test of problems. However, the required computational budget for their tuning is typically neglected in all relevant studies, although it can be comparable or even higher than the reported budgets required for solving a problem. The method introduced in the present thesis do not require such preliminary experimentation. Although a fair comparison would require to allocate to the proposed approaches this additional preprocessing budget, it was decided to push their performance to the limit and assess them under exactly the same computational budget as for the competitor tuned algorithms.

All implementations of the proposed methods in this thesis were made in the `C` programming language. The results for the competitor algorithms were either adopted from the publicly available data or obtained through the available sources in the corresponding references of the test suites. The OpenMPI library[2] was used for parallelization. The parallelization does not interfere with the method's dynamic but only expedites the experiments. Thus, the same results are received with the serial version of the method under identical initial conditions and seeding. All experiments were conducted on a Beowulf cluster consisting of Intel® i7 machines with 8GB RAM, providing 8 CPUs each, and running under Ubuntu Linux.

2.7 Synopsis

In this chapter, the required background information regarding the employed state-of-the-art metaheuristics was outlined. A general population-based optimization algorithm model was defined in order to be used for the general descriptions of the proposed methods in the following chapters. Additionally, the description of the test problems used in the dissertation was outlined. Finally, implementation details were provided.

[2] http://www.open-mpi.org

Chapter 3

New Grid-Based Parameter Adaptation Method

In this chapter, the novel grid-based parameter adaptation method is proposed and demonstrated on two state-of-the-art metaheuristics. The presentation includes a general description of the method's basic scheme, as well as specialization for the cases of Differential Evolution and Particle Swarm Optimization. Experimental assessment is offered on the two benchmark suites, SOCO and CEC-2013, including the relevant statistical analysis.

3.1 Introduction

Dynamic parameter adaptation allows the algorithm to identify suitable parameter settings during its run. Existing approaches are based on the algorithm's previous performance, as well as on current performance estimations [3, 16]. Typically, these methods need minimal user intervention, although at the cost of additional computational requirements in terms of running time.

The main goal of the proposed grid-based method is the dynamic adaptation of the control parameters of the corresponding algorithm during its run with minimal additional computational cost and user intervention. For the scalar parameters, this is achieved by discretizing the parameter space, forming a grid. Then, local search is conducted on the parameter grid, based on estimations of the algorithm's performance under different neighboring parameter settings. The most promising parameter setting is then adopted from the algorithm for a number of iterations. The same steps are iteratively applied in order to identify promising parameter settings in different stages of the optimization procedure. The search can be conducted either serially or in parallel, taking advantage of modern computer systems.

The grid-based method is demonstrated on two state-of-the-art metaheuristics, namely Differential Evolution (DE) and Particle Swarm Optimization (PSO), for dynamically adapting their control parameters. However, proposed method is applicable with any optimization algorithm. DE was selected mainly due to its recognized sensitivity in parameter values and operator type [10], which results in challenging parameter control problems. In the same vein, PSO was selected for further verification on different population-based metaheuristics.

3.2 Proposed Method

In the following paragraphs, the proposed grid-based parameter adaptation method, henceforth called *Grid-based Parameter Adaptation Method* (GPAM) is presented in detail. The basic scheme for scalar parameter adaptation of population-based metaheuristics is initially exposed, followed by an updated variant that dynamically adjusts also discrete parameters.

3.2.1 Basic Scheme

The first step in the proposed method is the discretization of the scalar parameter space. Let the parameters of the algorithm be $\rho_1, \rho_2, \ldots, \rho_{np}$, and a specific range $[l_{\rho_i}, u_{\rho_i}]$ be defined for each one. Discretization step sizes $\lambda_1, \lambda_2, \ldots \lambda_{np}$ are specified for the each parameter, respectively. Small step sizes offer more fine-grained search in parameter space, although the convergence to good parameter values may become slow. On the other hand, large discretization steps may result in overshooting appropriate parameter values. The optimal step size depends always on the algorithm and, specifically, its parameter sensitivity. Nevertheless, previous experience with an algorithm often provides useful insight for this purpose.

The discretized np-dimensional parameter space,

$$\mathcal{G} = \{(\rho_1, \rho_2, \ldots, \rho_{np});\ \rho_i \in \{l_{\rho_i}, l_{\rho_i} + \lambda_i, l_{\rho_i} + 2\lambda_i, \ldots, u_{\rho_i}\},\ \ \forall i = 1, \ldots, np\}\,,$$

is henceforth called the *grid*. Each interior point in $\mathcal{G}$ has $3^{np}-1$ immediate neighboring points. The proposed method assumes that the algorithm is initialized to a random population P_{pri}, called the *primary population*, and assumes an initial parameter vector at the center of the grid, i.e.,

$$\rho_i = \frac{l_{\rho_i} - u_{\rho_i}}{2}, \qquad i = 1, \ldots, np. \tag{3.1}$$

Then it evolves for t_{pri} iterations. The selection of initial parameters on the grid center is reasonable in lack of additional domain knowledge suggesting better choice. Naturally, if such information is available, it can be easily exploited to accelerate the detection of suitable parameter values. For example, consider the case where data is available a priori, suggesting that the objective function has a multitude of minimizers densely concentrated in specific parts of its domain. In view of such information, the initial parameters may be set accordingly to promote exploitation over exploration. On the other hand, small number of sparsely distributed minima may need initial parameters that promote exploration. Thus, domain knowledge can be beneficial for the algorithm's efficiency.

After the t_{pri} iterations, the primary population is copied into 3^{np} *secondary populations*, each one evolved using a neighboring parameter vector. The secondary populations are independently evolved using their assigned parameter values for a small

number of iterations, $t_{sec} \ll t_{pri}$, in order to locally estimate the performance of the current population with the corresponding parameter values. The best-performing secondary population is then adopted as the new primary population along with its parameter vector. These steps constitute a full cycle of the method, which continues with new cycles until the available computational budget (usually function evaluations) is exceeded. The main phases of the method are analyzed below:

Cloning Phase

The primary population P_{pri} is copied into 3^{np} *secondary populations* P_{sec_j} with $j \in \{1, 2, \ldots, 3^{np}\}$, one for each neighboring parameter vector in the grid, including the current one. If $(\rho_1, \rho_2, \ldots, \rho_{np})$ is the current parameter vector for the primary population, then the secondary population P_{sec_j} has a parameter vector $(\rho_{1j}, \rho_{2j}, \ldots, \rho_{np,j})$ with,

$$\rho_{ij} = \rho_{ij} + \gamma\, \lambda_i, \quad \gamma \in \{-1, 0, 1\}, \quad i \in \{1, 2, \ldots, np\}. \tag{3.2}$$

Obviously, the case $\gamma_i = 0$ corresponds to the current parameter vector of the primary population.

Performance Estimation Phase

Each secondary population is evolved for $t_{sec} \ll t_{pri}$ iterations. This is called the *performance estimation* phase of the method, and provides a local estimation of the algorithm's performance for the current primary population, using the specific neighboring parameter vectors. This step can be executed either serially or in parallel using one master node (primary population) and 3^{np} slave nodes (secondary populations). Modern desktop computers offer adequate resources for such implementations.

Subsequently, the primary and the secondary populations are compared in terms of their *average objective values* (AOV). For a population P of size N, this is defined as the average function value of its members:

$$\bar{f}_P = \frac{1}{N} \sum_{i=1}^{N} f\left(x_i\right). \tag{3.3}$$

The best population among the primary and the secondary ones, along with its parameters vector, are adopted, as thew new primary population.

The presented procedure produces a trajectory of parameter vectors in the grid

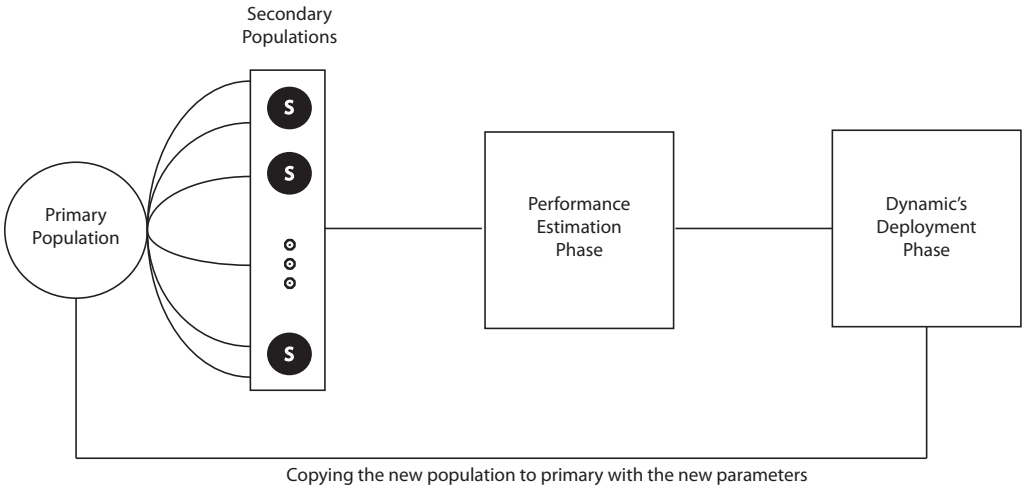


Figure 3.1: A complete cycle of the proposed GPAM method.

by tracking local improvements of estimated performance. This way, even marginal improvements achieved by a secondary population would result in adopting new parameter vectors that may be recalled in subsequent cycles of the procedure. This effect can produce undesirable cyclic or oscillating trajectories, which can be alleviated by imposing a minimal improvement threshold $\varepsilon_{\min}$ for accepting a new primary population. Thus, the best secondary population P^*_{sec} and its parameters $(\rho^*_1, \ldots, \rho^*_{np})$ replace the primary population P_{pri} and its current parameters $(\rho_1, \rho_2, ..., \rho_{np})$ only if it holds that,

$$\bar{f}_{pri} - \bar{f}_{sec^*} \geqslant \varepsilon_{\min} > 0.$$

The value of $\varepsilon_{\min}$ can be set taking into consideration the desired solution accuracy.

Dynamic's Deployment Phase

The new primary population, with its new parameter vector $(\rho_1, \ldots, \rho_{np})$, is evolved according to the implemented algorithm for t_{pri} iterations and this completes a full cycle of the proposed, grid-based parameter adaptation method.

Analysis and Improvements

The application of the three phases above is henceforth referred as a *cycle* of the parameter adaptation method, in order to avoid confusion with the iterations of the primary and secondary populations. Thus, a cycle c involves t_{pri} iterations of the

Algorithm 3.1 Pseudocode of the basic GPAM method

```
initialize(P_pri)
M ← 3^np   /* Number of secondary populations */
Evolve primary population P_pri for t_pri iterations.
while (not termination) do
   /* Performance estimation phase */
   for i = 1 : M do
      Copy P_pri to secondary population P_sec,i.
      Assign parameters to P_sec,i.
      Evolve P_sec,i for t_sec iterations.
   end for
   /* Update primary population */
   Find the best-performing secondary population P*_sec.
   if (f̄_pri − f̄_sec* ⩾ ε_min) then
      P_pri ← P*_sec
      Replace worst individuals of P_pri with the overall best vectors of
            all P_sec,i (if they are better).
   end if
   /* Dynamic's deployment phase */
   Evolve primary population P_pri for t_pri iterations.
end while
```

primary population and $3^{np} \times t_{sec}$ iterations of the secondary populations, i.e., a total number of,

$$q_c = (t_{pri} + 3^{np}\, t_{sec}) \times N,$$

function evaluations, where N stands for the population size. A complete cycle of the proposed GPAM method is illustrated in Fig. 3.1. The maximum number of complete cycles that can be performed by the method, given a maximum budget of q_{max} function evaluations, can be *a priori* determined as,

$$c_{\max} = \left\lfloor \frac{q_{\max}}{q_c} \right\rfloor, \tag{3.4}$$

where $\lfloor . \rfloor$ is the floor function.

Moreover, alternative performance metrics can be used instead of AOV. A typical alternative is the use of each population's overall best objective value. However, this metric may become misleading because, for practical purpose, the number t_{sec} of performance-estimation iterations shall be typically kept low (5 to 10 iterations) in order to spare computational budget. Thus, using solely the overall best value as performance measure renders the procedure vulnerable to temporarily optimal solu-

tions that may be rapidly discovered. This was also verified in experimental testing in early stages of GPAM's development, and for this reason it was abandoned.

Furthermore, in order to take full advantage of the discoveries of all secondary populations, the utilization of the overall best individual of each unselected secondary population in the new primary population is also considered. Thus, the worst 3^{np} individuals of the new primary population are replaced by the 3^{np} overall best members of the secondary populations, if they have better objective values.

The proposed GPAM approach with the modification discussed above is outlined in Algorithm 3.1. Steps 3-19 constitute a complete cycle of the method. The number of iterations t_{pri} and t_{sec} for the primary and secondary populations, respectively, can be set to fixed values. As previously mentioned, small values of 5 to 10 iterations are suggested for t_{sec} since only rough performance estimations are required for the secondary populations. Contrary to this, t_{pri} shall take higher values in order to allow the primary population with the selected parameter vector to deploy its dynamic. Based on suggestions in literature [6], the value $t_{pri} = 10 \times n$, with n being the problem's dimension can be considered as default value.

An alternative strategy is the dynamic adaptation of t_{pri} from a minimum value $t_{pri}^{\min}$ to a maximum value $t_{pri}^{\max}$ during the method's execution. The rationale behind it lies on the fact that at the latest stages of optimization the population is expected to have already identified promising regions of the search space and, hence, longer running time for the dynamic's deployment phase can be beneficial. The simplest strategy is the *linear* adaptation. Thus, if c denotes the current cycle of the method and $c_{\max}$ denotes the maximum number of cycles determined by Eq. (3.4), then t_{pri} can be linearly adapted between $t_{pri}^{\min}$ and $t_{pri}^{\max}$ as follows:

$$t_{pri}(c) = \left(t_{pri}^{\max} - t_{pri}^{\min}\right) \frac{c}{c_{\max}} + t_{pri}^{\min}. \tag{3.5}$$

The extremal values $t_{pri}^{\min}$ and $t_{pri}^{\max}$ can be set by the user taking into consideration the problem at hand (especially its dimension). The current experience on numerous problems has shown that setting $t_{pri}^{\max}$ at values 40% to 50% higher than $t_{pri}^{\min}$ is a good default choice. Thus, the following default setting:

$$t_{pri}^{\min} = 10 \times n, \qquad t_{pri}^{\max} = 14 \times n,$$

is suggested, where n is the problem's dimension. Note that these parameters as well

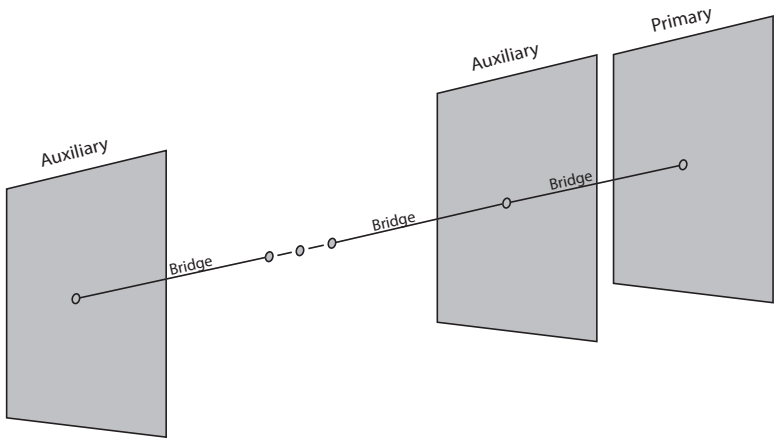


Figure 3.2: Bridging populations define multiple grids.

as the ε_{min} threshold for accepting a new primary population are optional. Hence, they do not contradict the main goal of the proposed approach to unburden the user from the effort of finding proper parameter values for the algorithm. The proposed approach can be straightforwardly parallelized in modern desktop computers by using a master node for the primary population and 3^{np} slave nodes for the concurrent evolution of the secondary populations. Naturally, additional nodes can further expedite the procedure, especially when denser grids are used.

3.2.2 Handling discrete parameters

The handling of discrete parameters is straightforward for the case of numerical values with ordering (e.g., integer parameters). In this case, the parameter space for these parameters is already discretized and the previously presented procedures can be applied without modification. However, this is not the case for discrete parameters without ordering (e.g., categorial parameters). For such cases, a new concept of bridging populations is introduced, which defines multiple grids of scalar parameters, one for each discrete parameter. This modification formulates a multi-grid parameter space, as the one illustrated in Fig. 3.2. For example, in the general algorithm model, the primary population would be copied in the 3^{np} secondary populations for adapting the np scalar parameters, as well as in dp bridging populations, which inherit the same scalar parameters as the primary population but different discrete parameters. Here, dp stands for the number of discrete parameters without ordering.

After that, each secondary and bridging population is evolved for t_{sec} iterations (performance estimation phase). The best secondary or bridging population is selected

Algorithm 3.2 Pseudocode of the GPAM* method

```
initialize(P_pri)
M ← 3^np + dp    /* Number of secondary populations */
Evolve primary population P_pri for t_pri iterations.
while (not termination) do
  /* Performance estimation phase */
  for i = 1 : M do
    if (i ⩽ 3^np) then
      /* Secondary population */
      Copy P_pri to secondary population P_sec,i.
      Assign scalar parameter vector to P_sec,i.
    else
      /* Bridging secondary population */
      Copy P_pri to secondary population P_sec,i.
      Inherit scalar parameters of P_pri to P_sec,i.
      Assign discrete parameters to P_sec,i.
    end if
    Evolve P_sec,i for t_sec iterations.
  end for
  /* Update primary population */
  Find the best-performing secondary population P*_sec.
  if (f̄_pri − f̄_sec* ⩾ ε_min) then
    P_pri ← P*_sec
    Replace worst individuals of P_pri with the overall best vectors of
        all P_sec,i (if they are better).
  end if
  /* Dynamic's deployment phase */
  Evolve primary population P_pri for t_pri iterations.
end while
```

to replace the primary population along with its parameters. This procedure can be viewed as jumping from one scalar parameter grid to another. The rest of the parameter adaptation procedure is identical to the GPAM method presented in the previous sections. The modified approach is outlined in Algorithm 3.2, and it is henceforth denoted as GPAM*.

3.3 Application on Differential Evolution

The proposed GPAM method is demonstrated on the state-of-the-art DE algorithm, initially for adapting its scalar parameters. The derived algorithm is henceforth denoted as DEGPA (Differential Evolution with Grid-Based Parameter Adaptation). Ad-

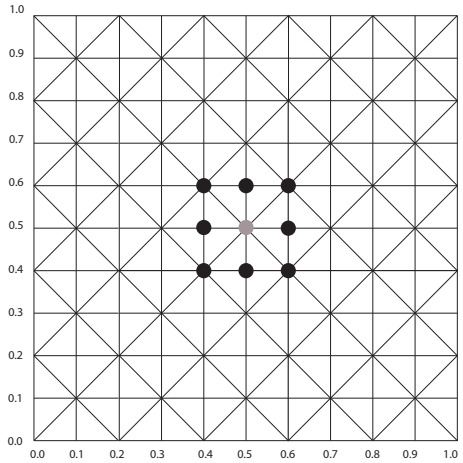


Figure 3.3: The parameter grid for $\lambda_F = \lambda_{CR} = 0.1$. Each interior point (gray node) has 8 immediate neighbors (black nodes).

ditionally, the GPAM* method is demonstrated on DE for adapting also its crossover or mutation operator along with the two scalar parameters. The experiments are conducted on the two established benchmark suites described in Section 2.6.

3.3.1 Online Adaptation of Scalar Parameters and Crossover Operator

The first step in DEGPA is the discretization of the scalar parameter space [64]. Recall that both parameters F and CR (scale factor and crossover rate) assume real values in the range $(0, 1]$. Although different ranges for F have been studied, the specific one appears to be the most common choice in relevant literature [8] and in the available source codes [65, 66]. Nonetheless, adaptation for different ranges is trivial.

Discretization step sizes λ_F and λ_{CR} are specified for the two parameters, respectively. The current experience has shown that performance differences are marginal under parameter differences smaller than 0.1. For this reason, $\lambda_F = \lambda_{CR} = 0.1$ was selected, as a reference step size to build a discretized 2-dimensional parameter space (grid),

$$\mathcal{G} = \{(F, CR);\ F, CR \in \{0.0, 0.1, \ldots, 1.0\}\},$$

Each interior point in $\mathcal{G}$ has 8 immediate neighbors as illustrated in Fig. 3.3.

The algorithm starts with a randomly initialized primary population, which is assigned the parameter pair at the center of the grid, i.e., $(F, CR) = (0.5, 0.5)$. The

primary population P_{pri} is then copied in 9 *secondary populations* $P_{sec}^{a,b}$, with $a, b \in \{-1, 0, 1\}$, one for each neighboring parameter pair in the grid. If (F, CR) is the current parameter pair for the primary population P_{pri}, then the secondary population $P_{sec}^{a,b}$ has a parameter pair (F', CR') with,

$$F' = F + a\,\lambda_F, \quad CR' = CR + b\,\lambda_{CR}, \quad a, b \in \{-1, 0, 1\}. \tag{3.6}$$

Obviously, the case $a = b = 0$ corresponds to the primary population itself.

Following the GPAM workflow, each secondary population is evolved according to the standard DE procedure for $t_{sec} \ll t_{pri}$ iterations. This step can be executed either serially or in parallel using one master node (primary population) and 9 slave nodes (secondary populations). Subsequently, the primary and the secondary populations are compared in terms of their *average objective values* (AOV), $\bar{f}_{pri}$, $\bar{f}_{sec}$, defined in Eq. (3.3). The best-performing secondary population P_{sec}^{*}, i.e., the one with minimum AOV value, and the corresponding parameter set are selected and replace the primary population P_{pri} and its current parameters if it holds that,

$$\bar{f}_{pri} - \bar{f}_{sec^*} \geqslant \varepsilon_{\min} > 0.$$

Then, the new primary population and its parameter set, is evolved according to the corresponding algorithm's procedure for $t_{pri} = 10 \times n$, iterations, where n stands for the problems dimension. The alternative dynamic adaptation of t_{pri} defined in Eq (3.5) can be used, following the guidelines of Section 3.2.1.

A cycle c of the method includes t_{pri} iterations of the primary population and $9 \times t_{sec}$ iterations of the secondary populations, i.e., a total number of,

$$q_c = (t_{pri} + 9\,t_{sec}) \times N,$$

function evaluations, where N stands for the population size. Thus, the maximum number of complete cycles that will be performed by the algorithm, can be *a priori* determined,

$$c_{\max} = \left\lfloor \frac{q_{\max}}{q_c} \right\rfloor, \tag{3.7}$$

for a prescribed maximum computational budget of $q_{\max}$ function evaluations as described in Section 3.4.

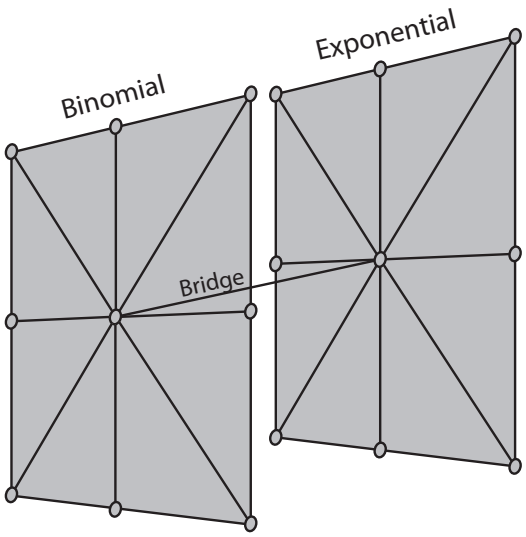


Figure 3.4: Jumping from binomial to exponential grid.

Moreover, the worst 8 individuals of the new primary population are replaced by the 8 overall bests of the secondary populations, if they have better values.

In addition to the scalar parameters adaptation, the basic DEGPA scheme can be further enhanced by adapting also the crossover operator type between binomial and exponential. This can be achieved by using an additional bridge population in the performance-estimation phase of the method, as described in Section 3.2.2. More specifically, the bridge population executes t_{sec} iterations (as the rest of the secondary populations) with the primary population, assuming the scalar parameter pair of the primary population but with different crossover type. Thus, if the primary population P_{pri} uses binomial crossover operator and parameters (F, CR), the bridge population P_{bri} is initially a copy of P_{pri} with parameters (F, CR) but uses the exponential crossover operator. After evolving P_{bri} for t_{sec} iterations, the resulting population competes with the rest of the secondary populations and the primary one, according to the GPAM* scheme presented in Section 3.2.2. If the bridge population is the winner, it becomes the primary population and its crossover operator is adopted for the following cycles of the method, until the next possible change. This procedure can be simply considered as a jump from the grid of the binomial variant to the grid of the exponential one and vice versa, as illustrated in Fig. 3.4.

This new DEGPA variant is henceforth called *enhanced DEGPA* (eDEGPA) and offers additional flexibility to DEGPA by further reducing the number of user-defined

parameters in DE. The reason for presenting both DEGPA and eDEGPA is that, as will be seen in the experimental assessment, there are problem types where the most efficient crossover operator type has been *a priori* identified. Thus, it is interesting and challenging to compare the performance between DEGPA exploiting this information and eDEGPA with no prior problem-dependent information.

The performance of DEGPA and eDEGPA was initially assessed on the SOCO test suite described in Section 2.6.1. The main goal in the experiments was to achieve competitive average performance against algorithms reported in the test suite, especially the ones with the exponential DE approach that was optimally tuned on the specific test problems. The tested DEGPA adopted the exponential crossover operator as well the mutation of Eq. (2.4), according to the setting of the base DE algorithm. On the other hand, eDEGPA was let to dynamically select between binomial and exponential crossover, while its mutation operator was the same with DEGPA.

Experiments were conducted with both DEGPA and eDEGPA on all test problems of the SOCO suite, for dimension $n = 50$, 100, 200, and 500. The available computational budget was determined according to the test suite's requirements as $q_{max} = 5000 \times n$ function evaluations [42]. Note again that the proposed algorithms did not use the additional time claimed by the competitor algorithms for their preliminary experimentation and optimal tuning.

The master-slave model was adopted for the parallel implementation, of DEGPA and eDEGPA,assigning one secondary population per slave node, while the master node was running the main procedure of the method. The considered performance measure was the objective value error defined in Eq. (2.13). For each algorithm, 25 independent experiments were conducted and the average errors were recorded. Following the setting of the DE algorithm in the SOCO test suite, the population size for both DEGPA and eDEGPA was set to $N = 60$.

Although the test suite includes only the exponential DE variant, for completeness reasons the corresponding binomial DE variant using the provided settings and source codes in [67] was also considered as a competitor algorithm. The control parameters for the two base DE variants were set to $(F, CR) = (0.7, 0.5)$ as suggested in the reported results in [67]. The two base DE variants are henceforth denoted as DE_{exp} and DE_{bin}. On the other hand, both DEGPA and eDEGPA were initiated at the central parameter pair $(F, CR) = (0.5, 0.5)$, and the initial crossover operator type for eDEGPA was the exponential one (dynamically changing during execution).

The experimental analysis was divided in two phases [64]. In the first phase, DEGPA and eDEGPA were compared against the base algorithms, namely DE_{exp}, DE_{bin}, CHC, and GCMAES. The available source code for each base algorithm was used for conducting 25 independent experiments per problem, using the exact settings reported in the SOCO test suite [57]. The achieved objective value errors were recorded for each algorithm and experiment. The means and standard deviations of the obtained errors for the proposed and the base algorithms are reported in Tables A.1 and A.2 of Appendix A. In the 500-dimensional case, results could not be obtained with the provided GCMAES source code due to excessive computation time (this is reported as "n/a" in the two Tables).

A close inspection of the results offers interesting information. Both the proposed approaches have competitive performance the base algorithms. In fact, they clearly outperform DE_{bin}, CHC, and GCMAES, in almost all test problems, especially for higher dimension. Also, they exhibit competitive performance against the best-performing base algorithm DE_{exp}. This achievement worths further attention because DE_{exp} was used with its optimal parameter setting and crossover type provided in the SOCO test suite, for the same computational budget with the proposed approaches.

For the lower dimensions ($n = 50$ and $n = 100$) eDEGPA attained superior average performance than DEGPA. This can be attributed to eDEGPA's ability to automatically select the most proper crossover type operator, since binomial crossover appears to be more beneficial in some test problems. We can also see that eDEGPA's performance gradually declines as dimension increases. This is anticipated, since eDEGPA needs additional effort to evolve the extra (bridge) populations, in order to dynamically decide on the crossover operator type. Thus, as the problems become harder, eDEGPA exceeds the available budget more rapidly than DEGPA.

In order to statistically verify the observed performance differences between each pair of algorithms, Wilcoxon rank-sum tests at confidence level 95% were conducted for all test functions. Each positive comparison where DEGPA or eDEGPA outperformed another algorithm with statistical significance was counted as a *win*. The corresponding negative comparisons were counted as *loses*. The lack of statistical significance was considered as a *tie*. The results for all statistical comparisons are given in Table 3.1, where wins, loses, and ties are denoted as "$+$", "$-$", and "$=$", respectively. The reported results verify the previous findings. Specifically, both DEGPA and eDEGPA achieved statistically better or equivalent performance with the rest of

Table 3.1: Number of wins (denoted as "+"), loses (denoted as "−"), and ties (denoted as "=") of DEGPA and eDEGPA against the base algorithms of the SOCO test suite.

		DEGPA			eDEGPA		
Dimension	Algorithm	+	−	=	+	−	=
50	DE_{bin}	14	3	2	14	3	2
	DE_{exp}	9	6	4	9	7	3
	CHC	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0
	eDEGPA	6	10	3			
100	DE_{bin}	17	1	1	17	1	1
	DE_{exp}	11	6	2	11	6	2
	CHC	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0
	eDEGPA	6	8	5			
200	DE_{bin}	17	1	1	17	1	1
	DE_{exp}	12	7	0	10	8	1
	CHC	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0
	eDEGPA	8	4	7			
500	DE_{bin}	19	0	0	19	0	0
	DE_{exp}	11	7	1	7	6	6
	CHC	19	0	0	19	0	0
	GCMAES	n/a	n/a	n/a	n/a	n/a	n/a
	eDEGPA	8	4	7			

the base algorithms in most of the test problems. Moreover, their numbers of wins exhibited increasing trend with dimension. Another interesting observation is that the best competitor, namely DE_{exp}, achieved in all cases less wins than DEGPA or eDEGPA.

Excluding DE_{exp}, eDEGPA had equal number of wins, loses, and ties with DEGPA against the other algorithms, despite the fact that eDEGPA has additional self-adaptation capabilities. Finally, the last line per dimension block in Table 3.1 reports the wins, loses, and ties of eDEGPA against DEGPA. These comparisons verify the previous observations on the superior performance of eDEGPA in lower dimension and its decline in higher dimension as a result of the extra effort imposed by the dynamic adaptation of the crossover operator type.

Figure. 3.5 illustrates some indicative trajectories of the parameter pairs in the grid for four test problems (corresponding to lines of different colors). In the case of eDEGPA, solid lines correspond to exponential crossover operators while dashed lines correspond to binomial operators.

In the second phase of experimentation on the SOCO suite, DEGPA and eDEGPA were compared to a number of different algorithms [42, 67]. The comparisons were

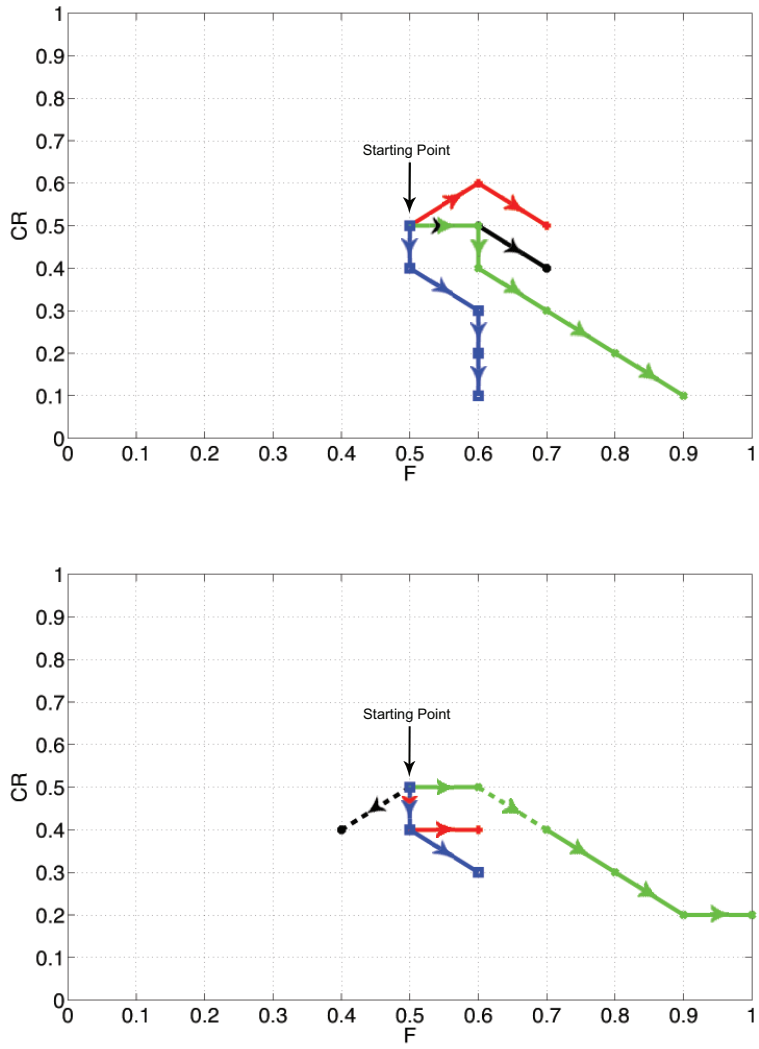


Figure 3.5: Trajectories of parameter pairs for DEGPA (up) and eDEGPA (down) for four test problems indicated with different colors. All trajectories start on the grid center $(F, CR) = (0.5, 0.5)$. For eDEGPA solid line indicates exponential crossover, while dashed line stands for binomial crossover.

based on the average error values per algorithm and test function, which are reported in Tables A.3 and A.4. The corresponding error values for the rest of the algorithms are reproduced from the original sources [67]. Also, the results of DEGPA and eDEGPA appear in both Tables A.3 and A.4 to facilitate comparisons.

The number of test problems where DEGPA and eDEGPA achieved non-inferior (equal or better) or inferior (worse) average errors than the other algorithms is graphically illustrated in Fig. 3.6 and reported in Table 3.2, offering some interesting information. On the one hand, we can see that DEGPA and eDEGPA have similarly-shaped lines, which implies consistent performance against the rest of the algorithms.

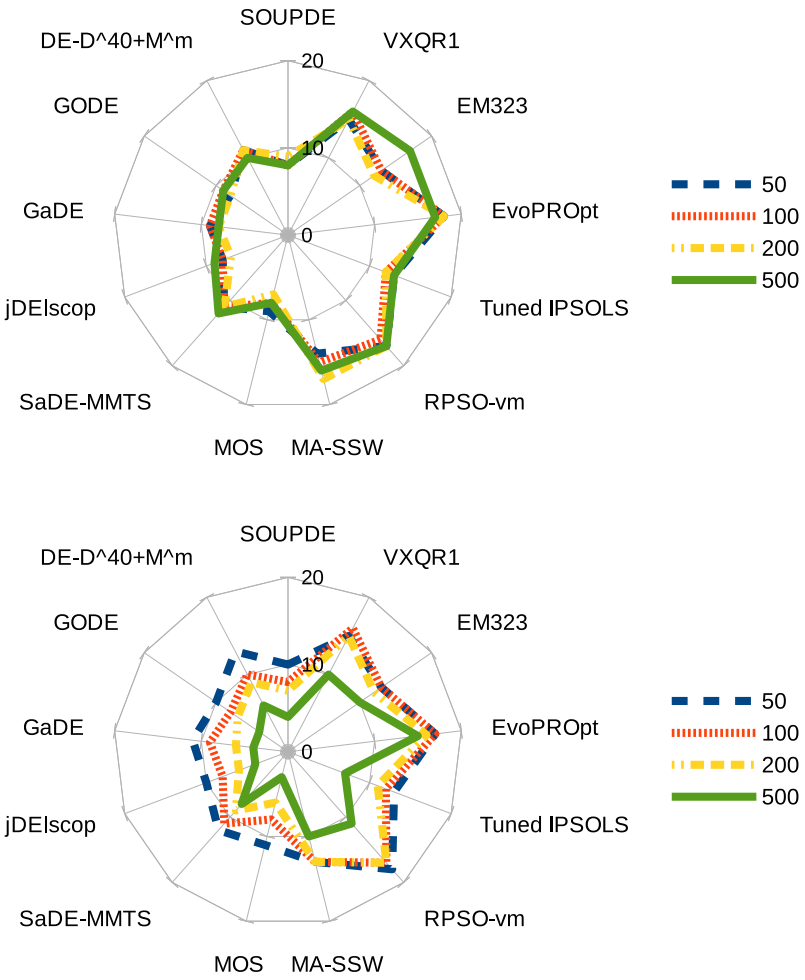


Figure 3.6: Number of test problems where DEGPA (up) and eDEGPA (down) achieved equal or better average error than different algorithms on the SOCO suite, for dimension $n = 50$, 100, 200, and 500.

Nevertheless, it can be observed that DEGPA retains its performance as dimension increases (overlapping lines), while eDEGPA exhibits declining behavior, with plot lines of higher dimension being enclosed by the lines of lower dimensions. This verifies the previous interpretation that it is the outcome of additional computational requirements. Nevertheless, it worths noting that the proposed approaches achieved highly competitive performance also to non-DE algorithms, such as PSO-based approaches, MA-SSW-Chains, EvoPROpt, EM323, and VXQR1, in all dimensions.

Table 3.2: Number of problems where DEGPA/eDEGPA exhibited inferior or non-inferior average error values than the competitor algorithms.

	Non-Inferior				Inferior			
	Dim.				Dim.			
Algorithm	50	100	200	500	50	100	200	500
DEGPA								
EvoPROpt	18	18	18	17	1	1	1	2
EM323	13	13	12	17	6	6	7	2
SOUPDE	8	8	9	8	11	11	10	11
DE-D^{40}+M^{m}	11	11	11	10	8	8	8	9
GODE	8	9	8	9	11	10	11	10
MA-SSW-Chains	14	15	17	16	5	4	2	3
GaDE	9	9	8	8	10	10	11	11
RPSO-vm	17	16	17	17	2	3	2	2
jDElscop	8	8	7	9	11	11	12	10
SaDE-MMTS	11	11	11	12	8	8	7	5
MOS	9	8	7	8	10	11	12	11
Tuned IPSOLS	13	12	12	13	6	7	7	6
VXQR1	15	16	15	16	4	3	4	3
eDEGPA								
EvoPROpt	17	17	16	15	2	2	3	4
EM323	13	13	12	10	6	6	7	9
SOUPDE	10	8	7	4	9	11	12	15
DE-D^{40}+M^{m}	13	10	9	6	6	9	10	13
GODE	10	8	7	4	9	11	12	15
MA-SSW-Chains	13	13	13	10	6	6	6	9
GaDE	11	9	6	4	8	10	13	15
RPSO-vm	18	17	17	11	1	2	2	8
jDElscop	10	8	6	4	9	11	13	15
SaDE-MMTS	12	11	9	8	7	8	10	11
MOS	11	8	6	3	8	11	13	16
Tuned IPSOLS	13	12	11	7	6	7	8	12
VXQR1	15	16	15	10	4	3	4	9

3.3.2 Online Adaptation of Scalar Parameters and Mutation Operator

In a second round of experiments with DE on the SOCO suite, the mutation operator was adapted along with the scalar control parameters using the GPAM* method [68]. The mutation operator defines the scheme that generates new search directions, hence, affecting the sampling dynamics of DE. For example, operators that involve the best individual, x_g, have been associated with rapid convergence but they are also more prone to get stuck in local minimizers. On the other hand, operators with purely random selection of the involved vectors have been shown to promote diversity. Also, the use of one or two difference vectors may have impact on the algorithm's performance.

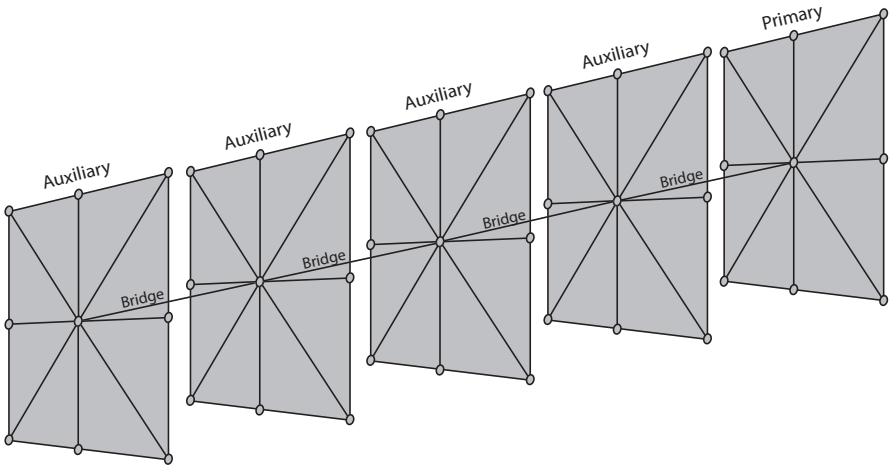


Figure 3.7: Bridging parallel grids through populations with different mutation operators.

In this framework, the eDEGPA scheme was considered with the exception that the adapted discrete parameter is the mutation operator, selected among the five operators defined in Eqs. (2.3)-(2.7). This requires the use of additional bridging populations besides the one defined in the previous eDEGPA implementation. The additional populations inherit the primary population and its parameters, but assume different mutation operator. Thus, they allow for exploration on multiple parallel grids of the scalar parameters, each one corresponding to a different mutation operator as illustrated in Fig. 3.7.

The proposed variant is henceforth called *DE with Grid-based Parameter and Operator Adaptation* (DEGPOA), and it closely follows the general GPAM* scheme of Algorithm 3.2. DEGPOA requires 9 secondary populations for the performance estimation under different parameter pairs (F, CR), as well as 4 additional secondary (bridging) populations that carry the same scalar parameter pair as the primary one, but with different mutation operators. Thus, a total of 13 secondary populations are needed for the application of DEGPOA.

The algorithm is initialized with a primary population assuming the initial parameters $(F, CR) = (0.5, 0.5)$ and a random initial mutation operator. The primary population is evolved for t_{pri} iterations (dynamic's deployment phase). Then, it is copied to the 9 secondary populations of its own scalar parameter grid, exactly as for eDEGPA in the previous section. However, in DEGPOA the primary population is also copied in the 4 bridging populations, which assume same scalar parameters but different mutation operator than the primary population. For example, if the pri-

mary population has the initial parameters mentioned above and uses the DE/Rand/1 operator, then the bridging populations would assume the same scalar parameters $(F', CR') = (F, CR) = (0.5, 0.5)$, but the DE/best/1, DE/Current-to-Best, DE/Rand/2, and DE/best/2 operators, respectively.

After that, each secondary population is evolved for t_{sec} iterations (performance estimation phase). The best secondary population is selected to replace the primary population along with its parameters and mutation operator. This procedure can be viewed as jumping from the one mutation operator's grid to another. The new primary population initiates a new cycle of the algorithm with a new dynamic's deployment phase and so on.

The selection of the primary population's initial mutation operator can be done in two ways. If an operator is known to perform well in the given problem, it is a reasonable choice to prefer it as the initial one. On the other hand, the initial operator can be randomly selected in absence of any relevant information.

Another issue that requires further investigation is the performance measure used for the assessment of the secondary populations. In [64] the AOV measure of Eq. (3.3) was used, because it is less sensitive to temporary performance improvements that may be caused by the rapid convergence to local minimizers (especially from the greedier operators). However, AOV exploits solely the objective value and neglets diversity, which is the main aspect of DE affected by the adapted mutation operator. In order to include diversity in the evaluation criteria, an additional, diversity-based performance measure, namely the *objective value standard deviation* (OVSD) was considered. For a population P of size N, OVSD is defined as:

$$\sigma_P = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\big(f(x_i) - \bar{f}_P\big)}, \qquad x_i \in P, \tag{3.8}$$

where $\bar{f}_P$ is the AOV defined in Eq. (3.3). OVSD measures the diversity of the objective values of the population and can be used as a rapidly calculated indicator of population's diversity. Higher values of OVSD can be associated to higher diversity, which is preferable for alleviating premature convergence.

The concurrent use of two performance measures is rather simple and draws ideas from the concept of Pareto dominance in multi-objective optimization. Specifically, after the execution of t_{sec} iterations by all secondary populations (including the

bridging populations), their (AOV, OVSD) pairs,

$$(\bar{f}_{sec}, \sigma_{sec}),$$

are recorded and stored in an external archive. Then, the non-dominated pairs, in terms of Pareto dominance, are identified. The non-dominated pairs are incomparable among them, since they are superior in the one performance criterion but inferior in the other. The final selection can be made either randomly among the non-dominated pairs or with respect to the overall best value of the corresponding secondary populations. The required time complexity for this procedure is negligible since there are only 13 (AOV, OVSD) pairs in the archive. The DEGPOA variant with the two performance measures will be henceforth denoted as eDEGPOA.

The DEGPOA and eDEGPOA algorithms were initially evaluated on the SOCO test suite, following closely the settings previously described for DEGPA. Specifically, a fixed population size $N = 60$ was assumed, and the initial parameter pair in all runs was set to the center of the grid, $(F, CR) = (0.5, 0.5)$. Both approaches used exponential crossover, which was proved to be the best one also in [42,64]. Regarding the rest of the parameters, the values,

$$t_{sec} = 10, \qquad \varepsilon_{\min} = 10^{-2},$$

were used, while t_{pri} was linearly adapted between its minimum and maximum values:

$$t_{pri}^{\min} = 10 \times n, \qquad t_{pri}^{\max} = 14 \times n,$$

according to the suggestions in Section 3.2.1 [64].

The experimental assessment consisted of two phases. In the first phase, full statistical comparisons between DEGPOA, eDEGPOA, and the base algorithms of the SOCO suite were conducted. In the second phase, comparisons of the average performance were made against the rest of the algorithms reported in the test suite. The base algorithms were applied again using the source codes and settings reported in Section 3.3.1. The obtained solution errors were recorder for all algorithms and test problems. The mean and standard deviations of solution values are reported in Tables A.5 and A.6.

The results clearly show that both DEGPOA and eDEGPOA are very competitive

Table 3.3: Number of wins (+), losses (−), and draws (=) of DEGPOA and eDEGPOA against the base algorithms.

		DEGPOA			eDEGPOA		
Dimension	Algorithm	+	−	=	+	−	=
50	DE_{bin}	14	3	2	13	3	3
	DE_{exp}	7	5	7	4	6	9
	CHC	18	0	1	19	0	0
	GCMAES	16	3	0	16	3	0
100	DE_{bin}	17	1	1	17	1	1
	DE_{exp}	7	5	7	5	7	7
	CHC	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0
200	DE_{bin}	17	1	1	17	1	1
	DE_{exp}	8	6	5	8	5	6
	CHC	19	0	0	19	0	0
	GCMAES	16	3	0	14	3	2
500	DE_{bin}	19	0	0	19	0	0
	DE_{exp}	8	7	4	11	6	2
	CHC	19	0	0	19	0	0
	GCMAES	n/a	n/a	n/a	n/a	n/a	n/a

against the base algorithms. They outperformed DE_{bin}, CHC, and GCMAES in almost all test problems and dimensions. Also, they exhibited remarkable competitiveness against DE_{exp}, which was recognized as the best-performing base algorithm in the SOCO suite [42].

In order to statistically verify the observed performance differences with the base algorithms, pairwise Wilcoxon rank-sum tests at confidence level 95% were conducted again for all test functions. Each favorable comparison was counted as a win for the algorithm and denoted as "+". Negative comparisons were counted as losses and denoted as "−", while draws were denoted as "=". Table 3.3 summarizes the number of wins, losses, and draws of DEGPOA and eDEGPOA against the base algorithms.

An interesting observation is that, as dimension increases, both DEGPOA and eDEGPOA exhibit higher number of wins against the best-performing competitor algorithm, namely DE_{exp}. Moreover, the diversity-promoting eDEGPOA variant has marginal differences with DEGPOA, although it improves its performance as dimension increases. This is an indication that the online adaptation mechanism of DEGPOA is capable of preserving diversity even without the use of specialized mechanisms or diversity-oriented performance measures.

The second phase of experimental analysis included comparisons of expected performance against the rest of the SOCO suite's algorithms [42,67]. The comparisons were based on the algorithms' average errors provided in their original sources [67].

Table 3.4: Number of problems where DEGPOA and eDEGPOA exhibited inferior and non-inferior average solution values against different algorithms for the SOCO suite.

	Non-Inferior (Dim.)				Inferior (Dim.)			
	50	100	200	500	50	100	200	500
DEGPOA								
EvoPROpt	17	17	17	16	2	2	2	3
EM323	9	9	10	12	10	10	9	7
SOUPDE	6	6	7	7	13	13	12	12
DE-D^{40}+M^{m}	6	7	8	8	13	12	11	11
GODE	6	6	7	7	13	13	12	12
MA-SSW-Chains	11	13	16	16	8	6	3	3
GaDE	7	7	8	7	12	12	11	12
RPSO-vm	13	13	16	16	6	6	3	3
jDElscop	6	6	7	7	13	13	12	12
SaDE-MMTS	7	7	9	11	12	12	10	8
MOS	7	6	7	7	12	13	12	12
Tuned IPSOLS	11	10	12	12	8	9	7	7
VXQR1	12	13	15	15	7	6	4	4
eDEGPOA								
EvoPROpt	17	17	16	17	2	2	3	2
EM323	9	10	12	16	10	9	7	3
SOUPDE	6	6	8	9	13	13	11	10
DE-D^{40}+M^{m}	7	8	9	11	12	11	10	8
GODE	6	7	8	9	13	12	11	10
MA-SSW-Chains	11	13	16	16	8	6	3	3
GaDE	7	7	8	7	12	12	11	12
RPSO-vm	14	14	16	16	5	5	3	3
jDElscop	6	6	7	9	13	13	12	10
SaDE-MMTS	7	8	9	12	12	11	10	7
MOS	7	7	7	7	12	12	12	12
Tuned IPSOLS	11	9	12	12	8	10	7	7
VXQR1	13	14	15	15	6	5	4	4

Table 3.4 reports the number of test problems were DEGPOA and eDEGPOA exhibited non-inferior or inferior average error values from the rest of the algorithms. Again, it is confirmed that both algorithms have similar non-inferior performance. However, as dimension increases, eDEGPOA has marginally better performance. It worths noting that both algorithms outperform also non-DE algorithms, such as EvoPROpt, MA-SSW-Chains, RPRSO-vm, Tuned IPSOLS and VXQR1, in all dimensions.

Since the selection of mutation operator is a central topic even for low-dimensional problems, the study was extended by applying the proposed DE approaches on the established CEC-2013 suite [69]. Although lower dimensions are also included in the test suite, only the challenging $n = 30$ and $n = 50$ cases were considered. The available computational budget, as dictated by the test suite, was equal to $q = 10000 \times n$, function

Table 3.5: Statistical comparisons between DEGPOA and standard DE on the CEC-2013 test problems.

	Dimension					
	30			50		
$\text{DEGPOA}_{0.2}$ vs	$+$	$-$	$=$	$+$	$-$	$=$
DE_1	16	4	8	15	4	9
DE_2	12	8	8	14	8	6
DE_3	19	3	6	20	6	2
DE_4	12	6	10	12	8	8
DE_5	12	7	9	14	9	5
$\text{DEGPOA}_{0.5}$ vs	$+$	$-$	$=$	$+$	$-$	$=$
DE_1	18	1	9	18	2	8
DE_2	18	5	5	22	3	3
DE_3	17	5	6	17	5	6
DE_4	19	4	4	20	3	5
DE_5	19	5	4	24	3	1
$\text{DEGPOA}_{0.8}$ vs	$+$	$-$	$=$	$+$	$-$	$=$
DE_1	13	6	9	14	9	5
DE_2	15	6	7	18	6	4
DE_3	13	7	8	16	7	5
DE_4	20	4	4	19	5	4
DE_5	20	3	5	22	2	4

"$+$" denotes wins; "$-$" denotes losses; "$=$" denotes ties.

evaluations. The performance criterion for the algorithms was the objective value error defined in Eq. (2.13). Following the CEC-2013 setting [58], fixed population size $N = 60$ was used for the algorithms, and 51 independent experiments were conducted per problem and algorithm. The settings of the test suite were closely followed in order to achieve results comparable to the rest of the algorithms reported in [58, 70].

The initial primary parameter pair of DEGPOA and eDEGPOA was set as previously at the central grid point $(F, CR) = (0.5, 0.5)$, while the initial mutation operator at each experiment was randomly selected from the ones in Eqs. (2.3)-(2.7). These variants are henceforth denoted as $\text{DEGPOA}_{0.5}$ and $\text{eDEGPOA}_{0.5}$, respectively. In addition, the cases of different initial pairs closer to their bound, namely, $(F, CR) = (0.2, 0.2)$, $(F, CR) = (0.8, 0.8)$, were also considered to investigate possible performance fluctuations related to the initial setting. The corresponding algorithms are henceforth denoted as $\text{DEGPOA}_{0.2}$, $\text{eDEGPOA}_{0.2}$, $\text{DEGPOA}_{0.8}$, and $\text{eDEGPOA}_{0.8}$, respectively. In all cases, exponential crossover was used according to the previous analysis. Also, the values $t_{sec} = 5$, and $\varepsilon_{\min} = 10^{-2}$ (optional parameter) were adopted [64], along with the linearly increasing t_{pri} in the range $[10 \times n, 14 \times n]$.

Table 3.6: Statistical comparisons between eDEGPOA and standard DE on the CEC-2013 test problems.

	Dimension					
	30			50		
$\text{eDEGPOA}_{0.2}$ vs	+	−	=	+	−	=
DE_1	14	2	12	21	2	5
DE_2	12	7	9	20	4	4
DE_3	20	3	5	21	6	1
DE_4	13	7	8	20	2	6
DE_5	12	9	7	21	4	3
$\text{eDEGPOA}_{0.5}$ vs	+	−	=	+	−	=
DE_1	16	1	11	15	1	12
DE_2	18	4	6	20	3	5
DE_3	15	3	10	14	5	9
DE_4	21	3	4	18	3	7
DE_5	20	5	3	23	3	2
$\text{eDEGPOA}_{0.8}$ vs	+	−	=	+	−	=
DE_1	15	2	11	19	1	8
DE_2	19	1	8	21	1	6
DE_3	16	2	10	17	1	10
DE_4	25	1	2	25	1	2
DE_5	23	1	4	26	1	1

"+" denotes wins; "−" denotes losses; "=" denotes ties.

All DEGPOA and eDEGPOA variants were applied on the CEC-2013 test suite according to the aforementioned settings. Their results were recorded and statistically analyzed in order to facilitate comparisons with a number of adaptive and non-adaptive algorithms. The results of the competitor algorithms were adopted directly from the relevant sources [59]. It shall be noted that these results refer to already tuned versions of the competitor algorithms. On the other hand, DEGPOA and eDEGPOA do not apply any preprocessing procedure or preliminary experimentation. Moreover, the computational budget for the CEC-2013 problems is quite restrictive. This renders the benchmarking even more challenging.

The performance comparisons were based on Wilcoxon rank-sum tests at confidence level 95% of the achieved solution errors between DEGPOA, eDEGPOA, and the following ten algorithms: SMADE [71], TLBSaDE [72], JANDE [73], DE_APC [74], TPC-GA [75], PVADE [76], CDASA [77], and PLES [78]. For completeness purpose, all standard variants of DE with mutation operators reported in Eqs. (2.3)-(2.7) were also included in the competitor algorithms. For each comparison, a win was counted for DEGPOA or eDEGPOA whenever it achieved statistically superior performance than the competitor algorithm. In the opposite case, a loss was counted. Statistically

insignificant differences between algorithms were considered as ties.

Tables 3.5 and 3.6 report the number of wins, losses, and ties for DEGPOA and eDEGPOA, respectively, against the corresponding standard DE algorithms. Note that each standard DE algorithm adopted the same parameter values as the initial parameter setting of the competing DEGPOA/eDEGPOA approach. As we can see in Table 3.5, the DEGPOA approaches outperformed all standard DE algorithms regardless of the initial parameter setting. This indicates that the observed improvements from the use of the grid-based parameter adaptation are not highly affected by the initial parameter setting. Instead, DEGPOA was capable of tuning the algorithm regardless of the initial parameters and operator, achieving far better results than the corresponding DE algorithm with the same configuration.

Interestingly, we can notice that the average number of wins for DEGPOA increases with dimension. Indeed, for the 50-dimensional problems the average numbers of wins for the three DEGPOA algorithms was equal to 15.0, 20.2, and 17.8, while the corresponding numbers for the 30-dimensional problems were 14.2, 18.2, and 16.2. This evidence suggests that the search stagnation of DE in higher dimensions can be ameliorated through the proposed parameter tuning. Similar results were obtained for eDEGPOA, with average numbers of wins equal to 14.2, 18.0, and 19.6 for the 30-dimensional problems, and 20.6, 18.0, and 21.6, for the 50-dimensional cases. All average numbers of wins are graphically illustrated in Fig. 3.8.

In the same vein, Tables 3.7 and 3.8 report results from statistical comparisons of DEGPOA and eDEGPOA with other algorithms. Among them are included top-performing algorithms for the CEC-2013 test suite, with their parameters being already tuned. Especially the (e)DEGPOA$_{0.2}$ and (e)DEGPOA$_{0.5}$ approaches were able to achieve same or higher number of wins than half or more of the rest of the algorithms, with better results being achieved in problem of higher dimension. Indeed, for the three DEGPOA algorithms the average numbers of wins were equal to 10.9, 11.5, and 8.0, in the 30-dimensional problems, while the corresponding numbers for the 50-dimensional case were 12.5, 13.1, and 9.4. Similarly, eDEGPOA variants achieved 10.6, 11.2, and 9.6 wins on average in the 30-dimensional case, and 11.5, 11.6, and 10.8 wins on average for the 50-dimensional case. For completeness purposes, the average solution errors achieved by the best-performing DEGPOA$_{0.5}$ and eDEGPOA$_{0.5}$ approaches are reported in Tables A.7-A.9.

Finally, the six variants of the proposed approach were compared among them.

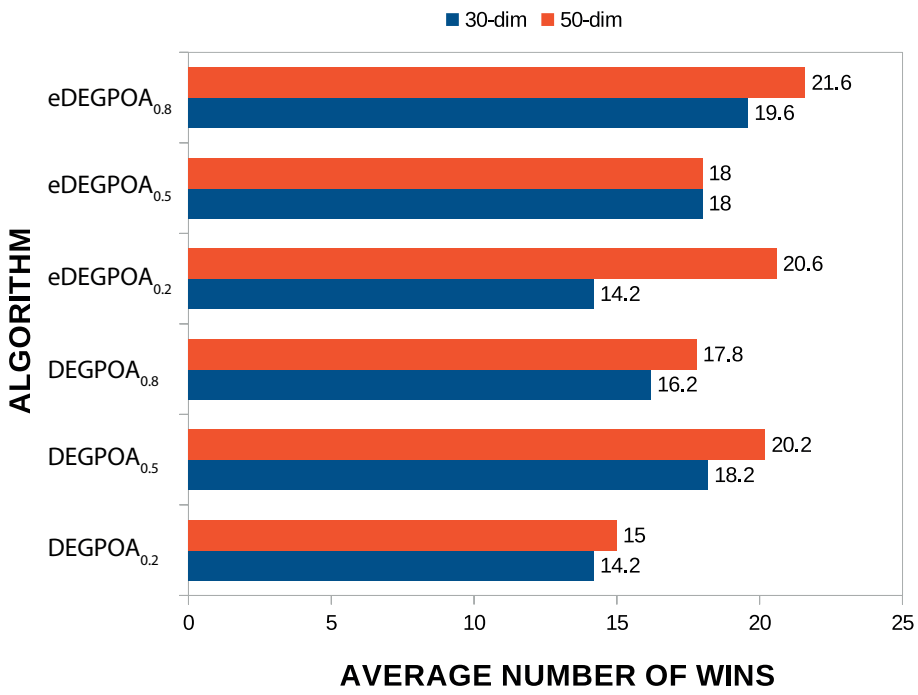


Figure 3.8: Average number of wins of DEGPOA and eDEGPOA against the standard DE algorithms.

For visualization purpose, a *rank* was calculated for each algorithm, defined as the difference between its total number of wins, w_{alg}, and its total number of losses, l_{alg}, achieved in all comparisons, i.e.,

$$\texttt{rank}(\text{alg}) = w_{\text{alg}} - l_{\text{alg}}.$$

Since each algorithm is compared to five other algorithms on 28 test problems, it holds that

$$0 \leqslant w_{\text{alg}}, l_{\text{alg}} \leqslant 140,$$

and, hence,

$$-140 \leqslant \texttt{rank}(\text{alg}) \leqslant 140.$$

Figure 3.9 illustrates the ranks for all DEGPOA and eDEGPOA algorithms, where we can see that $\text{DEGPOA}_{0.5}$ and $\text{eDEGPOA}_{0.5}$ clearly dominated the rest. This is a direct consequence of their better initial parameter pair, $(F, CR) = (0.5, 0.5)$, which proved to be better than the other two (distant) initial pairs. Note that the standard DE with these parameters exhibited inferior performance than all DEGPOA and eDEGPOA approaches. This implies that, even under defective initial parameters, the grid-based

Table 3.7: Statistical comparisons between DEGPOA and other algorithms on the CEC-2013 test problems.

	Dimension					
	30			50		
$DEGPOA_{0.2}$ vs	+	−	=	+	−	=
SMADE	6	16	6	9	15	4
TLBSaDE	8	13	7	11	14	3
JANDE	6	18	4	9	17	2
DE_APC	11	12	5	12	12	4
TPC-GA	11	9	8	13	10	5
PVADE	10	14	4	13	13	2
CDASA	13	8	7	11	12	5
PLES	22	1	5	22	4	2
$DEGPOA_{0.5}$ vs	+	−	=	+	−	=
SMADE	7	15	6	9	15	4
TLBSaDE	9	12	7	9	13	6
JANDE	6	17	5	13	13	2
DE_APC	11	12	5	11	12	5
TPC-GA	11	9	8	13	9	6
PVADE	11	12	5	15	11	2
CDASA	14	8	6	14	11	3
PLES	23	1	4	21	2	5
$DEGPOA_{0.8}$ vs	+	−	=	+	−	=
SMADE	5	21	2	5	19	4
TLBSaDE	6	17	5	8	18	2
JANDE	0	21	7	4	16	8
DE_APC	10	15	3	10	15	3
TPC-GA	10	15	3	12	12	4
PVADE	8	15	5	10	14	4
CDASA	9	12	7	9	13	6
PLES	16	7	5	17	8	3

"+" denotes wins; "−" denotes losses; "=" denotes ties.

adaptation method is highly beneficial for the algorithm. On the other hand, when starting from a favorable parameter pair the proposed approach can significantly boost performance.

Summarizing the experimental findings, although DEGPOA and eDEGPOA are more suitable for computationally demanding high-dimensional problems, they both exhibited very competitive performance against some of the most competitive adaptive DE approaches such as SMADE, TLBSaDE, and JANDE. Moreover, all DEGPOA and eDEGPOA approaches proved to be competitive against other DE-based approaches such as DE_APC and PVADE, as well as against different algorithms such as TPC_GA, CDASA, and PLES. This is a very promising result given that the proposed approaches were assigned only the computational budget specified by the test suite with no additional preprocessing or preliminary experimentation. Moreover, previous observations

Table 3.8: Statistical comparisons between eDEGPOA and other algorithms on the CEC-2013 test problems.

	Dimension					
	30			50		
eDEGPOA$_{0.2}$ vs	+	−	=	+	−	=
SMADE	6	16	6	7	15	6
TLBSaDE	8	13	7	11	13	4
JANDE	5	18	5	8	16	4
DE_APC	11	12	5	12	12	4
TPC-GA	11	10	7	12	10	6
PVADE	10	15	3	11	13	4
CDASA	12	9	7	11	13	4
PLES	22	2	4	20	5	3
eDEGPOA$_{0.5}$ vs	+	−	=	+	−	=
SMADE	6	17	5	9	15	4
TLBSaDE	8	14	6	9	13	6
JANDE	6	18	4	12	13	3
DE_APC	11	11	6	11	12	5
TPC-GA	11	10	7	13	9	6
PVADE	11	14	3	15	12	1
CDASA	15	9	4	13	12	3
PLES	22	1	5	21	3	4
eDEGPOA$_{0.8}$ vs	+	−	=	+	−	=
SMADE	5	18	5	7	16	5
TLBSaDE	6	15	7	8	16	4
JANDE	4	19	5	7	17	4
DE_APC	10	13	5	11	13	4
TPC-GA	10	11	7	13	11	4
PVADE	11	12	5	12	10	6
CDASA	12	9	7	11	11	6
PLES	19	5	4	17	5	6

"+" denotes wins; "−" denotes losses; "=" denotes ties.

regarding performance improvement as dimension increases [64, 68] were also verified for the CEC-2013 test suite. This is observed especially for the 50-dimensional test problems regardless of the initial parameter pair.

3.4 Preliminary Sensitivity Analysis

Similarly to all parameter adaptation methods, GPAM has a few user-defined parameters. In this section, a preliminary sensitivity analysis of the method on its parameters is offered [79]. For this purpose, the DEGPOA approach and the established CEC-2013 suite were used, which offers an abundant variety of test problems. Several levels of DEGPOA's parameters are considered and their impact on the algorithm's

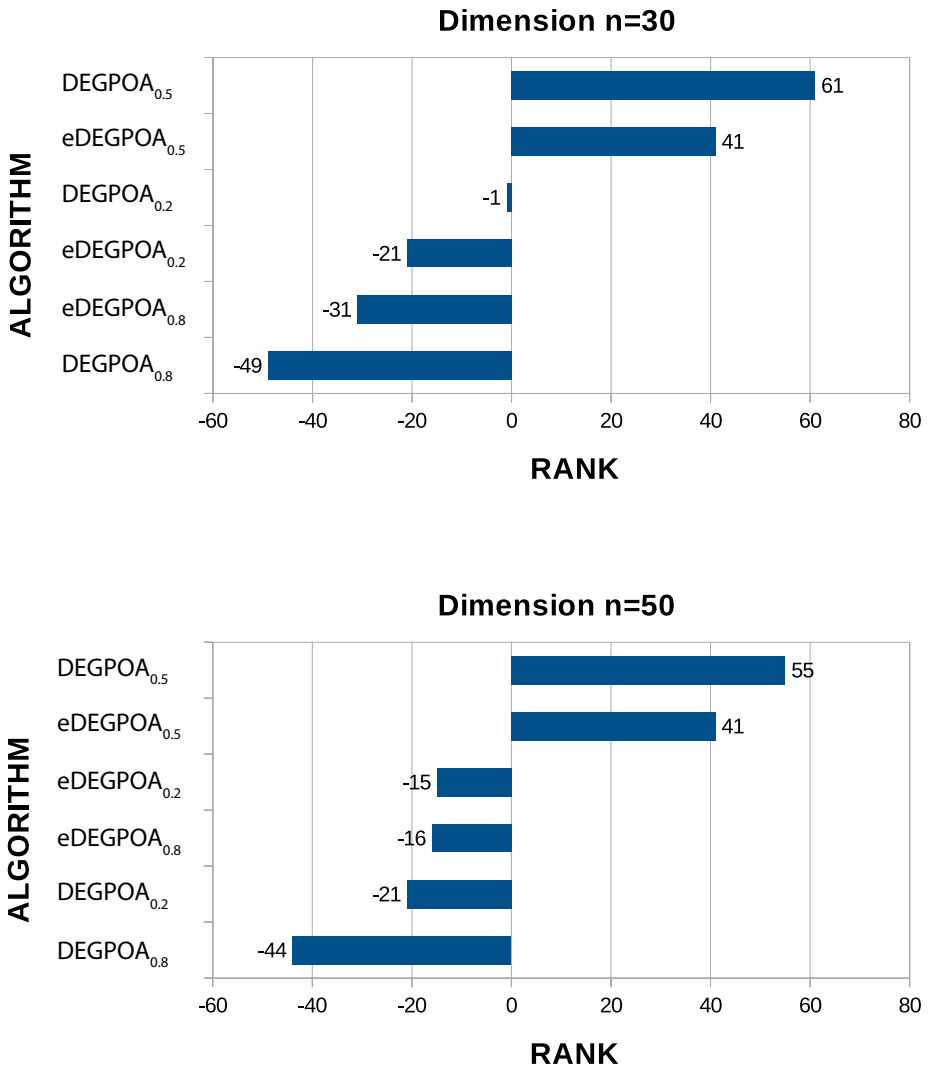


Figure 3.9: Ranks of the proposed algorithms in comparisons among them for the 30-dimensional case (upper figure) and the 50-dimensional case (lower figure).

performance is statistically analyzed, offering interesting conclusions.

Regarding the grid search parameters, the following values were adopted as default choices in previous experiments:

$$t_{sec} = 5, \quad t_{pri} = 10 \times n, \quad \lambda = \lambda_F = \lambda_{CR} = 0.1, \tag{3.9}$$

while the initial parameter vector was placed at the center of the grid, i.e., $(F, CR) = (0.5, 0.5)$, and the initial mutation operator was randomly selected from the ones in Eqs. (2.3)-(2.7). This DEGPOA variant is henceforth denoted as $\text{DEGPOA}_{\text{base}}$.

For the sensitivity study, we define the following sets for the three parameters:

$$W_{t_{sec}} = \{5, 10, 15, 20\}, \quad W_{t_{pri}} = \{5\times n, 10\times n, 15\times n, 20\times n\}, \quad W_{\lambda} = \{0.05, 0.1, 0.15, 0.2\}.$$

The values of Eq. (3.9) were considered as the baseline for assessing the new settings. DEGPOA was validated by changing one of its parameters to a different level from the sets above, while keeping the rest of the parameters fixed to the baseline values. This results in 12 new DEGPOA instances.

All experiments were conducted on the CEC-2013 test suite for the common cases $n = 10$ and $n = 30$, following the guidelines of the test suite. In order to avoid bias imposed by the initial parameters, the central parameter $(F, CR) = (0.5, 0.5)$ was used in all cases.

Henceforth, the $\text{DEGPOA}_{\text{base}}$ is the baseline version of the algorithm, and the rest are denoted with corresponding subscripts that reveal the parameter setting. For example, the instance with $t_s = 5$, $t_p = 5 \times n$, and $\lambda = 0.05$ is denoted as $\text{DEGPOA}_{5s\_5p\_0.05\lambda}$.

The considered DEGPOA instances were compared among them using Wilcoxon rank-sum tests at confidence level 95%, in terms of solution error. For each comparison of a new instance with the baseline variant, a win was counted if it achieved statistically superior performance than the baseline approach. In the opposite case, a loss was counted, while statistically insignificant differences between algorithms were considered as ties.

Table 3.9 reports the number of wins, losses, and ties of the new DEGPOA instances against $\text{DEGPOA}_{\text{base}}$. The fourth column denoted as "W-L" stands for the difference between the number of wins and loses, which provides a general performance trend of algorithm against the baseline. High positive values correspond to an instance that has far better performance than the baseline, while negative values imply inferior performance of the new instance. The next column denoted as ID reports the index value

$$ID = 28 + (W - L),$$

which characterizes the relevant performance of the corresponding DEGPOA instance against the baseline over all 28 test problems. The last column of the table denoted as NI is the normalized index,

$$NI = \frac{\text{ID}}{28}, \tag{3.10}$$

Table 3.9: Comparisons of new DEGPOA instances with $\text{DEGPOA}_{\text{base}}$.

	n	W	L	D	W-L	ID	NI
t_s modified							
$\text{DEGPOA}_{5s\_10p\_0.1\lambda}$	10	0	0	28	0	28	1.00
$\text{DEGPOA}_{5s\_10p\_0.1\lambda}$	30	0	0	28	0	28	1.00
$\text{DEGPOA}_{10s\_10p\_0.1\lambda}$	10	1	5	22	-4	24	0.86
$\text{DEGPOA}_{10s\_10p\_0.1\lambda}$	30	1	2	25	-1	27	0.96
$\text{DEGPOA}_{15s\_10p\_0.1\lambda}$	10	0	11	17	-11	17	0.61
$\text{DEGPOA}_{15s\_10p\_0.1\lambda}$	30	2	3	23	-1	27	0.96
$\text{DEGPOA}_{20s\_10p\_0.1\lambda}$	10	1	12	15	-11	17	0.61
$\text{DEGPOA}_{20s\_10p\_0.1\lambda}$	30	4	7	17	-3	25	0.89
			absolute sum:		31		
t_p modified							
$\text{DEGPOA}_{5s\_5p\_0.1\lambda}$	10	0	4	24	-4	24	0.86
$\text{DEGPOA}_{5s\_5p\_0.1\lambda}$	30	0	2	26	-2	26	0.93
$\text{DEGPOA}_{5s\_10p\_0.1\lambda}$	10	0	0	28	0	28	1.00
$\text{DEGPOA}_{5s\_10p\_0.1\lambda}$	30	0	0	28	0	28	1.00
$\text{DEGPOA}_{5s\_15p\_0.1\lambda}$	10	1	1	26	0	28	1.00
$\text{DEGPOA}_{5s\_15p\_0.1\lambda}$	30	1	0	27	1	29	1.04
$\text{DEGPOA}_{5s\_20p\_0.1\lambda}$	10	2	1	25	1	29	1.04
$\text{DEGPOA}_{5s\_20p\_0.1\lambda}$	30	3	1	24	2	30	1.07
			absolute sum:		10		
λ modified							
$\text{DEGPOA}_{5s\_10p\_0.05\lambda}$	10	1	3	24	-2	26	0.93
$\text{DEGPOA}_{5s\_10p\_0.05\lambda}$	30	1	2	25	-1	27	0.96
$\text{DEGPOA}_{5s\_10p\_0.1\lambda}$	10	0	0	28	0	28	1.00
$\text{DEGPOA}_{5s\_10p\_0.1\lambda}$	30	0	0	28	0	28	1.00
$\text{DEGPOA}_{5s\_10p\_0.15\lambda}$	10	0	2	26	-2	26	0.93
$\text{DEGPOA}_{5s\_10p\_0.15\lambda}$	30	2	0	26	2	30	1.07
$\text{DEGPOA}_{5s\_10p\_0.2\lambda}$	10	0	6	22	-6	22	0.79
$\text{DEGPOA}_{5s\_10p\_0.2\lambda}$	30	4	7	17	-3	25	0.89
			absolute sum:		16		

which offers a straightforward comparison measure between the competing algorithms (rounded to 2 decimal digits). Obviously, $NI = 1.00$ when the two compared algorithms have statistically equivalent performance (only ties in statistical tests), while $NI > 1.00$ holds whenever the new DEGPOA instance is superior than the baseline, and $NI < 1.00$ when it is inferior. Since $0 \leqslant ID \leqslant 56$, the normalized index is bounded in $0 \leqslant NI \leqslant 2$.

In order to facilitate comparisons, Fig. 3.10 illustrates NI for the different parameter levels and dimensions. The gray bar stands for the performance of $\text{DEGPOA}_{\text{base}}$, while the blue bars refer to the corresponding new instances. The figures offer some interesting conclusions. Firstly, we can see that t_{sec} can have significant impact on the algorithm's performance in lower dimension ($n = 10$) as illustrated in Fig. 3.10(a). Specifically, smaller values of t_{sec} offer better overall performance, which implies that

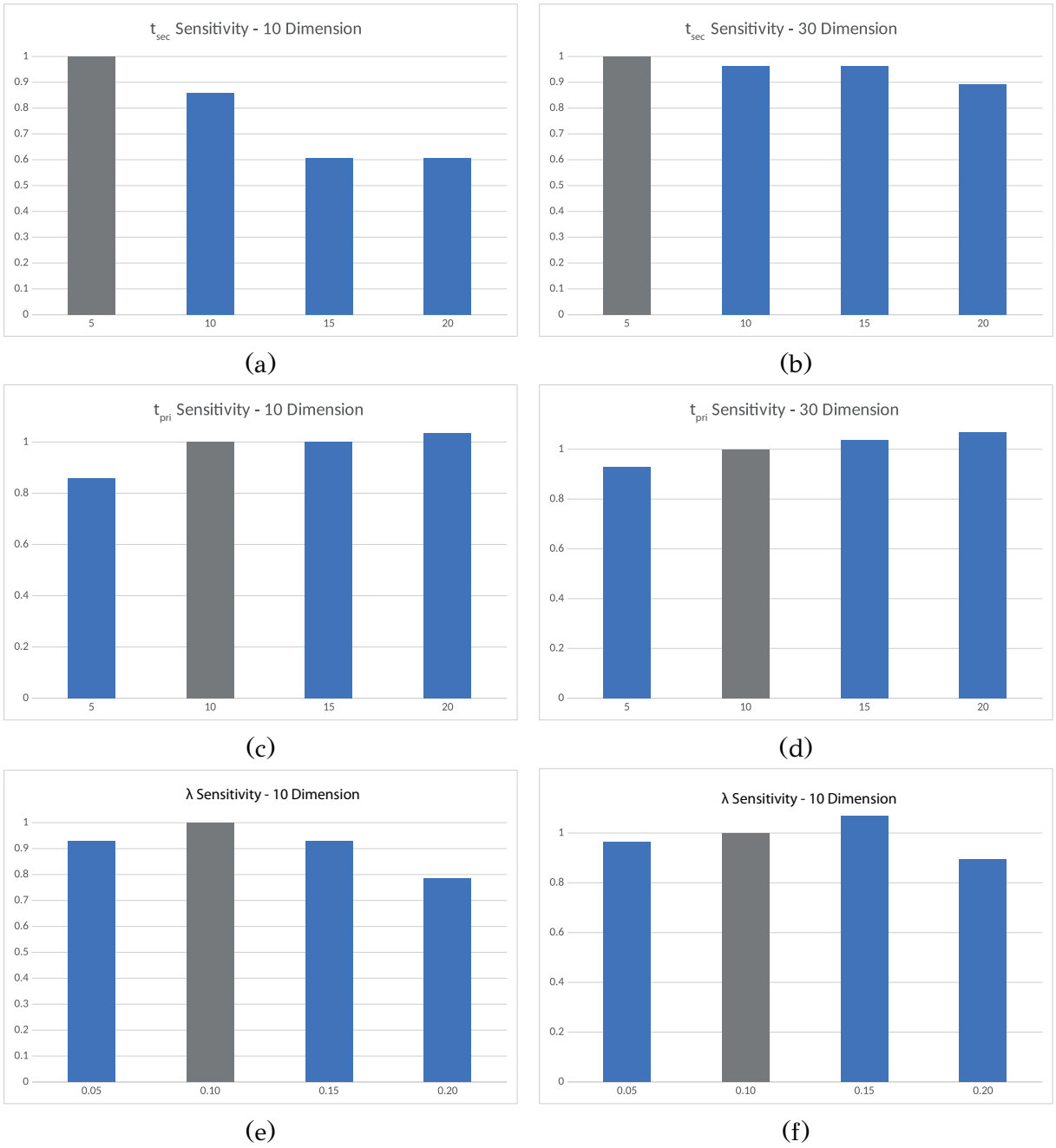


Figure 3.10: Values of the normalized index NI of Eq. (3.10) per dimension for the parameters t_s (cases (a) and (b)), t_p (cases (c) and (d)), and λ (cases (e) and (f)).

the estimations of the secondary populations are adequately accurate, sparing computational budget for the dynamic's deployment phase. On the other hand, in the high-dimensional case ($n = 30$) depicted in Fig. 3.10(b), this effect becomes milder as a direct consequence of the increased complexity of the problems, which requires longer estimation runs. Nevertheless, the value $t_{sec} = 5$ that was used in previous works [64] verifies its superiority for the specific dimensions.

Regarding the parameter t_{pri}, as we see in Figs 3.10(c) and 3.10(d), values lower than $10 \times n$ produce inferior performance, implying that the number is inadequate to reveal the primary population and parameters' dynamic. Instead, higher values are beneficial especially for the high-dimensional case. However, the effect remains bounded within 10% of the corresponding baseline value even after doubling the value of t_{pri}. This indicates that the effect of t_{pri} is not very significant for the algorithm's performance if the estimation evaluations t_{sec} retain a proper value. Recall that in all experiments for different t_{pri} values, the default value $t_{sec} = 5$ was used.

For both t_{sec} and t_{pri}, the performance trend (improving or worsening) was observed for both dimensions. However, this is not the case for the third parameter λ. Changing the discretization step from 0.1 to either lower or higher values produces inferior performance in the 10-dimensional case as illustrated in Fig. 3.10(e). This motif changes in the high-dimensional case as illustrated in Fig. 3.10(f), where slightly increasing λ to 0.15 improves performance up to 7%, while different values produce inferior performance of comparable magnitude. Notice that λ determines the search accuracy in the parameter space and has actual dependence both on the algorithm as well as on the problem at hand. Thus, there is no clear explanation for this behavior, which is probably the outcome of the interplay between the algorithm's dynamic with the specific parameters and the complexity of the problem itself.

The results show that DEGPOA can achieve stable performance under mild perturbations of the proposed default parameters. In order to identify the overall most influential parameter for all DEGPOA instances, the sum of the absolute differences $W - L$ for each parameter were considered, as they are reported in Table 3.9. Then, these three values were normalized by dividing with their sum, and the percentages that are graphically represented in Fig 3.11 were received. Each normalized value shows the participation of the corresponding parameter in the observed differences. The blue color refers to the t_{sec} parameter, which proves to be the most influential one, followed by λ and t_{pri}.

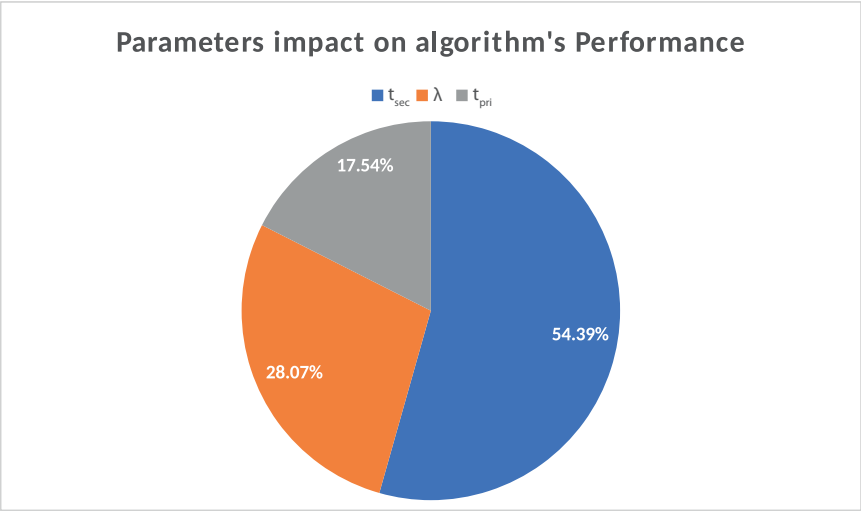


Figure 3.11: Overall impact of the parameters on algorithm's performance.

3.5 Application on Particle Swarm Optimization

The proposed method is further demonstrated on the state-of-the-art PSO algorithm [80]. The experiments were conducted using the high-dimensional SOCO suite. GPAM is used to dynamically adapt the scalar parameters w, c_1, c_2, and the neighborhood radius m of the lbest PSO model during its run. The proposed approach is henceforth called *Particle Swarm Optimization with Parameter and Neighborhood Adaptation* (PSOPNA).

Initially, the search space of the scalar parameters of PSO defined. The ranges,

$$w \in (0, 1], \qquad c_1, c_2 \in (0, 3],$$

commonly used in various applications are adopted from the relevant literature. These ranges are discretized with predefined step sizes,

$$\lambda_w = \lambda_{c_1} = \lambda_{c_2} = 0.1. \tag{3.11}$$

Also, a set of possible (integer) values for the neighborhood radius m is defined. In the current case, we consider,

$$m \in \{1, 3, 5, 7, 9\}.$$

The GPAM* approach is used to handle m. Specifically, for each value of m, a 3-

dimensional grid parameter space is defined as follows,

$$\mathcal{G}_m = \{(w, c_1, c_2);\ w \in \{0.0, 0.1, \ldots, 1.0\}, c_1, c_2 \in \{0.0, 0.1, 0.2, \ldots, 3.0\}\}\,.$$

The grid's density can be increased by decreasing the step sizes in Eq. (3.11) if more fine-grained parameter search is desirable. Each triplet (w, c_1, c_2) in the interior of $\mathcal{G}_m$ has 6 immediate neighboring points along the 3 orthogonal directional axes.

The algorithm starts by randomly initializing a swarm, called the *primary swarm* and denoted as S_{pri}, assuming an initial neighborhood radius m and a set of parameters $(w, c_1, c_2) \in \mathcal{G}_m$. A reasonable initial choice is,

$$m = 1, \quad (w, c_1, c_2) = (0.5, 1.5, 1.5),$$

but the algorithm works also for different choices as it will be shown later. The primary swarm is evolved for $t_{pri} = 10 \times n$ iterations, following the setting of Section 3.2.1. Then, the three main phases of GPAM* take place, namely *cloning*, *performance estimation*, and *dynamic deployment*.

The primary swarm, along with its best positions, is copied into 7 *secondary swarms*, $S_1, \ldots, S_7$, of same neighborhood radius m, each one assuming different scalar parameters as follows,

$$\begin{aligned} w' = w + s_w\,\lambda_w, \quad c_1' = c_1 + s_1\,\lambda_{c_1}, \quad c_2' = c_2 + s_2\,\lambda_{c_2}, \\ s_w, s_1, s_2 \in \{-1, 0, 1\}, \qquad |s_w| + |s_1| + |s_2| = 1 \text{ or } 0. \end{aligned} \tag{3.12}$$

Obviously, the case $s_w = s_1 = s_2 = 0$ corresponds to the parameter setting of the primary swarm. Besides the 7 secondary swarms of same radius m, four additional secondary (bridging) swarms, $S_8, \ldots, S_{11}$, are considered, adopting the scalar parameters (w, c_1, c_2) of the primary swarm but for different neighborhood radius values, i.e., for $m = 3, 5, 7$, and 9. Intuitively, these secondary swarms define the bridges from the 3-dimensional grid of $m = 1$ to the other grids, aiming at identifying a more beneficial neighborhood radius for the algorithm.

All the 11 secondary swarms are then evolved for a small number of iterations, $t_{sec} \ll t_{pri}$, according to the standard PSO procedure, updating also their best positions. These short runs reveal performance trends of the secondary swarms with their assigned parameter settings. For time-efficiency purpose, the short runs can be ex-

tended in parallel by evoking a separate thread for each individual secondary swarm, thereby taking full advantage of modern multi-core computer systems. Following the guidelines provided in previous sections, typical values for t_{sec} lie between 5 and 10 iterations.

After the short runs, for each secondary swarm S_j the AOV performance measure is computed on the best positions along with the OVSD defined in Eqs. (3.3) and (3.8), respectively. The obtained performance pairs,

$$(\bar{f}_j, \sigma_j), \qquad j = 1, \ldots, 11,$$

are then compared in terms of Pareto dominance and the non-dominated ones are selected. In order to further decide on one among the non-dominated secondary swarms but also reduce the possibility of being misled due to temporarily extremal (high or low) objective values, an additional diversity-oriented performance measure computed for each non-dominated swarm can be considered.

Specifically, the secondary swarm that has the highest *interquartile range* (IQR) is selected. IQR is a common statistical measure of variability, based on the division of a data set into four equal quartiles. The data is sorted and the IQR is defined as,

$$IQR = Q3 - Q1,$$

where $Q1$ and $Q3$ specify the 1st and 3rd quartile of the data, respectively. In our case, IQR is computed on the best positions' values $f(p_i)$ of the non-dominated secondary swarms. The secondary swarm with the highest IQR value is then selected in order to retain search diversity.

The selected secondary swarm along with all its parameters becomes the primary swarm, replacing the existing one. In order to make complete use of newly detected best positions in the short runs, the overall bests of all the unselected secondary swarms are also inserted into the new primary swarm, replacing equal number of worst individuals. The new primary swarm is then evolved for t_{pri} iterations to fully exploit the new parametrization.

The experimental evaluation of PSOPNA was conducted on the SOCO test suite for dimensions $n = 50$, 100, 200, and 500, following the guidelines of the test suite as in previous experiments with DE. Besides, the central point of the grid that was used as the initial parameter vector, two extremal initial vectors, closer to the boundaries

Table 3.10: Statistical comparisons of PSOPNA against plain PSO in SOCO suite.

		$\text{PSOPNA}_{0.5/1.5}$			$\text{PSOPNA}_{0.2/1.0}$			$\text{PSOPNA}_{0.8/2.0}$		
Dimension	Algorithm	$+$	$-$	$=$	$+$	$-$	$=$	$+$	$-$	$=$
50	PSO	7	6	6	15	0	4	12	3	4
100	PSO	8	6	5	14	1	4	16	3	0
200	PSO	8	3	8	15	1	3	17	2	0
500	PSO	8	7	4	8	3	8	16	2	1

"+" denotes wins, "−" denotes losses, and "=" denotes ties

Table 3.11: Statistical comparisons of $\text{PSOPNA}_{0.5/1.5}$ against the base algorithms in SOCO suite.

	Dimension											
	50			100			200			500		
	$+$	$-$	$=$	$+$	$-$	$=$	$+$	$-$	$=$	$+$	$-$	$=$
DE_{bin}	6	12	1	6	10	3	7	8	4	9	5	5
DE_{exp}	3	13	3	2	17	0	1	15	3	6	12	1
CHC	10	4	5	11	4	4	9	5	5	17	1	1
GCMAES	14	3	2	13	4	2	13	5	1	n/a	n/a	n/a

"+" denotes wins, "−" denotes losses, and "=" denotes ties

of the grid, were also considered:

$$(w, c_1, c_2) = (0.2, 1.0, 1.0), \qquad (w, c_1, c_2) = (0.8, 2.0, 2.0).$$

The corresponding PSOPNA instances for the three initial parameter settings are henceforth denoted as $\text{PSOPNA}_{0.5/1.5}$, $\text{PSOPNA}_{0.2/1.0}$, and $\text{PSOPNA}_{0.8/2.0}$.

The experimental assessment consisted of two phases. In the first phase, the three PSOPNA instances were compared against their PSO counterparts with the corresponding (fixed) parameter sets and neighborhood radius $m = 1$. Wilcoxon rank-sum tests were conducted for each pair of algorithms at confidence level 95% for all test problems. Similarly to previous analysis, a favorable comparison was counted as a win for the PSOPNA approach and denoted with "+". Respectively, negative comparisons are denoted with "−", and ties (no statistical difference between the algorithms) are denoted as "=".

Table 3.10 summarizes the results. We can see that for the two extremal initial points PSOPNA dramatically improved the performance over the corresponding PSO approaches. Note that the improvement was achieved without any additional prepro-

cessing or preliminary experimentation. Even for the case of the near-optimal initial parameter setting, the proposed approach achieved better or equivalent performance in more than 60% of the problems. Also, we can see that increasing the dimension from 50 to 500 does not radically change the observed performance, which is a trait of nice scaling properties.

Table 3.11 reports the results of the second experimentation phase where the $\text{PSOPNA}_{0.5/1.5}$ version is compared against the base algorithms of the test suite. The specific PSOPNA approach was considered as it constitutes a common parameter setting. PSOPNA was competitive against two of the base algorithms, namely CHC and GCMAES, while it was outperformed by the DE variants. However, it shall be noted that plain PSO was completely out of competition against all base algorithms, while the dominant DE algorithm was shown to be the best one for the specific test suite [67].

Moreover, it shall be emphasized that all base algorithms assumed their tuned parameter settings for the considered test suite, while PSOPNA did not spend any function evaluations on fine-tuning. Thus, in a completely fair comparison, PSOPNA should be receiving also the additional computational budget that is spent by the rest of the algorithms for their laborious fine-tuning. For completeness purpose, the achieved averages and standard deviations of the obtained solution values of PSOPNA and the base algorithms are reported in Tables A.10-A.11 for all test problems.

3.6 Synopsis

In this chapter, a general online parameter adaptation method based on grid search was introduced. The verification of the proposed method was established on two state-of-the-art metaheuristics, namely DE and PSO, attaining competitive performance against other methods. DE was selected due to its known sensitivity on parameter values that rendered their dynamic adaptation a challenging task, while PSO was selected in order to demonstrate the method's applicability on similar algorithms. The efficiency of the proposed method was established on the two widely used test suites that include both low-dimensional and high-dimensional problems, with significant results. The results suggested that the proposed approaches can relieve the user from the burden of parameter setting without loss in average performance. Furthermore,

the proposed grid-based adaptation can be straightforwardly parallelized resulting in significant gain in terms of running time.

A number of different performance criteria were also proposed in this chapter in order to promote diversity and deter premature convergence of the grid-based method. Similarly to other methods, the grid search has also a few (mostly optional) user-defined parameters. In the last part of this chapter, a preliminary sensitivity study of these parameters were examined. The results verify experimental evidence regarding the method's tolerance on its parameters as well as the efficiency of the default proposed parameters.

Chapter 4

New Gradient-Based Parameter Adaptation Method with Line Search

This chapter presents an enhanced dynamic parameter adaptation method, motivated by the grid-based approach presented in the previous chapter. It is based on estimations of the algorithm's performance using approximate gradients in the parameter space and line search. The new method offers significant advantages compared to the grid-based method and it is demonstrated on the two test suites previously used.

4.1 Introduction

A new general-purpose online parameter adaptation method that refines and improves the grid-based approach is introduced. Its core mechanism lies in the replacement of the grid search presented in Chapter 3 with an approximate gradient

search with line search in the parameter domain. This approach offers two significant advantages against the grid. First, the search mechanism is allowed to perform informative large steps in the parameter domain based on the line search procedure. Second, the scalar parameters are not confined in a discretized subset of their domain, but they can assume continuous values. Moreover, this approach has inherent parallelization properties. The proposed method is demonstrated on DE [9] on the SOCO and CEC-2013 suites, as well as on PSO [11].

4.2 Proposed Method

The proposed approach, henceforth called *Gradient-based Parameter Adaptation with Line Search* (GPALS) [81], draws inspiration from the GPAM parameter adaptation method GPAM proposed in Chapter 3. In that case, the scalar parameter domain is discretized forming a grid. In the general population-based algorithm model the grid consists of all scalar parameter vectors $(\rho_1, \rho_2, \ldots, \rho_{np})$ in their corresponding domains, using a fixed discretization step $\lambda > 0$.

A limitation of the grid-based method is the constant step size λ for the discretization of the parameter space. If λ is small and better parameter values exist in a specific direction in the parameter space, GPAM may need a large number of steps to reach them. Conversely, if λ is too large then promising values may be overshot. Also, the search directions are always restricted to the main axes directions and diagonal moves. These issues motivated the development of the proposed GPALS approach, where the grid is replaced by a dense parameter search space. In this framework, trajectories of scalar parameter vectors are produced by following approximate gradient directions in the parameter domain, while line search is used to determine a suitable step size.

These procedures require estimations of the algorithm's performance under different parameter settings. Similarly to GPAM, the estimations are based on short runs of the algorithm under the corresponding parameter vectors. The runs can be conducted either serially or in parallel. The main scheme of GPALS can be generalized to any algorithm and arbitrary number of parameters.

The general population-based algorithm presented in Section 2.3 is used as an example to describe a complete cycle of the proposed GPALS method. Consider the

general optimization problem of Eq. (2.1), let N be the population size, and $[l_{\rho_i}, u_{\rho_i}]$ be prescribed ranges for the parameters $\rho_1, \rho_2, \ldots, \rho_{np}$, respectively. Then, the parameter domain is defined as

$$G = [l_{\rho_1}, u_{\rho_1}] \times [l_{\rho_2}, u_{\rho_2}] \times \cdots \times \left[l_{\rho_{np}}, u_{\rho_{np}}\right].$$

GPALS assumes a *primary population* P_{pri}, sampled in the search space X of the problem at hand. The primary population is assigned an initial parameter vector $\bar{\rho}_0 = (\rho_{1,0}, \rho_{2,0}, \ldots, \rho_{np,0})$, which can be either the central point of G or a randomly selected one. Naturally, if additional information is available regarding the most promising parameter values (e.g., due to previous experimentation), the initial vector can be properly adjusted.

After initialization, the primary population is evolved for t_{pri} iterations. According to the presentation in the previous chapter, the default choice:

$$t_{pri} = 10 \times n,$$

is suggested, where n is the dimension of the optimization problem. In general, t_{pri} shall be adequate for the algorithm to deploy its dynamic, but not too large in order to avoid rapidly consuming the available computational budget.

For the evolved primary population with the current parameter vector $\bar{\rho}_c$, the AOV performance measure of Eq. (3.3) is computed. Then, three main procedures take place, namely *performance gradient estimation*, *line search*, and *dynamic deployment*, which are described below.

Performance Gradient Estimation

The gradient estimation phase aims at detecting a direction in the scalar parameter space G at which promising parameters of the algorithm exist. Such parameters shall improve the current population in terms of the AOV performance criterion. For better presentation, let us denote the AOV measure of population P after it has been evolved for t iterations with parameter vector $\bar{\rho}_c = (\rho_{1,c}, \ldots, \rho_{np,c})$, as:

$$H(P[\bar{\rho}_c, t]) = \frac{1}{N} \sum_{x \in P[\bar{\rho}_c, t]} f(x) \tag{4.1}$$

Then the negative of the approximate gradient of H at $\bar{\rho_c}$ is estimated by using the symmetric difference formula

$$-\nabla H\left(P_{pri}[\bar{\rho_c}, t]\right) = -\begin{pmatrix} \frac{\partial H(P_{pri}[\bar{\rho_c}, t])}{\partial \rho_{1,c}} \\ \vdots \\ \frac{\partial H(P_{pri}[\bar{\rho_c}, t])}{\partial \rho_{np,c}} \end{pmatrix}, \tag{4.2}$$

where the partial derivatives for the i-th parameter are defined as:

$$\frac{\partial H\left(P_{pri}[\bar{\rho_c}, t]\right)}{\partial \rho_{i,c}} = \frac{H\left(P_{pri}[\bar{\rho_c} + \lambda e_i, t]\right)) - H\left(P_{pri}[\bar{\rho_c} - \lambda e_i, t]\right))}{2\,\lambda}, \qquad i = 1, 2, \ldots, np,$$

where $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ is the $i - th$ row of the $np \times np$ identify matrix. The central difference estimation is preferred against the simple forward difference formula due to its better accuracy, despite its requirement for two function evaluations per estimation. In practice, the estimated quantities in the nominators are computed by copying the primary population P_{pri} into $2np$ secondary populations, each one assigned one of the $2np$ parameter vectors:

$$\bar{\rho_c} + i\lambda e_1, \ldots, \bar{\rho_c} + i\lambda e_{np}, \qquad i \in \{-1, +1\}.$$

Each secondary population is evolved for a small number of iterations, $t = t_{sec}$, using its assigned parameters. Based on the previous experience with GPAM (see Chapter 3), setting $t_{sec} \in [5, 10]$ is suggested. The estimation of derivatives is adequate since only local performance trends of the algorithm are needed. Moreover, the computed gradient vectors are normalized to become unit vectors, in order to alleviate scaling issues in the forthcoming steps.

The gradient estimation phase can be conducted either serially or in parallel. In the serial case, the $2 \times np$ secondary populations are sequentially evolved for t_{sec} iterations each, and their final AOV values are recorded. In the parallel case, the procedure becomes more efficient by evolving each secondary population on a different CPU. Obviously, further parallelization within each secondary population (e.g., through fork-join procedures) can be used if additional CPUs are available. Note that modern desktop systems usually offer eight CPUs or more.

An arguable issue in the above procedure is the comparability of the estimated performance (AOV) values of the secondary populations, This is due to the small t_{sec} number of iterations and the different sequences of random numbers used in each. This may render questionable whether the observed performance differences are the outcome of the different parameters or random fluctuations due to the different sequences of randomly selected numbers. Such performance fluctuations may distort the computed gradient directions in the parameter domain. In order to ameliorate this effect, the secondary populations assume exactly the same sequence of random numbers by seeding their random number generators with the same random seed prior to each performance estimation phase. Thus, the parameter pairs become the sole source of variability among the secondary populations in the simulations, thereby providing comparable estimations.

Naturally, even in this case the performance estimations do not provide exact gradient directions. However, the stochastic nature of population-based metaheuristics shall be taken into consideration. Hence, possible advantages of using more accurate gradient directions are most probably absorbed in the long-run due to the algorithm's stochasticity. Instead, the main purpose of using the gradient estimations is the identification of directions (or trends) in the parameter domain that seem to locally improve the algorithm's performance.

Line Search

The gradient estimation phase determines a direction in the parameter domain that locally improves the algorithm's performance. Given this direction, a mechanism is needed to determine the corresponding step size. In mathematical optimization this is typically addressed through line search. An estimation-based analogue of line search is adopted in the proposed GPALS approach to refine the outcome of the procedure described above.

Specifically, line search is used to determine the appropriate step size $s > 0$, in the direction of the estimated gradients, which is used for the production of new parameter vectors as follows:

$$\bar{\rho}_{c+1} = \bar{\rho}_c - s \nabla H \left(P_{pri}[\bar{\rho}_c, t] \right). \tag{4.3}$$

For this purpose, the derivative-free line search algorithm proposed in [82] is adopted.

Algorithm 4.1 Pseudocode of Bracketing and Bisection

Input: x (current point); l, u (bracketing scalars); g (gradient vector); G (parameters domain);
`/* Bracketing */`
$l \leftarrow 0$, $u \leftarrow 0.5$
$y \leftarrow x - u\, g$
while $(y \in G)$ **do**
  $l \leftarrow u$, $u \leftarrow u + 0.5$
  $y \leftarrow x - u\, g$
end while
`/* Bisection */`
while $(u > l)$ **do**
  $\mu \leftarrow 0.5\,(l + u)$
  $y \leftarrow x - \mu\, g$
  if $(y \notin G)$ **then**
    $u \leftarrow \mu$
  else
    $l \leftarrow \mu$
  end if
end while
$s_4 \leftarrow 0.5\,(l + u)\, g$
return s_4

The specific approach is based on the Golden Section method and has rapid convergence properties. For its application, four step size values are required,

$$s_1 < s_2 < s_3 < s_4,$$

each one defining a different point in the parameter domain through Eq. (4.3). The first value is taken as $s_1 = 0$ and corresponds to the current parameter vector $\bar{\rho}_c$. The rest are defined according to the Golden Section approach as

$$s_2 = s_4 - \gamma\,\Delta, \qquad s_3 = s_1 + \gamma\,\Delta,$$

where $\Delta = s_4 - s_1$, and $\gamma = (\sqrt{5}-1)/2$. The step size s_4 corresponds to the intersection point of the estimated direction with the boundary of the parameter domain G. Thus, it is determined through a bracketing and bisection procedure, which is reported in Algorithm 4.1 following the instructions in [82]. Note that this procedure does not add any computational overhead in terms of function evaluations.

The line search iterates on the step sizes until a termination condition is met. This

condition is related to the distance $0.5\,(s_4 - s_1)$. A reasonable choice is to terminate line search when this quantity becomes smaller than the reference step size λ used for the gradient estimation. For a more thorough description of the line search procedure, the reader is referred to [82].

The parameter vectors that correspond to the four step sizes in the line search procedure are evaluated through short runs of four secondary populations, similarly to the gradient estimation phase. Thus, four secondary populations, $P_{s_1}, \ldots, P_{s_4}$, are initiated as copies of the primary population P_{pri}. Then, each step size s_i is evaluated by performing t_{sec} iterations on P_{s_i}, using the parameter vector:

$$\bar{\rho}_{s_i} = \bar{\rho}_c - s_i \nabla H\left(P_{s_i}[\bar{\rho}_c, t_{sec}]\right).$$

Eventually, line search provides a step size s^* that corresponds to an improving parameter pair:

$$\bar{\rho}_* = \bar{\rho}_c - s^* \nabla H\left(P_{s^*}[\bar{\rho}_c, t_{sec}]\right), \tag{4.4}$$

where P_{s^*} is the corresponding secondary population. This parameter pair $\bar{\rho}_c$ and its corresponding secondary population P_{s^*} are used in the next phase of the method.

If all the estimated partial derivatives are almost zero within a prescribed tolerance $\delta > 0$ (10^{-8} is a typical value) for a parameter vector, then a local minimizer in the parameter domain may have been reached. In this case, line search is temporarily abandoned until at least one partial derivative becomes again higher than the prescribed tolerance δ. Note that, the gradients are still computed in every cycle of the method because the evolved primary population may perform better with different parameter values after some iterations. Alternatively, the parameter search procedure can be restarted if adequate computational budget is still available.

Dynamic Deployment Phase

In this phase, the best-performing secondary population P_{s^*} becomes the primary population, and its parameter vector $\bar{\rho}_*$ replaces the current parameter vector $\bar{\rho}_c$ if adequate performance improvement is achieved, i.e.,

$$H\left(P_{pri}[\bar{\rho}_c, t]\right) - H\left(P_{s^*}[\bar{\rho}_*, t]\right) > \theta_{\min} \geqslant 0,$$

where $\theta_{\min}$ is the smallest acceptable improvement (this is an optional parameter) and

Algorithm 4.2 Workflow of the proposed GPALS method

```
1: Initialize()
2: Evolve-Population()
3: while (not termination) do
4:    /* Performance Gradient Estimation */
5:    Gradient-Estimation()
6:    Normalization()
7:    /* Line Search */
8:    Bracketing()
9:    Golden-Section()
10:   /* Dynamic Deployment */
11:   Update-Population()
12:   Evolve-Population()
13: end while
```

t equals to either t_{sec} or t_{pri}. This step completes the GPALS cycle, and a new cycle begins by performing t_{pri} iterations on the new primary population with the new parameter pair. In case of insufficient improvement from the secondary poulations, the new cycle retains the previous primary population and parameter pair. In any case, the new primary population inherits the best individuals of all secondary populations. Thus, good solutions that may be sampled during the performance estimation procedure are not neglected.

The GPALS workflow is outlined in Algorithm 4.2 and graphically illustrated in Fig. 4.1. If N denotes the population size and n the problem dimension, the general population-based algorithm requires $N \times n$ memory positions while GPALS require $5 \times N \times n$. For a typical population size of hundreds (or even thousands) of individuals, this increase is bearable in modern desktop systems. Moreover, the new approach does not add in terms of computational burden since it is desired to use exactly the same budget of function evaluations as the plain algorithm. The gradient estimation and the line search add a fixed amount of operations, while the ratio of calls to the random number generator over the function evaluations is the same as for the standard algorithm. Thus, there are no time-consuming procedures introduced by GPALS.

Presumably the most critical part of the method is the estimation phase, especially the length of the short runs, followed by the dynamic deployment phase. Both these procedures are strongly based on the available computational budget. Subsequently follows the gradient step size λ, which is completely related to the algorithm and the desirable level of search granularity in the parameter domain. In the following

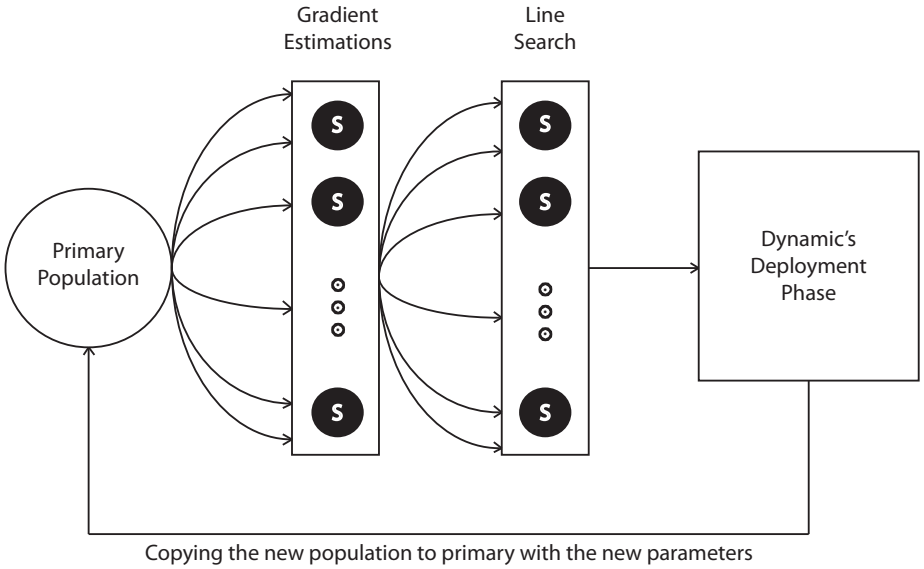


Figure 4.1: Graphical illustration of the GPALS method.

paragraphs, the general GPALS scheme demonstrated on DE and PSO, providing interesting results.

4.3 Application on Differential Evolution

The proposed GPALS method is demonstrated on the state-of-the-art DE algorithm for adapting its scalar parameters. The derived algorithm is henceforth denoted as GPALS-DE. The experiments were conducted on the two established test suites SOCO and CEC-2013 [81].

Consider the general optimization problem of Eq. (2.1), and N be the population size. Then, the parameter domain for the parameters F and CR, with prescribed ranges $[F_{\min}, F_{\max}]$, $[CR_{\min}, CR_{\max}]$ respectively, is defined as:

$$G = [F_{\min}, F_{\max}] \times [CR_{\min}, CR_{\max}] \,.$$

The primary population is assigned a current parameter pair $(F_c, CR_c) \in G$.

After initialization, the primary population is evolved for $t = t_{pri}$ iterations. The suggested value $t_{pri} = 10 \times n$, where n is the dimension of the considered optimization problem, is used. For the evolved primary population with the current parameter

pair, the AOV performance measure $H\left(P_{pri}[(F_c, CR_c), t]\right)$ is computed according to Eq. (4.1).

Then, the negative approximate gradient of H at (F_c, CR_c) is computed as:

$$-\nabla H\left(P_{pri}[(F_c, CR_c), t]\right) = -\begin{pmatrix} \frac{\partial H(P_{pri}[(F_c, CR_c), t])}{\partial F_c} \\ \\ \frac{\partial H(P_{pri}[(F_c, CR_c), t])}{\partial CR_c} \end{pmatrix}, \tag{4.5}$$

where the partial derivatives are given as:

$$\frac{\partial H\left(P_{pri}[(F_c, CR_c), t]\right)}{\partial F_c} = \frac{H\left(P_{pri}[(F_c + \lambda, CR_c), t]\right) - H\left(P_{pri}[(F_c - \lambda, CR_c), t]\right)}{2\,\lambda}, \tag{4.6}$$

and

$$\frac{\partial H\left(P_{pri}[(F_c, CR_c), t]\right)}{\partial CR_c} = \frac{H\left(P_{pri}[(F_c, CR_c + \lambda), t]\right) - H\left(P_{pri}[(F_c, CR_c - \lambda), t]\right)}{2\,\lambda}. \tag{4.7}$$

according to Eq. (4.3). The estimated quantities in the nominators are computed by copying the primary population P into four secondary populations, each one assigned one of the four parameter pairs

$$(F_c + i\,\lambda, CR_c), \qquad (F_c, CR_c + i\,\lambda), \qquad i \in \{-1, +1\}.$$

Each secondary population is then evolved for a small number of iterations, $t = t_{sec}$, using its assigned parameter pair. In the current implementation, $t_{sec} = 10$ was used following the suggestion in the previous chapter. Also, $\lambda = 0.1$ was used as the standard step size for the gradient estimation, since DE gives only marginal performance differences under smaller perturbation of its parameters [64, 79].

Following the general form of GPALS, line search is used to determine the appropriate step size $s > 0$. Specifically, in GPALS-DE the new parameter pair is generated as follows:

$$(F, CR) = (F_c, CR_c) - s\,\nabla H\left(P[(F_c, CR_c), t]\right). \tag{4.8}$$

For the application of the derivative-free line search algorithm, four step size values are required,

$$s_1 < s_2 < s_3 < s_4,$$

each one defining a different point in the parameter domain through Eq. (4.8). The

step sizes are determined following the procedure described in the previous section.

The line search iterates on the step sizes until a termination condition is met. The condition is related to the distance $0.5\,(s_4 - s_1)$, and specifically when this becomes lower than the step size λ. The four secondary populations, $P_{s_1}, \ldots, P_{s_4}$, are initiated as copies of the primary population P. Then, each step size s_i is evaluated by performing t_{sec} iterations on P_{s_i}, using the parameter pair:

$$(F_{s_i}, CR_{s_i}) = (F_c, CR_c) - s_i \nabla H\left(P_{s_i}[(F_c, CR_c), t_{sec}]\right).$$

Eventually, line search provides a step size s^* that corresponds to the improving parameter pair:

$$(F^*, CR^*) = (F_c, CR_c) - s^* \nabla H\left(P_{s^*}[(F_c, CR_c), t_{sec}]\right). \tag{4.9}$$

This parameter pair and its corresponding secondary population P_{s^*} become the primary population if adequate performance improvement is achieved, i.e.,

$$H\left(P_{pri}[(F_c, CR_c), t]\right) - H\left(P_{s^*}[(F^*, CR^*), t]\right) > \theta_{\min} \geqslant 0,$$

where $\theta_{\min}$ is the smallest acceptable improvement and t equals to either t_{sec} or t_{pri}. This step completes the GPALS-DE cycle, and a new cycle begins by performing t_{pri} iterations with the new primary population and the new parameter pair. In case of insufficient improvement from the secondary populations, the new cycle retains the previous primary population and parameter pair. In any case, the new primary population inherits the best individuals of all secondary populations.

Sample trajectories of DE's scalar parameters for two test problems are illustrated in Fig. 4.2. The estimated gradient directions are shown in red dashed lines, while two different parameter trajectories appear in darker colors with different markers denoting different problems.

The first experimental assessment was conducted on the SOCO test suite. According to the suggestions in previous sections, the parameters of GPALS assumed the following values in all experiments:

$$t_{pri} = 10 \times n, \quad t_{sec} = 10, \quad \lambda = 0.1, \quad \delta = 10^{-8}.$$

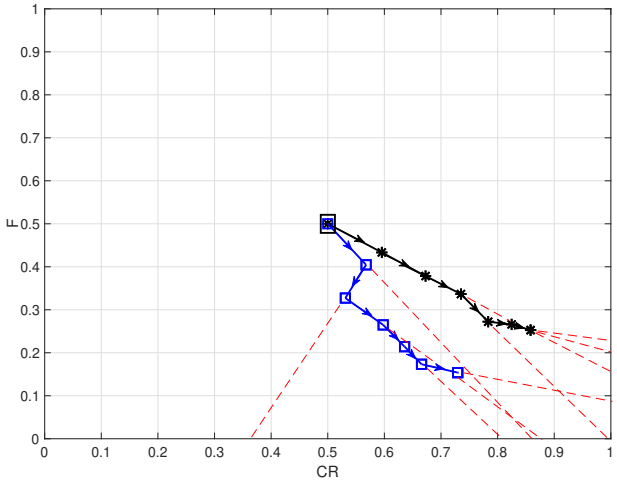


Figure 4.2: Sample trajectories of the DE's parameters for two test functions.

The available computational budget, as dictated by the test suite, was equal to $q_{\max} = 10000 \times n$ function evaluations, while the performance criterion for the algorithms was the objective value error as defined in Eq. (2.13).

The GPALS-DE algorithm adopted the exact settings of the base DE algorithm in the test suite [57]. Specifically, it was based on the exponential crossover operator with the DE/rand/1 mutation operator (see Eq. (2.4)), and population size $N = 60$. For each test problem, 25 independent experiments were conducted, recording the best solution that was detected within the available computational budget $q_{\max}$. Following the experimental setup of GPAM in the previous chapter, the initial parameters were selected in the center of the search space. However, additional experiments with extremal parameter pairs were also conducted. It shall be noted that the competitor algorithms were again already tuned for the test suite, while the proposed GPALS-DE algorithm assumed no prior information.

The DE parameters F and CR are usually set in the range $(0, 1]$ [8], hence, the parameter domain is defined as:

$$G = (0, 1] \times (0, 1]. \tag{4.10}$$

GPALS-DE was initially tested using the central initial parameter pair $(F, CR) = (0.5, 0.5)$. The extreme initial pairs $(F, CR) = (0.2, 0.2)$ and $(F, CR) = (0.8, 0.8)$ were subsequently investigated. These approaches are henceforth denoted as GPALS-DE$_{0.5}$, GPALS-DE$_{0.2}$, and GPALS-DE$_{0.8}$, respectively.

Table 4.1: Statistical comparisons between the three GPALS-DE variants and the base algorithms for the SOCO test suite.

		GPALS-$DE_{0.5}$			GPALS-$DE_{0.2}$			GPALS-$DE_{0.8}$		
Dim.	Algorithm	+	−	=	+	−	=	+	−	=
50	DE_{bin}	15	3	1	14	3	2	13	3	3
	DE_{exp}	10	5	4	11	6	2	4	8	7
	CHC	19	0	0	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0	16	3	0
100	DE_{bin}	17	1	1	17	1	1	17	1	1
	DE_{exp}	13	4	2	12	4	3	10	6	3
	CHC	19	0	0	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0	16	3	0
200	DE_{bin}	17	1	1	17	1	1	17	1	1
	DE_{exp}	13	4	2	14	4	1	13	6	0
	CHC	19	0	0	19	0	0	19	0	0
	GCMAES	16	3	0	17	2	0	16	3	0
500	DE_{bin}	19	0	0	19	0	0	19	0	0
	DE_{exp}	13	5	1	11	4	4	13	6	0
	CHC	19	0	0	19	0	0	19	0	0
	GCMAES	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

The average error and the standard deviation per test problem for the three GPALS-DE variants and the base algorithms of the test suite are reported in Tables B.1-B.4 of Appendix B for all dimensions. In order to verify the significance of the observed performance differences and facilitate the extraction of sound conclusions, Wilcoxon rank-sum tests were conducted, according to the settings in the previous chapter.

The obtained results are reported in Table 4.1 where wins, losses, and ties are denoted as "+", "−", and "=", respectively. The GPALS-DE variants clearly outperformed the base algorithms in all dimensions with one exception, namely GPALS-$DE_{0.8}$ which slightly deviated from this performance only for the lowest dimension $n = 50$. In that case, it was outperformed by the fine-tuned DE_{exp} algorithm, which is reported to be the best-performing algorithm in the test suite repository. This shortcoming of GPALS-$DE_{0.8}$ can be attributed to the specific initial parameter pair, which reduces the convergence speed of DE in the specific test problems (similar findings were discovered for GPAM in the previous chapter). Thus, GPALS needs additional effort to reach appropriate parameter values that lead the algorithm to optimal solutions. Nevertheless, it offers clear evidence that even under defective parameter

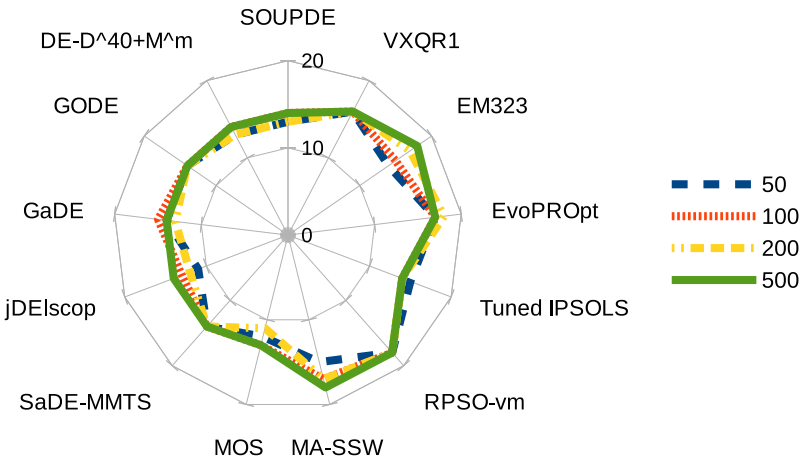


Figure 4.3: Number of test problems where GPALS-DE$_{0.5}$ achieved equal or better average error than the competitor algorithms.

initialization, and without any additional experimentation, GPALS-DE can be highly effective.

The GPALS-DE$_{0.5}$ approach exhibited better or equivalent performance than the rest of the GPALS-DE variants, and it was further compared against the rest of the algorithms of the test suite in terms of the average error values [42]. The results are reported in Tables B.5 and B.6 of Appendix B. The data for the rest of the algorithms are reproduced from the original sources [67]. Figure 4.3 illustrates the number of problems (out of 19 problems) where GPALS-DE$_{0.5}$ was not outperformed by other algorithms. It is easily perceived that GPALS-DE was highly competitive also against non-DE algorithms for all dimensions.

Moreover, Wilcoxon rank-sum tests were conducted between GPALS-DE$_{0.5}$ and the previously proposed DEGPA approach presented in the previous chapter [64]. The percentage of wins, losses, and ties between GPALS-DE$_{0.5}$ and DEGPA are graphically illustrated in Fig. 4.4. Clearly, GPALS-DE$_{0.5}$ achieved better or equivalent performance in almost all test problems, particularly for higher dimensions. This observation verifies the additional benefits of using the approximate performance gradients with line search against the simpler grid-based DEGPA approach.

Besides the comparisons with the competitor algorithms of the test suite, GPALS-DE$_{0.5}$ was additionally compared with three state-of-the-art adaptive DE algorithms, namely GaDE [27], SHADE [29] and L-SHADE [30], on the SOCO test problems. For this purpose, source codes available on the test suites' repositories were used. The provided source code of GaDE was already tuned on the specific test suite,

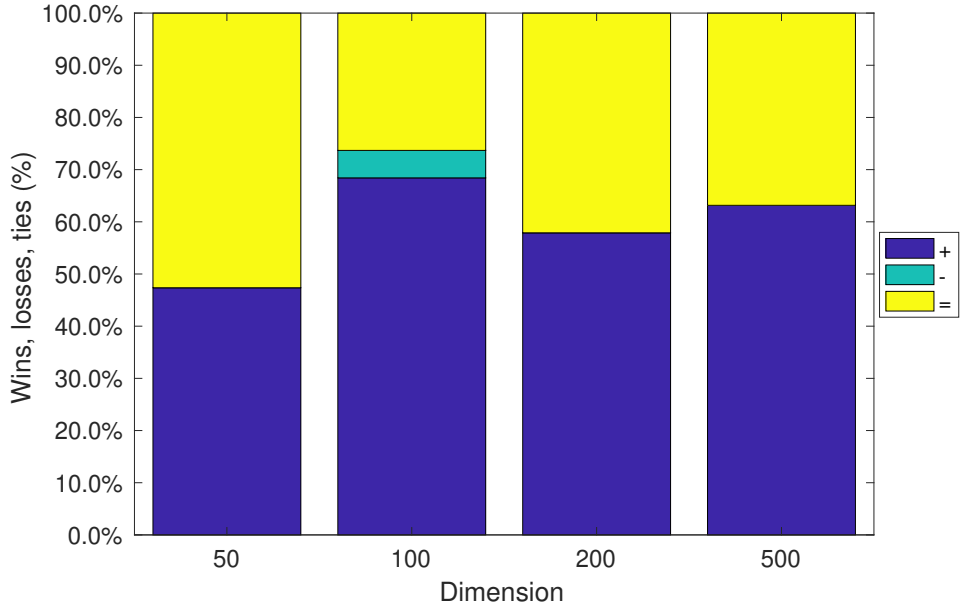


Figure 4.4: Wins, losses, and ties of GPALS-DE$_{0.5}$ against the DEGPA approach.

requiring no further modification. SHADE and L-SHADE are based on an adaptive scheme that employs a special mutation operator, which differs from the one in the standard DE, which is also used in the GPALS-DE approach. Also, their available codes came with proposed parameters that were tuned on the CEC-2013 test suite. Thus, in a head-to-head comparison with the GPALS approaches, it would be difficult to identify whether the observed good or bad performance trends can be attributed to the main parameter adaptation procedure of these algorithms or to their special mutation scheme, which is irrelevant to the parameter adaptation. For this reason, both the original versions of SHADE and L-SHADE were considered, as well as a modified version of each one, denoted as mSHADE and mL-SHADE, respectively. In these modified versions, the DE/rand/1 operator with exponential mutation that is used in GPALS-DE was adopted in SHADE and L-SHADE instead of their special mutation operator.

Moreover, the test suite requires populations of size $N = 60$. However, the proposed size for SHADE is $N = 100$ [29]. Thus, both population sizes were considered in the comparisons and denoted with a subscript, i.e., SHADE$_{60}$ and SHADE$_{100}$. Moreover, L-SHADE assumes linearly decreasing population size during its run, starting from a fixed value $N = 18 \times n$. This scheme was also retained in the experiments.

The results of the Wilcoxon rank-sum tests of GPALS-DE$_{0.5}$ against GaDE, SHADE and the L-SHADE variants are reported in Table 4.2. In the 500-dimensional case,

Table 4.2: Number of wins, losses, and ties of the GPALS-$DE_{0.5}$ algorithm against GaDE, SHADE, and L-SHADE for the SOCO test problems.

	Dimension											
	50			100			200			500		
GPALS-$DE_{0.5}$ vs.	+	−	=	+	−	=	+	−	=	+	−	=
GaDE	7	3	9	8	3	8	8	4	7	7	4	8
$SHADE_{60}$	7	1	11	13	1	5	15	1	3	17	1	1
$SHADE_{100}$	5	3	11	13	1	5	13	1	5	16	1	2
$mSHADE_{60}$	6	2	11	6	1	12	8	1	10	16	1	2
$mSHADE_{100}$	13	1	5	13	1	5	14	1	4	14	1	4
L-SHADE	9	2	8	13	2	4	14	2	3	n/a	n/a	n/a
mL-SHADE	18	1	0	18	0	1	19	0	0	n/a	n/a	n/a

the provided source codes of L-SHADE failed to execute due to excessive memory demand. The results show that GPALS-$DE_{0.5}$ is highly competitive to all competitor algorithms, with evident superiority as dimension increases. It is interesting to notice that, despite the significant number of function evaluations spared by L-SHADE due to its decreasing population size, GPALS-$DE_{0.5}$ was superior. For completeness purpose, the corresponding average errors for SHADE and L-SHADE are reported in Tables B.7 and B.8 of Appendix B.

In order to gain further insight on the impact of population size on the performance of GPALS-$DE_{0.5}$, additional experiments were conducted for $N = 40$ and 80 and compared to the base algorithms. The obtained results are reported in Table 4.3, where the corresponding population size is denoted as superscript. As can be seen, GPALS-$DE_{0.5}^{40}$ and GPALS-$DE_{0.5}^{60}$ exhibited similar and superior performance than the competitor algorithms, while GPALS-$DE_{0.5}^{80}$ slightly deviated from this performance especially for lower dimensions against the best base algorithm, namely DE_{exp}. Nevertheless, this effect is ameliorated as dimension increases. This is a consequence of using higher population sizes in easier problems of smaller dimension, which results in futile spending of additional function evaluations in the performance estimation phase. The same effect can also be observed against the SHADE variants in Table 4.4.

Finally, the running time of GPALS-$DE_{0.5}$ against SHADE was also recorded. Running time is not a measure that can offer sound conclusions regarding the algorithm's performance, because it depends on many external factors such as implementation quality, machine configuration, employed compilers, and machine workload at the moment of experimentation. Nevertheless, it is an indication that the nice performance

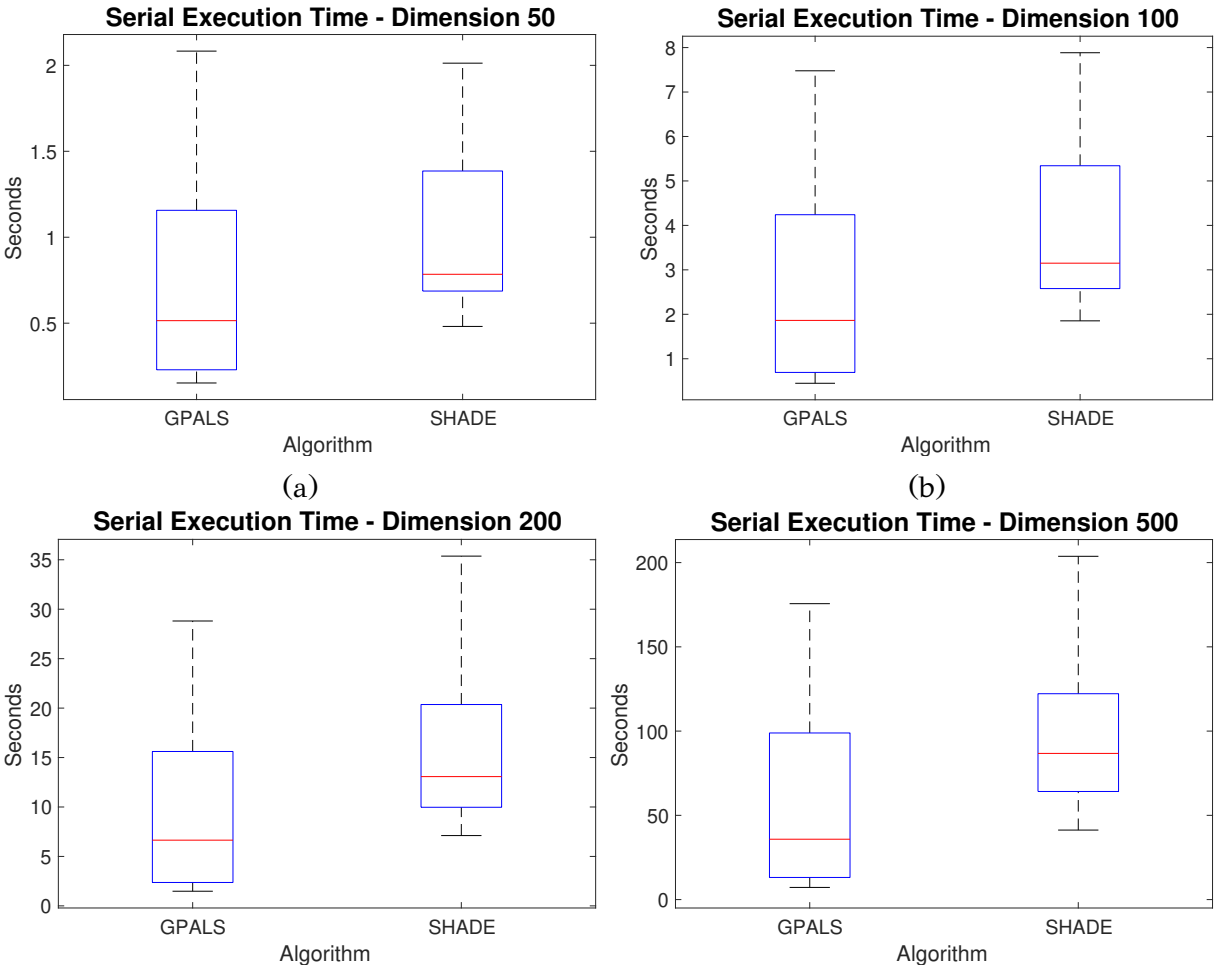


Figure 4.5: Running time for the serial version of GPALS-DE$_{0.5}$ and SHADE in the SOCO test suite for dimension (a) $n = 50$, (b) $n = 100$, (c) $n = 200$, and (d) $n = 500$.

Table 4.3: Statistical comparisons between the GPALS-$DE_{0.5}$ algorithms with different population size and the base algorithms for the SOCO test suite.

		GPALS-$DE^{40}_{0.5}$			GPALS-$DE^{60}_{0.5}$			GPALS-$DE^{80}_{0.5}$		
Dim.	Algorithm	+	−	=	+	−	=	+	−	=
50	DE_{bin}	14	4	1	15	3	1	11	5	3
	DE_{exp}	11	5	3	10	5	4	3	15	1
	CHC	19	0	0	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0	16	3	0
100	DE_{bin}	17	1	1	17	1	1	16	1	1
	DE_{exp}	13	5	1	13	4	2	6	8	5
	CHC	19	0	0	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0	16	3	0
200	DE_{bin}	17	1	1	17	1	1	17	1	1
	DE_{exp}	12	4	3	13	4	2	6	9	4
	CHC	19	0	0	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0	16	3	0
500	DE_{bin}	19	0	0	19	0	0	19	0	0
	DE_{exp}	11	4	4	13	5	1	11	6	2
	CHC	19	0	0	19	0	0	19	0	0
	GCMAES	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a

of an algorithm is not achieved in excessively higher time than another algorithm. In this case, the available SHADE source code was serial, so 25 serial experiments were conducted on the same machine for each algorithm. The recorded running times are illustrated in Fig. 4.5. Obviously, the proposed approach achieves lower execution time than the competitor SHADE algorithm in all dimensions. In addition, differences in running times were statistical evaluated with Wilcoxon rank-sum tests, and wins, losses, and ties are reported in Table 4.5, verifying the observations.

The performance of GPALS-$DE_{0.5}$ was further investigated on the CEC-2013 test suite [58]. The parameters of GPALS-DE were the same as for the SOCO test suite. For comparison purposes, SHADE and L-SHADE were considered as the baseline algorithms for comparisons. Both have been shown to be among the most efficient adaptive DE algorithms for the specific test suite [70]. For notation simplicity, GPALS-$DE_{0.5}$ is henceforth denoted simply as GPALS-DE.

The main goal remained the investigation of GPALS as a general parameter adaptation method, using DE for demonstration purposes. Thus, in order to achieve a fair comparison between GPALS-DE and the parameter adaptation scheme of SHADE, GPALS-DE with the special mutation operator with population archive was also

Table 4.4: Number of wins, losses, and ties of GPALS-DE against SHADE for different population sizes in the SOCO test problems.

	Dimension											
	50			100			200			500		
GPALS-DE$^{40}_{0.5}$ vs.	$+$	$-$	$=$	$+$	$-$	$=$	$+$	$-$	$=$	$+$	$-$	$=$
SHADE$_{40}$	13	2	4	16	1	2	18	1	0	18	1	0
mSHADE$_{40}$	3	8	8	5	5	9	13	5	1	18	1	0
GPALS-DE$^{80}_{0.5}$ vs.	$+$	$-$	$=$	$+$	$-$	$=$	$+$	$-$	$=$	$+$	$-$	$=$
SHADE$_{80}$	5	13	1	12	6	1	13	5	1	18	1	0
mSHADE$_{80}$	15	3	1	14	4	1	14	4	1	18	1	0

Table 4.5: Statistical comparisons of running times between the serial version of GPALS-DE$_{0.5}$ and SHADE for the SOCO test problems.

Dim.	GPALS-DE vs.	$+$	$-$	$=$
50	SHADE	16	2	1
100	SHADE	17	0	2
200	SHADE	19	0	0
500	SHADE	18	1	0

considered. Note that the corresponding modification was made for SHADE and L-SHADE previously in the SOCO test suite, resulting in mSHADE and mL-SHADE. Thus, two GPALS-DE approaches was considered, namely GPALS-DE that uses the original DE/rand/1 mutation operator with exponential crossover, and mGPALS-DE that uses SHADE's mutation operator with population archive. Whenever a mutant vector component of the GPALS approach was violating the boundary of the search space, the SHADE correction [29] was applied. Besides the original SHADE and L-SHADE algorithms, the mSHADE and mL-SHADE defined in the previous section were also considered for the CEC-2013 test suite. This was motivated by the necessity to investigate whether the efficiency of the parameter adaptation scheme of SHADE and L-SHADE is intimately related to its special mutation operator or it can work equally good with the plain DE mutation operator as well. The population size for all algorithms was equal to $N = 60$. In all runs, GPALS-DE assumed as initial parameter pair the central point of the search space, namely $(F, CR) = (0.5, 0.5)$. The rest of the settings followed closely those of the test suite.

Table 4.6 reports the results of the Wilcoxon rank-sum tests between GPALS-DE, SHADE, and L-SHADE. We can see that SHADE and L-SHADE dominated the mGPALS-DE algorithm in most of the test problems. However, this was not the case

Table 4.6: Statistical comparisons between GPALS-DE, SHADE, and L-SHADE on the CEC-2013 test suite.

	Dimension					
	30			50		
	+	−	=	+	−	=
mGPALS-DE vs SHADE	3	21	4	3	19	6
GPALS-DE vs mSHADE	8	9	11	12	9	7
mGPALS-DE vs L-SHADE	1	23	4	1	22	5
GPALS-DE vs mL-SHADE	5	12	11	5	18	5

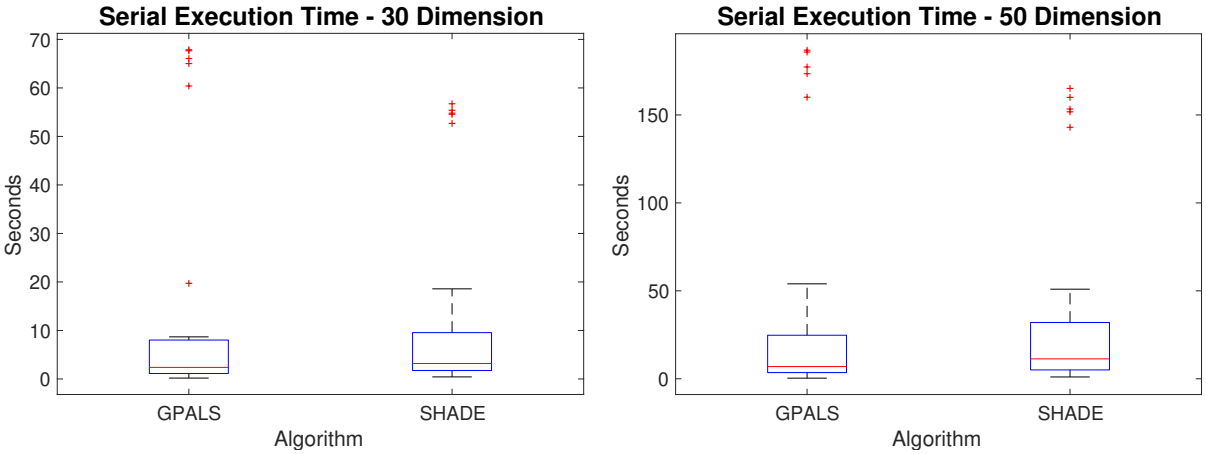
Table 4.7: Statistical comparisons between GPALS-DE and other adaptive DE-based algorithms on the CEC-2013 test suite.

	Dimension					
	30			50		
GPALS-DE vs.	+	−	=	+	−	=
DEcfbLS	10	12	6	11	11	6
jande	12	7	9	14	12	2
DE_APC	12	7	9	11	10	7
PVADE	14	8	6	18	4	6

when the standard DE operator was used in SHADE, as revealed by the comparisons against GPALS-DE. Especially when dimension increases to 50, GPALS-DE shows superior performance compared to SHADE. On the other hand, L-SHADE achieves better performance than GPALS-DE, which obviously stems from the population size reduction, that spares computational budget. These results suggest that the parameter adaptation scheme of SHADE and L-SHADE, although very effective in the low-dimensional CEC-2013 test suite, seems to be intimately related to its special mutation operator.

Further comparisons of GPALS-DE were conducted with various other adaptive DE algorithms. Table 4.7 reports the results for four adaptive DE-based algorithms, which can be found in [59–63]. Obviously, GPALS-DE exhibited competitive performance even for the low-dimensional problems.

Finally, running time analysis was conducted for the serial version of the algorithms, following the same methodology as in the previous section. The corresponding results are illustrated in Fig. 4.6. The GPALS-DE approach achieved again better running times than SHADE. These findings are confirmed in the statistical tests reported in Table 4.8.



(a) (b)

Figure 4.6: Running time for the serial version of GPALS-DE and the baseline SHADE algorithm in the CEC-2013 test suite, for dimension (a) $n = 30$, and (b) $n = 50$.

Table 4.8: Statistical comparisons of the running times between GPALS-DE and the SHADE algorithm for the CEC-2013 test suite.

Dim.	GPALS-DE vs.	+	−	=
30	SHADE	20	7	1
50	SHADE	20	7	1

4.4 Application on Particle Swarm Optimization

The proposed GPALS method is further demonstrated on the state-of-the-art PSO algorithm for adapting its scalar control parameters. The derived algorithm is henceforth denoted as GPALS-PSO. The experiments were conducted on the the SOCO test suite. PSO has three real-valued parameters, namely w, c_1, and c_2, while it exhibits moderate performance in the SOCO test suite. This fact offered additional motivation in order to investigate whether the proposed approach could render the plain PSO more competitive.

Closely following the previous analysis for DE, a 3-dimensional search space G was considered for the PSO parameters. Specifically, both c_1 and c_2 were assumed to lie in the range $[0, 3]$, while the range $[0, 1]$ was considered for w. Thus, the search space was defined as,

$$G = [0, 1] \times [0, 3] \times [0, 3]. \tag{4.11}$$

The experiments were conducted by initializing GPALS-PSO at the central parameter point $(w, c_1, c_2) = (0.5, 1.5, 1.5)$, as well as the more extremal points $(w, c_1, c_2) =$

$(0.2, 1.0, 1.0)$ and $(w, c_1, c_2) = (0.8, 2.0, 2.0)$. These approaches are denoted as GPALS-PSO$_{0.5}$, GPALS-PSO$_{0.2}$, and GPALS-PSO$_{0.8}$, respectively. The swarm size was set to $N = 100$ and the computational budget was equal to $q_{\max} = 5000 \times n$, as dictated by the test suite. Each competenting PSO variant assumed the same parameter set with the corresponding GPALS-PSO. Note that all PSO variants were based on the *lbest* PSO model with ring neighborhood topology of radius $r = 1$.

For the evolved primary population with the current parameter triplet, the performance measure $H\left(P_{pri}[(w_c, c_{1,c}, c_{2,c}), t]\right)$ is computed according to Eq. (4.1) and the approximate gradient of H closely follows the same approach as for DE described in the previous section, according to Eqs. (4.2) and (4.3).

The estimated quantities in the gradient are computed by copying the primary population P into six secondary populations, each one assigned one of the four parameter pairs

$$(w_c + i\,\lambda, c_{1,c}, c_{2,c}), \qquad (w_c, c_{1,c} + i\,\lambda, c_{2,c}), \qquad (w_c, c_{1,c}, c_{2,c} + i\,\lambda), \qquad i \in \{-1, +1\}.$$

Each secondary population is evolved for t_{sec} iterations, using its assigned parameters. In the current implementation, $t_{sec} = 10$ and $\lambda = 0.1$ was used similarly to the DE case.

The best-performing secondary population P_{s^*} becomes the primary population, and its parameter triplet $(w_*, c_{1,*}, c_{2,*})$ replaces the current one $(w_c, c_{1,c}, c_{2,c})$ if adequate performance improvement is achieved, i.e.,

$$H\left(P_{pri}[(w_c, c_{1,c}, c_{2,c}), t]\right) - H\left(P_{s^*}[(w_*, c_{1,*}, c_{2,*}), t]\right) > \theta_{\min} \geqslant 0,$$

where $\theta_{\min}$ is the smallest acceptable improvement and t equals to either t_{sec} or t_{pri}. This step completes the GPALS-PSO cycle, and a new cycle begins by performing t_{pri} iterations with the new primary population and the new parameter triplet. In case of insufficient improvement from the secondary populations, the new cycle retains the previous primary population and parameters. In any case, the new primary population inherits the best individuals of all secondary population.

The experiments on the SOCO test suite were conducted using the same parameters for GPALS-PSO as for GPALS-DE, namely,

$$t_{pri} = 10 \times n, \quad t_{sec} = 10, \quad \lambda = 0.1, \quad \delta = 10^{-8}.$$

Table 4.9: Statistical comparisons between GPALS-PSO, the base algorithms, and the plain PSO for the SOCO test suite. The symbols "+", "−", and "=", denote wins, losses, and ties of GPALS-PSO against the competitor algorithms.

		GPALS-PSO$_{0.5}$			GPALS-PSO$_{0.2}$			GPALS-PSO$_{0.8}$		
Dim.	Algorithm	+	−	=	+	−	=	+	−	=
50	DE$_{\text{bin}}$	6	12	1	2	16	1	4	14	1
	DE$_{\text{exp}}$	3	13	3	2	17	0	2	17	0
	CHC	13	4	2	8	9	2	9	3	7
	GCMAES	14	3	2	9	6	4	11	4	4
	PSO	3	5	11	19	0	0	19	0	0
100	DE$_{\text{bin}}$	4	10	5	1	16	2	5	13	1
	DE$_{\text{exp}}$	1	16	2	1	18	0	1	17	1
	CHC	14	4	1	7	6	6	11	3	5
	GCMAES	13	5	1	8	5	6	11	6	2
	PSO	3	5	11	19	0	0	19	0	0
200	DE$_{\text{bin}}$	4	7	8	4	10	5	5	12	2
	DE$_{\text{exp}}$	1	18	0	1	18	0	1	18	0
	CHC	10	5	4	10	8	1	7	9	3
	GCMAES	12	6	1	5	8	6	6	10	3
	PSO	4	2	13	18	0	1	19	0	0
500	DE$_{\text{bin}}$	12	5	2	3	15	1	3	11	5
	DE$_{\text{exp}}$	1	18	0	1	18	0	1	18	0
	CHC	9	6	4	3	13	3	7	11	1
	GCMAES	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	PSO	9	2	8	9	0	10	19	0	0

The available computational budget, as dictated by the test suite, was equal to $q_{\max} = 1000 \times n$, function evaluations, while the performance criterion for the algorithms was again the objective value error as defined in Eq. (2.13).

The average and standard deviation of the obtained solution errors of GPALS-PSO are reported in Tables B.9–B.12 of Appendix B along with the results of the base algorithms. Wilcoxon rank-sum tests at confidence level 95% were conducted between the GPALS-PSO algorithms and the base algorithms. Table 4.9 reports the corresponding numbers of wins, losses, and ties. We see, the GPALS-PSO approach was competitive against two out of four base algorithms, while it was outperformed by the rest. However, it shall be underlined that GPALS-PSO does not use any kind of preprocessing, contrary to the tuned base algorithms.

The benefits from using GPALS become evident when GPALS-PSO is compared against the plain PSO with fixed parameters equal to the initial parameters of the GPALS-PSO. As revealed in Table 4.9, two of the GPALS-PSO approaches outper-

Table 4.10: Statistical comparisons between GPALS-PSO, plain PSO and SPSO 2011 on the CEC-2013 test problems.

	Dimension					
	30			50		
GPALS-PSO vs	+	−	=	+	−	=
PlainPSO	10	5	13	10	5	13
SPSO2011	9	5	14	14	6	8

formed the corresponding standard PSO algorithm. However, this was not the case for GPALS-PSO$_{0.5}$. The reason is that the parameter set $w = 0.5$, $c_1 = c_2 = 1.5$, was near-optimal in the experiments for the plain PSO on the specific problems and, thus, the dominant result between GPALS-PSO and plain PSO was the tie. Although, even for this case, PSO outperformed GPALS-PSO only in 2 out of 19 problems for the highest dimensional problems. This comes in line with the results observed for the GPALS-DE algorithms, suggesting that GPALS can be beneficial in higher dimensions even for the well-performing algorithms.

The performance of GPALS-PSO$_{0.5}$ was investigated also on the CEC-2013 suite [58].

The GPALS-PSO algorithm assumed in all experiments the initial parameter set $(w, c_1, c_2) = (0.5, 1.5, 1.5)$ but with swarm size increased to 100, similarly to its DE-based counterpart. Comparisons were conducted with the plain PSO approach and the new PSO variant, namely SPSO2011 [83]. The results are reported in Table 4.10. Evidently, GPALS-PSO is very competitive to both approaches. However, it shall be noticed that SPSO2011 is tuned for the specific test suite, while GPALS-PSO determines its parameters on the run. The average errors achieved for all algorithms for the CEC-2013 test suite are reported in Tables B.13-B.15 of Appendix B.

4.5 Synopsis

The grid-base parameter adaptation method presented in the previous chapter admits improvements in two points. The first one is the restricted number of search directions in the discretized parameter domain, and the second one is the limited step size allowed on these directions. These weaknesses may result in rather slow adaptation of the algorithm's parameters.

The proposed GPALS method addresses these issues by extending the parameter

search procedure from the simplistic grid-based search to the more sophisticated gradient search with line search. While gradient estimations enhance the number of search directions, line search tackles the step size problem. All estimations of the underlying mathematical quantities are based on short-runs of the algorithm, similarly to the grid-based method.

The proposed GPALS method was illustrated on the two state-of-the-art algorithms, namely, DE and PSO. The effectiveness of the corresponding GPALS-DE approach was shown on the two established test suites that include a variety of high-dimensional and low-dimensional test problems. The results suggested that GPALS can disburden the user from the need of finding proper parameter settings, while it can significantly improve the algorithm's performance. Very competitive performance was observed also against other already fine-tuned algorithms from the relevant literature, despite the fact that GPALS neither undergoes fine-tuning nor is assigned the additional computational budget spent by the rest of the algorithms for their optimal tuning. Moreover, compared to other methods, GPALS has shown satisfactory running time performance on the specific test problems.

Chapter 5

Concluding Remarks and Future Work

Parameter tuning is a laborious task in metaheuristics, intimately related to the corresponding optimization problem. The rising popularity of metaheuristics rendered parameter tuning a central problem, since their performance has been shown to be highly dependent on their proper parameterization. Inappropriate parameters may render the algorithm incapable of detecting solutions of good quality. On the other hand, parameter tuning through trial-and-error procedures expands the necessary experimentation time and consumes valuable computational resources. Up-to-date, a number of adaptive algorithms have been proposed, mostly based on ad-hoc procedures especially designed for the specific algorithm.

In the present thesis, two novel and general-purpose online parameter adaptation methods for population-based metaheuristics were proposed. First, a general-purpose parameter adaptation method based on grid search in the parameter domain was proposed. The parameter adaptation is conducted on a grid defined over the parameter space. Performance estimations of the algorithm are used to guide the adaptation procedure. The method was successfully demonstrated on two metaheuristics, namely Differential Evolution and Particle Swarm Optimization, attaining competitive performance against other state-of-the-art methods. Two test suites that include a variety of high-dimensional and low-dimensional test problems were used for the experimental evaluation. The proposed approaches do not require any preprocessing phase, contrary to the competitor algorithms. The reported results verified the competitiveness of the proposed methods. This was achieved despite the fact that the dictated

experimental setting of both test suites are rather restrictive due to the limited computational budget, and the fact that the parameters of the competitor algorithms were already tuned.

Similarly to other methods, the grid-based search has a few user-defined parameters. A preliminary study of its sensitivity on these parameters was also conducted. The results offered evidence of the method's tolerance on its parameters. The analysis revealed that the performance estimation phase is the most sensitive one, while the rest of the parameters have only mild influence on the algorithm's dynamic. Also, it revealed that the proposed default parameters are very efficient.

A more sophisticated version of the grid-based method, inspired by deterministic gradient-based optimization, was also developed. In this method, gradient estimations replace the core mechanism of the grid-based method, offering an abundance of search directions, while line search tackles the step size problem. The effectiveness of the proposed approach was successfully verified on the two test suites for the same metaheuristics, namely DE and PSO. The reported results verified the superiority of the proposed method against other already fine-tuned algorithms from the relevant literature, despite the fact that the proposed approach was neither fined-tuned nor using the additional computation budget spent by the rest of the algoirthms for their fine tuning. Finally, compared to other methods, the proposed approach exhibited satisfactory running-time performance on the specific test problems, and mild memory requirements.

The research reported in the present thesis has offered strong motivation for further elaboration on different metaheuristics and test suites, especially ones containing real world problems. Additionally, it would be interesting to further extend the methods with different adaptation techniques, e.g., different search methods or online learning algorithms. Finally, the proposed method could benefit from the use of modern computing infrastructures, such as GPU-based hardware and parallel programming tools.

Bibliography

[1] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[2] M. Gendreau and J. Potvin, *Handbook of Metaheuristics, 2nd edn.* Springer New York Dordrecht, Heidelberg London, 2010.

[3] A. E. Eiben and S. K. Smit, "Evolutionary algorithm parameters and methods to tune them," in *Autonomous Search* (Y. Hamadi, E. Monfroy, and F. Saubion, eds.), ch. 2, pp. 15–36, Berlin Heidelberg: Springer, 2011.

[4] G. Karafotias, M. Hoogendoorn, and A. E. Eiben, "Parameter control in evolutionary algorithms: Trends and challenges," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 167–187, 2015.

[5] M. Birattari, Z. Yuan, P. Balaprakash, and T. Stützle, "F-Race and iterated F-Race: An overview," in *Experimental Methods for the Analysis of Optimization Algorithms* (T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, eds.), pp. 311–336, Berlin, Heidelberg: Springer, 2010.

[6] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimization*, vol. 11, pp. 341–359, 1997.

[7] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," in *WSEAS Int. Conf. on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation*, pp. 293–298, Press, 2002.

[8] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

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

[10] A. Qing, *Differential Evolution: Fundamentals and applications in Electrical Engineering*. Singapore: John Wiley & Sons, 2009.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948 vol.4, 1995.

[12] K. E. Parsopoulos and M. N. Vrahatis, *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Publishing (IGI Global), 2010.

[13] K. E. Parsopoulos, "Particle swarm methods," in *Handbook of Heuristics* (M. Resende, R. Marti, and P. Pardalos, eds.), Springer, 2016.

[14] T. Bartz-Beielstein, *Experimental Research in Evolutionary Computation*. Berlin: Springer-Verlag, 2006.

[15] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy. Selected Papers* (A. C. Coello Coello, ed.), pp. 507–523, Berlin, Heidelberg: Springer, 2011.

[16] H. H. Hoos, "Automated algorithm configuration and parameter tuning," in *Autonomous Search* (Y. Hamadi, E. Monfroy, and F. Saubion, eds.), ch. 3, pp. 37–72, Berlin Heidelberg: Springer, 2011.

[17] T. Bartz-Beielstein, "SPOT: An R package for automatic and interactive tuning of optimization algorithms by sequential parameter optimization," *arXiv preprint arXiv:1006.4645*, 2010.

[18] T. Bartz-Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, *Experimental methods for the analysis of optimization algorithms*. Springer, 2010.

[19] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation.*, vol. 10, no. 6, pp. 646–657, 2006.

[20] S. Z. Zhao, P. N. Suganthan, and S. Das, "Self-adaptive differential evolution with multi-trajectory search for large-scale optimization," *Soft Computing*, vol. 15, no. 11, pp. 2175–2185, 2011.

[21] M. Weber, V. Tirronen, and F. Neri, "Scale factor inheritance mechanism in distributed differential evolution," *Soft Computing*, vol. 14, pp. 1187–1207, 2010.

[22] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, April 2009.

[23] D. Zaharie and D. Petcu, "Parallel implementation of multi-population differential evolution," *Concurrent Information Processing and Computing*, pp. 223–232, 2005.

[24] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[25] R. Poláková, J. Tvrdík, and P. Bujok, "Controlled restart in differential evolution applied to CEC'2014 benchmark functions," in *2014 IEEE Congress on Evolutionary Computation*, 2014.

[26] J. Brest and M. S. Maucec, "Self-adaptive differential evolution algorithm using population size reduction and three strategies," *Soft Computing*, vol. 15, pp. 2157–2174, 2011.

[27] Z. Yang, K. Tang, and X. Yao, "Scalability of generalized adaptive differential evolution for large-scale continuous optimization," *Soft Computing*, vol. 15, no. 11, pp. 2141–2155, 2011.

[28] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, pp. 945–958, 2009.

[29] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *2013 IEEE Congress on Evolutionary Computation*, pp. 71–78, 2013.

[30] R. Tanabe and A. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *2014 IEEE Congress on Evolutionary Computation*, 2014.

[31] J. Tvrdík, "Competitive differential evolution," in *12th International Coference on Soft Computing*, 2006.

[32] J. Tvrdík and R. Poláková, "Competitive differential evolution applied to CEC'2013 problems," in *2013 IEEE Congress on Evolutionary Computation*, pp. 1651–1657, IEEE, 2013.

[33] D. Zaharie, "A comparative analysis of crossover variants in differential evolution," *in Proceedings of IMCSIT*, pp. 171–181, 2007.

[34] D. Zaharie, "Influence of crossover on the behavior of differential evolution algorithms," *Applied Soft Computing*, vol. 9, no. 3, pp. 1126–1138, 2009.

[35] J. Brest, B. Bošković, and A. Zamuda, "Self-adaptive differential evolution algorithm with a small and varying population size," in *WCCI 2012 IEEE World Congress on Computational Intelligence*, 2012.

[36] A. LaTorre, S. Muelas, and J. Peña, "Multiple offspring sampling in large scale global optimization," in *2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, IEEE, 2012.

[37] A. P. Piotrowski, "Adaptive memetic differential evolution with global and local neighborhood-based mutation operators," *Information Sciences*, vol. 241, pp. 164–194, 2013.

[38] C. Segura, A. C. Coello Coello, E. Segredo, and C. León, "On the adaptation of the mutation scale factor in differential evolution," *Optimization Letters*, vol. 9, no. 1, pp. 189–198, 2015.

[39] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1362–1381, 2009.

[40] T. Cai, F. Pan, and J. Chen, "Adaptive particle swarm optimization algorithm," in *Fifth World Congress on Intelligent Control and Automation (WCICA 2004)*, vol. 3, pp. 2245–2247 Vol.3, 2004.

[41] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 101–106 vol. 1, 2001.

[42] M. Lozano, F. Herrera, and D. Molina, "Scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems," *Soft Computing*, vol. 15, pp. 2085–2087, 2011.

[43] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen, and Z. Yang, "Benchmark functions for the CEC'2008 special session and competition on large scale global optimization," *Nature Inspired Computation and Applications Laboratory, USTC, China*, pp. 153–177, 2007.

[44] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," *Foundations of Genetic Algorithms*, vol. 2, pp. 187–202, 1993.

[45] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC)*, pp. 769–1776, 2005.

[46] M. Weber, F. Neri, and V. Tirronen, "Shuffle or update parallel differential evolution for large scale optimization," *Soft Computing*, vol. 15, pp. 2089–2107, 2011.

[47] C. García-Martínez, F. J. Rodríguez, and M. Lozano, "Role differentiation and malleable mating for differential evolution: An analysis on large scale optimisation," *Soft Computing*, vol. 15, pp. 2109–2126, 2011.

[48] Z. Yang, K. Tang, and X. Yao, "Scalability of generalized adaptive differential evolution for large-scale continuous optimization," *Soft Computing*, vol. 15, pp. 2141–2155, 2011.

[49] A. LaTorre, S. Muelas, and J. Peña, "A MOS-based dynamic memetic differential evolution algorithm for continuous optimization a scalability test," *Soft Computing*, vol. 15, pp. 2187–2199, 2011.

[50] D. Molina, M. Lozano, A. M. Sánchez, and F. Herrera, "Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains," *Soft Computing*, vol. 15, pp. 2201–2220, 2011.

[51] J. García-Nieto and E. Alba, "Restart particle swarm optimization with velocity modulation: A scalability test," *Soft Computing*, vol. 15, pp. 2221–2232, 2011.

[52] M. A. Montes de Oca, D. Aydin, and T. Stützle, "An incremental particle swarm for large-scale optimization problems: An example of tuning-in-the-loop (re)design of optimization algorithms," *Soft Computing*, vol. 15, pp. 2233–2255, 2011.

[53] A. Duarte, R. Martí, and F. Gortazar, "Path relinking for large scale global optimization," *Soft Computing*, vol. 15, pp. 2257–2273, 2011.

[54] V. Gardeux, R. Chelouah, P. Siarry, and F. Glover, "EM323: A line search based algorithm for solving high-dimensional continuous non-linear optimization problems," *Soft Computing*, vol. 15, pp. 2275–2285, 2011.

[55] A. Neumaier, H. Fendl, H. Schilly, and T. Leitner, "VXQR: Derivative-free unconstrained optimization based on QR factorizations," *Soft Computing*, vol. 15, pp. 2287–2298, 2011.

[56] H. Wang, Z. Wu, and S. Rahnamayan, "Role differentiation and malleable mating for differential evolution: An analysis on large scale optimisation," *Soft Computing*, vol. 15, pp. 2127–2140, 2011.

[57] "Complementary material: Soco special issue on large scale continuous optimization problem." `https://sci2s.ugr.es/EAMHCO`.

[58] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, "Problem definitions and evaluation criteria for the CEC'2013 special session on real-parameter optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, 2013.

[59] "Complementary material: Special session & competition on real-parameter single objective optimization at CEC'2013." `http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/CEC2013.htm`.

[60] I. Poikolainen and F. Neri, "Differential evolution with concurrent fitness based local search," in *2013 IEEE Congress on Evolutionary Computation*, pp. 384–391, IEEE, 2013.

[61] J. Brest, B. Bošković, A. Zamuda, I. Fister, and E. Mezura-Montes, "Real parameter single objective optimization using self-adaptive differential evolution algorithm with more strategies," in *2013 IEEE Congress on Evolutionary Computation*, pp. 377–383, IEEE, 2013.

[62] S. M. M. Elsayed, R. A. Sarker, and T. Ray, "Differential evolution with automatic parameter configuration for solving the CEC'2013 competition on real-parameter optimization," in *2013 IEEE Congress on Evolutionary Computation*, pp. 1932–1937, IEEE, 2013.

[63] L. dos Santos Coelho, H. V. H. Ayala, and R. Z. Freire, "Population's variance-based adaptive differential evolution for real parameter optimization," in *2013 IEEE Congress on Evolutionary Computation*, pp. 1672–1677, IEEE, 2013.

[64] V. A. Tatsis and K. E. Parsopoulos, "Differential evolution with grid-based parameter adaptation," *Soft Computing*, vol. 21, no. 8, pp. 2105–2127, 2017.

[65] "Differential evolution (DE) for continuous function optimization (an algorithm by Kenneth Price and Rainer Storn)." http://www1.icsi.berkeley.edu/~storn/code.html.

[66] "Sample source code of Differential Evolution (coded by T. Takahama)." http://www.ints.info.hiroshima-cu.ac.jp/~takahama/download/DE.html.

[67] M. Lozano, F. Herrera, and D. Molina, "Evolutionary algorithms and other metaheuristics for continuous optimization problems." http://sci2s.ugr.es/eamhco/, 2010.

[68] V. A. Tatsis and K. E. Parsopoulos, "Grid search for operator and parameter control in differential evolution," in *9th Hellenic Conference on Artificial Intelligence*, SETN '16, pp. 1–9, ACM, 2016.

[69] V. A. Tatsis and K. E. Parsopoulos, "Experimental assessment of differential evolution with grid-based parameter adaptation," *International Journal on Artificial Intelligence Tools*, vol. 27, no. 04, pp. 1–20, 2018.

[70] I. Loshchilov, T. Stützle, and T. Liao, "Ranking results of CEC'2013 special session & competition on real-parameter single objective optimization," 2013.

[71] F. Caraffini, F. Neri, J. Cheng, G. Zhang, L. Picinali, G. Iacca, and E. Mininno, "Super-fit multicriteria adaptive differential evolution," in *2013 IEEE Congress on Evolutionary Computation*, pp. 1678–1685, IEEE, 2013.

[72] S. Biswas, S. Kundu, S. Das, and A. V. Vasilakos, "Teaching and learning best differential evoltuion with self adaptation for real parameter optimization," in *2013 IEEE Congress on Evolutionary Computation*, pp. 1115–1122, IEEE, 2013.

[73] J. Brest, B. Bošković, A. Zamuda, I. Fister, and E. Mezura-Montes, "Real parameter single objective optimization using self-adaptive differential evolution algorithm with more strategies," in *2013 IEEE Congress on Evolutionary Computation*, pp. 377–383, IEEE, 2013.

[74] S. M. M. Elsayed, R. A. Sarker, and T. Ray, "Differential evolution with automatic parameter configuration for solving the CEC'2013 competition on real-parameter optimization," in *2013 IEEE Congress on Evolutionary Computation*, pp. 1932–1937, IEEE, 2013.

[75] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A genetic algorithm for solving the CEC'2013 competition problems on real-parameter optimization," in *2013 IEEE Congress on Evolutionary Computation*, pp. 356–360, IEEE, 2013.

[76] L. dos Santos Coelho, H. V. H. Ayala, and R. Z. Freire, "Population's variance-based adaptive differential evolution for real parameter optimization," in *2013 IEEE Congress on Evolutionary Computation*, pp. 1672–1677, IEEE, 2013.

[77] P. Korošec and J. Šilc, "The continuous differential ant-stigmergy algorithm applied on real-parameter single objective optimization problems," in *2013 IEEE Congress on Evolutionary Computation*, pp. 1658–1663, IEEE, 2013.

[78] G. Papa and J. Šilc, "The parameter-less evolutionary search for real-parameter single objective optimization," in *2013 IEEE Congress on Evolutionary Computation*, pp. 1131–1137, IEEE, 2013.

[79] V. A. Tatsis and K. E. Parsopoulos, "On the sensitivity of the grid-based parameter adaptation method," in *7th International Conference on Metaheuristics and Nature Inspired Computing (META 2018)*, pp. 86–94, 2018.

[80] V. A. Tatsis and K. E. Parsopoulos, "Grid-based parameter adaptation in particle swarm optimization," in *12th Metaheuristics International Conference (MIC 2017)*, pp. 217–226, 2017.

[81] V. A. Tatsis and K. E. Parsopoulos, "Dynamic parameter adaptation in metaheuristics using gradient approximation and line search," *Applied Soft Computing*, vol. 74, pp. 368–384, 2019.

[82] D. A. G. Vieira and A. C. Lisboa, "Line search methods with guaranteed asymptotical convergence to an improving local optimum of multimodal functions," *European Journal of Operational Research*, vol. 235, no. 1, pp. 38 – 46, 2014.

[83] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, "Standard particle swarm optimisation 2011 at CEC'2013: A baseline for future pso improvements," in *2013 IEEE Congress on Evolutionary Computation*, pp. 2337–2344, IEEE, 2013.

Appendix A

Average errors of GPAM and GPAM*

Average errors of GPAM and GPAM* for the SOCO and CEC-2013 test suites.

Table A.1: Average errors and standard deviations of the DEGPA, eDEGPA, and the base algorithms in the SOCO suite, for dimension $n = 50$ and 100.

Problem	DEGPA		eDEGPA		DE_{bin}		DE_{exp}		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
50-dimensional												
$F1$	$8.87e-14$	$3.31e-14$	$5.46e-14$	$1.14e-14$	$3.00e-17$	$7.69e-18$	$2.78e-17$	$6.29e-33$	$2.90e+02$	$5.69e+02$	$2.78e-17$	$6.29e-33$
$F2$	$4.48e+00$	$3.15e+00$	$3.19e+00$	$4.16e+00$	$3.87e+01$	$8.90e+00$	$3.31e-01$	$5.90e-02$	$7.72e+01$	$1.23e+01$	$7.69e-11$	$4.83e-11$
$F3$	$4.59e+01$	$1.18e+01$	$5.56e+01$	$2.59e+01$	$6.99e+01$	$3.58e+01$	$3.10e+01$	$8.65e+00$	$5.64e+07$	$1.42e+08$	$6.38e-01$	$1.49e+00$
$F4$	$1.30e-13$	$2.60e-14$	$6.60e-11$	$3.29e-10$	$3.21e+01$	$1.38e+01$	$4.79e-02$	$2.01e-01$	$1.12e+02$	$2.74e+01$	$3.72e+02$	$8.68e+01$
$F5$	$4.55e-14$	$1.64e-14$	$2.73e-14$	$5.68e-15$	$9.86e-04$	$2.76e-03$	$0.00e+00$	$0.00e+00$	$9.02e-01$	$1.82e+00$	$2.16e-01$	$5.64e-01$
$F6$	$2.07e-13$	$7.32e-14$	$1.47e-13$	$9.83e-14$	$7.16e-14$	$1.86e-14$	$1.39e-13$	$9.43e-15$	$3.23e+00$	$2.44e+00$	$1.90e+01$	$1.02e+00$
$F7$	$5.53e-14$	$2.09e-13$	$0.00e+00$	$0.00e+00$	$2.22e-15$	$1.17e-15$	$8.88e-17$	$1.96e-16$	$1.23e-09$	$1.45e-09$	$2.10e+01$	$1.38e+01$
$F8$	$5.94e+01$	$1.16e+02$	$1.86e+02$	$2.59e+02$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$9.02e+06$	$9.03e+10$	$9.39e+07$
$F9$	$1.30e-04$	$6.07e-04$	$6.58e-05$	$2.98e-04$	$2.85e+02$	$5.30e+00$	$2.73e+02$	$7.40e-01$	$3.11e+02$	$4.98e+00$	$3.16e+02$	$7.03e+00$
$F10$	$3.30e-28$	$1.08e-27$	$9.12e-27$	$2.65e-26$	$1.53e+00$	$1.29e+00$	$6.50e-29$	$3.60e-29$	$7.72e+00$	$2.93e+00$	$9.25e+00$	$2.82e+00$
$F11$	$6.98e-05$	$2.81e-04$	$7.25e-05$	$3.05e-04$	$9.65e-01$	$2.02e+00$	$6.26e-05$	$1.30e-05$	$1.01e-02$	$1.26e-02$	$1.95e+02$	$3.65e+01$
$F12$	$1.08e-08$	$5.30e-08$	$1.71e-28$	$6.56e-28$	$5.82e+00$	$1.03e+01$	$5.26e-13$	$1.64e-13$	$8.23e+01$	$1.53e+02$	$1.14e+02$	$1.01e+01$
$F13$	$2.97e+01$	$4.79e+00$	$5.02e+01$	$3.29e+01$	$5.97e+01$	$2.22e+01$	$2.48e+01$	$1.31e+00$	$1.43e+07$	$3.29e+07$	$1.16e+02$	$1.43e+01$
$F14$	$3.15e-08$	$8.46e-08$	$3.34e-06$	$1.30e-05$	$3.35e+01$	$1.86e+01$	$3.55e-08$	$2.26e-08$	$6.76e+01$	$1.30e+01$	$2.71e+02$	$7.30e+01$
$F15$	$3.98e-13$	$1.99e-12$	$0.00e+00$	$0.00e+00$	$2.29e-01$	$6.07e-01$	$1.99e-24$	$3.22e-24$	$3.07e+00$	$5.32e+00$	$3.94e+01$	$1.25e+02$
$F16$	$1.04e-06$	$4.96e-06$	$2.72e-09$	$1.36e-08$	$5.64e+00$	$8.47e+00$	$1.56e-09$	$2.81e-10$	$5.60e+01$	$5.16e+01$	$2.23e+02$	$1.50e+01$
$F17$	$2.32e+00$	$2.84e+00$	$7.44e+00$	$3.19e+00$	$1.51e+01$	$1.43e+01$	$8.52e-01$	$4.92e-01$	$7.61e+06$	$2.44e+07$	$3.47e+02$	$2.18e+01$
$F18$	$9.50e-07$	$2.96e-06$	$4.67e-06$	$9.95e-06$	$5.73e+00$	$5.26e+00$	$1.28e-04$	$4.63e-05$	$6.76e+01$	$3.46e+01$	$3.59e+02$	$8.45e+01$
$F19$	$9.44e-23$	$4.71e-22$	$0.00e+00$	$0.00e+00$	$1.23e+00$	$9.26e-01$	$2.00e-24$	$1.50e-24$	$1.95e+02$	$5.01e+02$	$1.71e+03$	$5.84e+03$
100-dimensional												
$F1$	$2.34e-13$	$5.76e-14$	$5.68e-14$	$3.86e-29$	$1.12e-16$	$4.28e-17$	$7.77e-17$	$1.13e-17$	$4.67e+02$	$7.02e+02$	$5.55e-17$	$1.26e-32$
$F2$	$1.60e+01$	$7.54e+00$	$2.90e+01$	$1.50e+01$	$7.74e+01$	$7.77e+00$	$4.60e+00$	$4.24e-01$	$9.96e+01$	$1.16e+01$	$2.61e-03$	$1.30e-02$
$F3$	$1.10e+02$	$2.85e+01$	$1.36e+02$	$4.96e+01$	$4.43e+02$	$3.63e+02$	$8.01e+01$	$1.03e+01$	$1.52e+08$	$2.69e+08$	$1.23e+01$	$1.80e+01$
$F4$	$3.16e-13$	$5.93e-14$	$4.64e-13$	$8.17e-13$	$1.01e+02$	$2.25e+01$	$9.53e-03$	$4.76e-02$	$2.92e+02$	$5.16e+01$	$8.38e+02$	$1.39e+02$
$F5$	$1.15e-13$	$2.90e-14$	$3.30e-14$	$1.34e-14$	$2.93e-02$	$5.32e-02$	$2.55e-17$	$5.19e-18$	$5.95e+00$	$1.29e+01$	$2.68e+00$	$1.05e+01$
$F6$	$4.12e-13$	$9.61e-14$	$1.96e-13$	$1.50e-13$	$1.55e+00$	$3.88e-01$	$3.10e-13$	$1.62e-14$	$4.79e+00$	$1.87e+00$	$1.86e+01$	$2.45e+00$
$F7$	$6.10e-15$	$2.69e-14$	$1.53e-15$	$4.35e-15$	$1.39e-14$	$7.12e-15$	$3.80e-17$	$5.29e-17$	$8.67e-02$	$3.70e-01$	$6.35e+01$	$2.36e+01$
$F8$	$9.82e+02$	$1.46e+03$	$2.55e+03$	$3.34e+03$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$1.92e+07$	$1.80e+11$	$3.54e+08$
$F9$	$3.85e-04$	$7.11e-04$	$1.07e-03$	$2.29e-03$	$5.43e+02$	$1.36e+01$	$5.06e+02$	$9.16e-01$	$5.87e+02$	$1.01e+01$	$6.08e+02$	$1.07e+01$
$F10$	$1.08e-26$	$5.39e-26$	$3.53e-30$	$1.55e-29$	$1.54e+01$	$3.31e+00$	$1.35e-28$	$3.86e-29$	$2.89e+01$	$1.01e+01$	$1.93e+01$	$5.10e+00$
$F11$	$6.60e-04$	$2.17e-03$	$1.01e-03$	$2.65e-03$	$4.31e+01$	$2.09e+01$	$1.25e-04$	$1.43e-05$	$2.80e+01$	$3.02e+01$	$4.82e+02$	$4.27e+01$
$F12$	$2.31e-07$	$9.00e-07$	$1.43e-02$	$7.15e-02$	$7.21e+01$	$3.21e+01$	$6.44e-11$	$1.52e-11$	$8.72e+02$	$2.55e+03$	$2.41e+02$	$1.23e+01$
$F13$	$7.23e+01$	$1.89e+01$	$9.07e+01$	$3.09e+01$	$2.76e+02$	$6.18e+01$	$6.13e+01$	$1.00e+00$	$9.37e+07$	$4.02e+08$	$2.59e+02$	$2.16e+01$
$F14$	$3.58e-08$	$8.99e-08$	$1.78e-06$	$2.78e-06$	$9.37e+01$	$1.56e+01$	$4.48e-02$	$2.24e-01$	$2.25e+02$	$4.59e+01$	$6.19e+02$	$9.25e+01$
$F15$	$2.09e-15$	$1.04e-14$	$5.97e-10$	$2.07e-09$	$3.67e+00$	$1.76e+00$	$7.10e-23$	$7.00e-23$	$5.99e+00$	$1.19e+01$	$5.57e+01$	$5.22e+01$
$F16$	$3.05e-06$	$1.16e-05$	$1.48e-08$	$4.17e-08$	$1.10e+02$	$3.80e+01$	$1.94e-02$	$9.70e-02$	$2.08e+02$	$1.49e+02$	$4.84e+02$	$2.08e+01$
$F17$	$2.27e+01$	$2.76e+01$	$3.43e+01$	$2.64e+01$	$1.78e+02$	$5.49e+01$	$1.19e+01$	$2.62e+00$	$4.36e+07$	$7.09e+07$	$7.04e+02$	$3.92e+01$
$F18$	$2.56e-06$	$1.23e-05$	$8.40e-06$	$2.40e-05$	$1.04e+02$	$4.39e+01$	$2.92e-04$	$6.77e-05$	$2.37e+02$	$7.02e+01$	$1.09e+03$	$4.15e+02$
$F19$	$2.69e-22$	$1.34e-21$	$1.18e-31$	$4.83e-31$	$1.17e+01$	$2.61e+00$	$4.79e-23$	$2.65e-23$	$4.70e+02$	$1.84e+03$	$5.83e+03$	$9.85e+03$

Table A.2: Average errors and standard deviations of the DEGPA, eDEGPA, and the base algorithms in the SOCO suite, for dimension $n = 200$ and 500.

Problem	DEGPA		eDEGPA		DE_{bin}		DE_{exp}		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
200-dimensional												
$F1$	$5.50e-13$	$1.07e-13$	$1.32e-13$	$8.32e-14$	$6.39e-16$	$5.32e-16$	$1.78e-16$	$1.60e-17$	$9.61e+02$	$1.65e+03$	$1.17e-16$	$1.13e-17$
$F2$	$3.67e+01$	$1.17e+01$	$6.22e+01$	$2.00e+01$	$1.01e+02$	$5.90e+00$	$1.89e+01$	$1.05e+00$	$1.17e+02$	$7.60e+00$	$7.47e-02$	$2.44e-01$
$F3$	$2.17e+02$	$3.18e+01$	$2.56e+02$	$8.22e+01$	$6.38e+02$	$3.80e+02$	$1.79e+02$	$8.89e+00$	$2.54e+08$	$3.97e+08$	$1.24e+02$	$8.85e+01$
$F4$	$8.00e-04$	$4.00e-03$	$4.90e-09$	$9.12e-09$	$4.21e+02$	$5.63e+01$	$8.52e-02$	$3.98e-01$	$6.32e+02$	$8.43e+01$	$1.57e+03$	$1.54e+02$
$F5$	$2.60e-13$	$2.80e-14$	$7.21e-07$	$3.61e-06$	$3.00e-01$	$7.73e-01$	$7.49e-17$	$6.94e-18$	$1.02e+01$	$1.59e+01$	$1.13e+00$	$2.84e+00$
$F6$	$8.39e-13$	$2.35e-13$	$3.73e-13$	$3.20e-13$	$5.28e+00$	$9.65e-01$	$6.46e-13$	$2.53e-14$	$8.14e+00$	$2.26e+00$	$1.93e+01$	$7.39e-01$
$F7$	$7.87e-14$	$3.74e-13$	$6.55e-15$	$2.35e-14$	$1.78e-11$	$4.65e-11$	$2.25e-16$	$1.92e-16$	$3.95e-01$	$1.21e+00$	$1.25e+02$	$1.92e+01$
$F8$	$1.10e+04$	$1.43e+04$	$2.79e+04$	$2.70e+04$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$3.09e+08$	$8.56e+11$	$3.36e+09$
$F9$	$9.47e-05$	$3.00e-04$	$6.28e-03$	$1.48e-02$	$1.13e+03$	$1.87e+01$	$1.01e+03$	$1.30e+00$	$1.18e+03$	$8.29e+00$	$1.22e+03$	$1.79e+01$
$F10$	$1.62e-30$	$7.01e-30$	$1.63e-30$	$7.46e-30$	$5.52e+01$	$1.02e+01$	$2.77e-28$	$5.31e-29$	$7.34e+01$	$6.25e+01$	$3.76e+01$	$2.78e+01$
$F11$	$7.82e-04$	$2.26e-03$	$5.31e-03$	$1.24e-02$	$3.95e+02$	$6.11e+01$	$2.55e-04$	$3.20e-05$	$4.03e+02$	$8.45e+01$	$1.08e+03$	$8.25e+01$
$F12$	$1.49e-06$	$6.91e-06$	$3.72e-03$	$1.37e-02$	$2.84e+02$	$4.97e+01$	$9.97e-10$	$2.01e-10$	$8.11e+02$	$1.58e+03$	$5.87e+02$	$4.52e+02$
$F13$	$1.40e+02$	$1.68e+01$	$1.95e+02$	$6.12e+01$	$7.52e+02$	$2.71e+02$	$1.40e+02$	$1.26e+01$	$2.06e+08$	$3.51e+08$	$5.92e+02$	$1.08e+02$
$F14$	$1.55e-08$	$3.27e-08$	$1.38e-07$	$4.05e-07$	$3.11e+02$	$3.66e+01$	$8.08e-03$	$4.04e-02$	$4.90e+02$	$5.23e+01$	$1.26e+03$	$1.81e+02$
$F15$	$2.78e-18$	$1.39e-17$	$4.64e-14$	$2.07e-13$	$1.17e+01$	$2.85e+00$	$3.71e-24$	$2.32e-24$	$1.40e+01$	$9.80e+00$	$1.95e+02$	$1.66e+02$
$F16$	$1.71e-06$	$6.61e-06$	$1.33e-03$	$4.71e-03$	$5.58e+02$	$7.96e+01$	$7.85e-09$	$1.11e-09$	$6.77e+02$	$6.04e+02$	$9.56e+02$	$3.33e+01$
$F17$	$6.24e+01$	$3.58e+01$	$6.29e+01$	$3.10e+01$	$1.03e+03$	$1.11e+02$	$3.71e+01$	$8.30e-01$	$1.17e+07$	$1.70e+07$	$1.49e+03$	$8.00e+01$
$F18$	$1.06e-05$	$5.14e-05$	$1.67e-04$	$6.00e-04$	$7.53e+02$	$6.55e+01$	$5.10e-04$	$9.97e-05$	$7.67e+02$	$2.14e+02$	$3.94e+03$	$3.91e+03$
$F19$	$2.29e-17$	$1.15e-16$	$4.20e-02$	$2.10e-01$	$4.04e+01$	$7.71e+00$	$1.67e-22$	$7.58e-23$	$7.51e+02$	$1.76e+03$	$2.53e+04$	$2.45e+04$
500-dimensional												
$F1$	$1.43e-12$	$1.58e-13$	$5.59e-13$	$2.32e-13$	$3.88e-05$	$7.93e-05$	$5.17e-16$	$1.36e-17$	$9.25e+02$	$1.27e+03$	n/a	n/a
$F2$	$7.97e+01$	$1.28e+01$	$7.31e+01$	$1.85e+01$	$1.25e+02$	$5.37e+00$	$5.38e+01$	$1.21e+00$	$1.35e+02$	$5.52e+00$	n/a	n/a
$F3$	$4.96e+02$	$2.01e+01$	$6.22e+02$	$1.69e+02$	$3.44e+04$	$1.62e+05$	$4.74e+02$	$1.48e+00$	$6.93e+08$	$1.72e+09$	n/a	n/a
$F4$	$2.08e-12$	$3.22e-13$	$3.98e-02$	$1.99e-01$	$2.35e+03$	$1.60e+02$	$7.12e-01$	$9.64e-01$	$2.11e+03$	$1.66e+02$	n/a	n/a
$F5$	$7.21e-13$	$1.00e-13$	$3.60e-13$	$1.79e-13$	$3.11e-01$	$5.07e-01$	$2.38e-16$	$1.18e-17$	$1.45e+01$	$2.52e+01$	n/a	n/a
$F6$	$2.08e-12$	$3.66e-13$	$1.24e-01$	$3.30e-01$	$1.49e+01$	$8.38e-01$	$1.64e-12$	$4.85e-14$	$1.27e+01$	$1.26e+00$	n/a	n/a
$F7$	$4.02e-18$	$2.01e-17$	$3.64e-03$	$1.82e-02$	$2.74e-03$	$6.49e-03$	$7.29e-16$	$3.58e-16$	$3.33e-05$	$1.12e-04$	n/a	n/a
$F8$	$9.55e+04$	$8.31e+04$	$1.77e+05$	$1.53e+05$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$1.42e+08$	n/a	n/a
$F9$	$2.32e-03$	$4.18e-03$	$2.10e-02$	$6.44e-02$	$2.97e+03$	$3.17e+01$	$2.52e+03$	$2.10e+00$	$3.00e+03$	$1.64e+01$	n/a	n/a
$F10$	$0.00e+00$	$0.00e+00$	$1.45e-32$	$7.23e-32$	$1.36e+02$	$2.08e+01$	$9.79e-28$	$1.43e-28$	$1.64e+02$	$5.62e+01$	n/a	n/a
$F11$	$2.16e-03$	$4.28e-03$	$2.78e-03$	$6.39e-03$	$2.34e+03$	$9.22e+01$	$6.78e-04$	$3.60e-05$	$1.67e+03$	$1.44e+02$	n/a	n/a
$F12$	$4.24e-06$	$1.45e-05$	$1.66e-04$	$7.86e-04$	$1.02e+03$	$6.68e+01$	$6.80e-09$	$8.58e-10$	$1.62e+03$	$1.83e+03$	n/a	n/a
$F13$	$3.71e+02$	$2.14e+01$	$5.77e+02$	$1.77e+02$	$2.49e+03$	$3.13e+02$	$3.60e+02$	$9.23e+00$	$3.41e+08$	$4.29e+08$	n/a	n/a
$F14$	$1.60e-08$	$3.23e-08$	$3.30e-03$	$1.61e-02$	$1.67e+03$	$1.51e+02$	$3.93e-01$	$1.05e+00$	$1.59e+03$	$1.57e+02$	n/a	n/a
$F15$	$2.33e-15$	$8.81e-15$	$5.50e-01$	$1.77e+00$	$4.44e+01$	$5.59e+00$	$2.93e-18$	$7.16e-18$	$3.50e+01$	$1.20e+01$	n/a	n/a
$F16$	$1.71e-05$	$4.77e-05$	$1.46e+01$	$7.27e+01$	$2.02e+03$	$8.60e+01$	$2.05e-08$	$1.64e-09$	$1.92e+03$	$1.44e+03$	n/a	n/a
$F17$	$1.41e+02$	$3.61e+01$	$2.67e+02$	$5.51e+02$	$3.83e+03$	$1.41e+02$	$1.12e+02$	$1.02e+00$	$6.64e+08$	$1.64e+09$	n/a	n/a
$F18$	$7.67e-06$	$2.90e-05$	$1.19e-01$	$3.30e-01$	$3.37e+03$	$4.34e+02$	$1.25e-03$	$1.87e-04$	$2.74e+03$	$3.59e+02$	n/a	n/a
$F19$	$3.18e-27$	$9.97e-27$	$2.16e+00$	$3.41e+00$	$1.29e+02$	$2.34e+01$	$3.35e-21$	$2.15e-21$	$2.05e+03$	$4.03e+03$	n/a	n/a

Table A.3: Average errors of the DEGPA, eDEGPA, and other algorithms provided in the SOCO test suite.

		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
DEGPA																				
	50	$0.00e+00$	$4.48e+00$	$4.59e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$5.94e+01$	$1.30e-04$	$0.00e+00$	$6.98e-05$	$1.08e-08$	$2.97e+01$	$3.15e-08$	$0.00e+00$	$1.03e-06$	$2.32e+00$	$9.50e-07$	$0.00e+00$	
	100	$0.00e+00$	$1.60e+01$	$1.10e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.82e+02$	$3.85e-04$	$0.00e+00$	$6.60e-04$	$2.31e-07$	$7.23e+01$	$3.58e-08$	$0.00e+00$	$3.05e-06$	$2.27e+01$	$2.56e-06$	$0.00e+00$	
	200	$0.00e+00$	$3.67e+01$	$2.17e+02$	$8.00e-04$	$0.00e+00$	$0.00e+00$	$1.10e+04$	$9.47e-05$	$0.00e+00$	$7.82e-04$	$1.49e-06$	$1.40e+02$	$1.55e-08$	$0.00e+00$	$1.71e-06$	$6.24e+01$	$1.06e-05$	$0.00e+00$	
	500	$0.00e+00$	$7.97e+01$	$4.96e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.55e+04$	$2.32e-03$	$0.00e+00$	$2.16e-03$	$4.24e-06$	$3.71e+02$	$1.60e-08$	$0.00e+00$	$1.71e-05$	$1.41e+02$	$7.67e-06$	$0.00e+00$	
eDEGPA																				
	50	$0.00e+00$	$3.19e+00$	$5.56e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.86e+02$	$6.58e-05$	$0.00e+00$	$7.25e-05$	$0.00e+00$	$5.02e+01$	$3.34e-06$	$0.00e+00$	$0.00e+00$	$7.44e+00$	$4.67e-06$	$0.00e+00$	
	100	$0.00e+00$	$2.90e+01$	$1.36e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$2.55e+03$	$1.07e-03$	$0.00e+00$	$1.01e-03$	$1.43e-02$	$9.07e+01$	$1.78e-06$	$0.00e+00$	$1.48e-08$	$3.43e+01$	$8.40e-06$	$0.00e+00$	
	200	$0.00e+00$	$6.22e+01$	$2.56e+02$	$0.00e+00$	$7.21e-07$	$0.00e+00$	$2.79e+04$	$6.28e-03$	$0.00e+00$	$5.31e-03$	$3.72e-03$	$1.95e+02$	$1.38e-07$	$0.00e+00$	$1.33e-03$	$6.29e+01$	$1.67e-04$	$4.20e-02$	
	500	$0.00e+00$	$7.31e+01$	$6.22e+02$	$3.98e-02$	$0.00e+00$	$1.24e-01$	$3.64e-03$	$1.77e+05$	$2.10e-02$	$0.00e+00$	$2.78e-03$	$1.66e-04$	$5.77e+02$	$3.30e-03$	$5.50e-01$	$1.46e+01$	$2.67e+02$	$1.19e-01$	$2.16e+00$
SOUPDE																				
	50	$0.00e+000$	$1.18e+000$	$3.10e+001$	$3.98e-002$	$0.00e+000$	$1.47e-014$	$2.28e-014$	$9.69e-002$	$3.75e-006$	$0.00e+000$	$3.09e-006$	$0.00e+000$	$2.06e+001$	$0.00e+000$	$1.38e-014$	$0.00e+000$	$2.53e-001$	$0.00e+000$	$0.00e+000$
	100	$0.00e+000$	$7.47e+000$	$7.92e+001$	$3.98e-002$	$0.00e+000$	$3.03e-014$	$3.88e-014$	$6.55e+001$	$7.82e-006$	$0.00e+000$	$6.75e-006$	$0.00e+000$	$5.85e+001$	$9.09e-015$	$2.79e-014$	$0.00e+000$	$8.55e+000$	$0.00e+000$	$0.00e+000$
	200	$0.00e+000$	$2.38e+001$	$1.80e+002$	$1.19e-001$	$0.00e+000$	$6.40e-014$	$7.46e-014$	$2.46e+003$	$1.51e-005$	$0.00e+000$	$1.43e-005$	$0.00e+000$	$1.35e+002$	$3.98e-002$	$5.79e-014$	$0.00e+000$	$3.31e+001$	$0.00e+000$	$1.91e-014$
	500	$0.00e+000$	$6.50e+001$	$4.71e+002$	$7.96e-002$	$0.00e+000$	$1.67e-013$	$1.78e-013$	$4.36e+004$	$3.59e-005$	$0.00e+000$	$4.66e-004$	$0.00e+000$	$3.58e+002$	$1.31e-012$	$1.39e-013$	$0.00e+000$	$1.09e+002$	$2.82e-013$	$4.95e-014$
DE-$D^{40}+M^m$																				
	50	$3.33e-018$	$1.67e-001$	$1.34e+001$	$1.99e-001$	$0.00e+000$	$4.55e-014$	$0.00e+000$	$6.11e-001$	$0.00e+000$	$1.89e-031$	$6.06e-004$	$1.58e-021$	$1.39e+001$	$1.19e-001$	$0.00e+000$	$1.76e-016$	$4.31e-002$	$3.98e-002$	$0.00e+000$	
	100	$2.78e-017$	$2.24e+000$	$7.61e+001$	$1.99e-001$	$1.39e-017$	$1.01e-013$	$5.33e-017$	$4.75e+005$	$4.29e-004$	$0.00e+000$	$0.00e+000$	$4.62e-017$	$5.33e+001$	$1.19e-001$	$0.00e+000$	$8.99e-015$	$3.22e+000$	$8.11e-010$	$0.00e+000$	
	200	$6.66e-017$	$9.58e+000$	$1.69e+002$	$2.39e-001$	$2.78e-017$	$2.51e-013$	$0.00e+000$	$2.19e+008$	$0.00e+000$	$3.51e+001$	$0.00e+000$	$5.45e-015$	$1.23e+002$	$3.98e-002$	$0.00e+000$	$2.26e-013$	$2.72e+001$	$3.98e-002$	$9.47e-032$
	500	$2.23e-016$	$3.72e+001$	$4.54e+002$	$9.15e-001$	$1.03e-016$	$7.14e-013$	$0.00e+000$	$1.41e+010$	$6.78e-009$	$2.43e-031$	$0.00e+000$	$5.05e-013$	$3.48e+002$	$2.39e-001$	$0.00e+000$	$4.17e-012$	$1.02e+002$	$9.97e-008$	$0.00e+000$
GaDE																				
	50	$0.00e+000$	$1.46e+001$	$1.18e+001$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$1.08e-008$	$6.24e-007$	$0.00e+000$	$1.31e-006$	$0.00e+000$	$1.19e+001$	$9.78e-013$	$0.00e+000$	$4.78e-012$	$4.97e-001$	$4.82e-008$	$0.00e+000$	
	100	$0.00e+000$	$3.88e+001$	$5.89e+001$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$1.23e-003$	$3.87e-007$	$0.00e+000$	$4.34e-007$	$0.00e+000$	$4.99e+001$	$7.90e-013$	$0.00e+000$	$2.45e-012$	$3.28e+000$	$1.96e-008$	$0.00e+000$	
	200	$0.00e+000$	$5.76e+001$	$1.61e+001$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$3.02e+000$	$4.53e-009$	$4.20e-002$	$1.85e-007$	$4.92e-014$	$1.24e+002$	$2.87e-012$	$0.00e+000$	$1.58e-012$	$2.45e+001$	$2.53e-008$	$0.00e+000$	
	500	$0.00e+000$	$7.42e+001$	$4.40e+002$	$0.00e+000$	$0.00e+000$	$1.46e-014$	$0.00e+000$	$1.33e+003$	$0.00e+000$	$3.78e-001$	$0.00e+000$	$1.07e-012$	$3.34e+002$	$2.79e-011$	$0.00e+000$	$1.67e-012$	$9.26e+001$	$5.59e-008$	$4.20e-002$
jDElscop																				
	50	$0.00e+00$	$3.15e-02$	$2.28e+01$	$0.00e+00$	$0.00e+00$	$9.55e-14$	$0.00e+00$	$9.97e-03$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.36e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$7.43e-03$	$2.41e-14$	$0.00e+00$	
	100	$0.00e+00$	$1.21e+00$	$6.13e+01$	$0.00e+00$	$0.00e+00$	$2.00e-13$	$0.00e+00$	$5.57e+00$	$7.18e-09$	$0.00e+00$	$8.17e-09$	$0.00e+00$	$5.11e+01$	$0.00e+00$	$0.00e+00$	$3.21e-01$	$6.33e-14$	$0.00e+00$	
	200	$0.00e+00$	$7.54e+00$	$1.40e+02$	$0.00e+00$	$0.00e+00$	$4.52e-13$	$0.00e+00$	$2.52e+02$	$4.30e-08$	$0.00e+00$	$9.58e-09$	$0.00e+00$	$1.10e+02$	$4.11e-16$	$0.00e+00$	$2.39e+01$	$2.04e-13$	$0.00e+00$	
	500	$0.00e+00$	$3.06e+01$	$4.06e+02$	$1.59e-01$	$0.00e+00$	$1.18e-12$	$0.00e+00$	$5.66e+03$	$6.10e-08$	$0.00e+00$	$4.40e-08$	$0.00e+00$	$3.14e+02$	$8.00e-02$	$0.00e+00$	$7.65e+01$	$1.11e-12$	$0.00e+00$	
SaDE-MMTS																				
	50	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$4.13e-09$	$1.35e-01$	$0.00e+00$	$5.19e-05$	$0.00e+00$	$4.23e+00$	$3.93e-08$	$0.00e+00$	$0.00e+00$	$4.78e-01$	$9.38e-03$	$0.00e+00$	
	100	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.05e-04$	$3.18e-01$	$0.00e+00$	$2.00e-04$	$0.00e+00$	$3.30e+01$	$1.02e-02$	$0.00e+00$	$0.00e+00$	$1.17e+01$	$4.70e-02$	$0.00e+00$	
	200	$0.00e+00$	$1.34e+00$	$0.00e+00$	$8.08e-02$	$0.00e+00$	$0.00e+00$	$2.67e+01$	$1.24e+00$	$0.00e+00$	$2.39e-04$	$0.00e+00$	$8.89e+01$	$1.57e-02$	$0.00e+00$	$0.00e+00$	$3.50e+01$	$3.35e-01$	$0.00e+00$	
	500	$0.00e+00$	$1.25e+01$	$0.00e+00$	$3.85e+00$	$0.00e+00$	$0.00e+00$	$3.01e+02$	$2.81e+01$	$0.00e+00$	$2.53e+01$	$0.00e+00$	$3.27e+02$	$4.01e-01$	$0.00e+00$	$0.00e+00$	$9.80e+01$	$1.18e+00$	$0.00e+00$	
MOS																				
	50	$0.00e+00$	$4.64e-13$	$9.61e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.54e-08$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$4.55e-01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.40e+01$	$0.00e+00$	$0.00e+00$	
	100	$0.00e+00$	$2.94e-12$	$2.03e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.17e-02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.75e+01$	$1.68e-11$	$0.00e+00$	$0.00e+00$	$1.43e+01$	$0.00e+00$	$0.00e+00$	
	200	$0.00e+00$	$1.24e-11$	$4.01e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.16e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.03e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$5.03e+00$	$0.00e+00$	$0.00e+00$	
	500	$0.00e+00$	$5.51e-04$	$4.57e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.28e+04$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.78e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.21e+01$	$0.00e+00$	$0.00e+00$	
MA-SSW-Chains																				
	50	$1.67e-17$	$7.61e-02$	$4.79e+01$	$1.19e-01$	$0.00e+00$	$4.89e-14$	$9.33e-17$	$3.06e-01$	$2.94e+02$	$1.67e-30$	$4.49e-03$	$6.27e-41$	$3.02e+01$	$1.37e-17$	$3.91e-16$	$4.06e-03$	$2.60e+01$	$3.88e-19$	$4.02e-31$
	100	$2.78e-17$	$7.01e+00$	$1.38e+02$	$1.19e-01$	$1.39e-17$	$6.03e-14$	$8.17e-16$	$3.48e+01$	$5.63e+02$	$1.05e-29$	$1.09e-01$	$3.28e-03$	$8.35e+01$	$2.21e-16$	$1.59e-15$	$1.61e-02$	$9.92e+01$	$2.71e-18$	$3.15e-30$
	200	$5.33e-17$	$3.36e+01$	$2.50e+02$	$4.43e+00$	$2.72e-17$	$1.19e-13$	$6.96e-15$	$7.23e+02$	$1.17e+03$	$5.41e-29$	$3.50e-01$	$1.75e-02$	$1.68e+02$	$9.76e-01$	$5.32e-15$	$6.02e-02$	$7.55e+01$	$4.29e-04$	$1.51e-16$
	500	$1.01e-16$	$7.86e+01$	$6.07e+02$	$1.78e+02$	$7.70e-17$	$2.63e-13$	$4.69e-14$	$1.32e+04$	$2.53e+03$	$2.80e-01$	$4.21e+01$	$2.55e+01$	$4.00e+02$	$5.65e+01$	$5.53e+00$	$1.08e-01$	$1.38e+02$	$2.41e-03$	$7.84e-17$

Table A.4: Average errors of the DEGPA, eDEGPA, and other algorithms provided in the SOCO test suite.

		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
DEGPA	50	0.00e+00	4.48e+00	4.59e+01	0.00e+00	0.00e+00	0.00e+00	5.94e+01	1.30e−04	0.00e+00	6.98e−05	1.08e−08	2.97e+01	3.15e−08	0.00e+00	1.03e−06	2.32e+00	9.50e−07	0.00e+00	
	100	0.00e+00	1.60e+01	1.10e+02	0.00e+00	0.00e+00	0.00e+00	9.82e+02	3.85e−04	0.00e+00	6.60e−04	2.31e−07	7.23e+01	3.58e−08	0.00e+00	3.05e−06	2.27e+01	2.56e−06	0.00e+00	
	200	0.00e+00	3.67e+01	2.17e+02	8.00e−04	0.00e+00	0.00e+00	1.10e+04	9.47e−05	0.00e+00	7.82e−04	1.49e−06	1.40e+02	1.55e−08	0.00e+00	1.71e−06	6.24e+01	1.06e−05	0.00e+00	
	500	0.00e+00	7.97e+01	4.96e+02	0.00e+00	0.00e+00	0.00e+00	9.55e+04	2.32e−03	0.00e+00	2.16e−03	4.24e−06	3.71e+02	1.60e−08	0.00e+00	1.71e−05	1.41e+02	7.67e−06	0.00e+00	
eDEGPA	50	0.00e+00	3.19e+00	5.56e+01	0.00e+00	0.00e+00	0.00e+00	1.86e+02	6.58e−05	0.00e+00	7.25e−05	0.00e+00	5.02e+01	3.34e−06	0.00e+00	0.00e+00	7.44e+00	4.67e−06	0.00e+00	
	100	0.00e+00	2.90e+01	1.36e+02	0.00e+00	0.00e+00	0.00e+00	2.55e+03	1.07e−03	0.00e+00	1.01e−03	1.43e−02	9.07e+01	1.78e−06	0.00e+00	1.48e−08	3.43e+01	8.40e−06	0.00e+00	
	200	0.00e+00	6.22e+01	2.56e+02	0.00e+00	7.21e−07	0.00e+00	2.79e+04	6.28e−03	0.00e+00	5.31e−03	3.72e−03	1.95e+02	1.38e−07	0.00e+00	1.33e−03	6.29e+01	1.67e−04	4.20e−02	
	500	0.00e+00	7.31e+01	6.22e+02	3.98e−02	0.00e+00	1.24e−01	3.64e−03	1.77e+05	2.10e−02	0.00e+00	2.78e−03	1.66e−04	5.77e+02	3.30e−03	5.50e−01	1.46e+01	2.67e+02	1.19e−01	2.16e+00
RPSO-vm	50	2.62e−14	7.54e−03	1.75e+03	2.30e−14	9.53e−02	1.49e−12	1.14e−15	1.06e+03	2.94e−01	0.00e+00	1.68e−02	8.58e−02	6.57e+02	6.81e−02	0.00e+00	7.88e−11	8.73e+02	5.05e−02	0.00e+00
	100	2.66e−14	1.98e−01	1.42e+03	2.61e−14	1.07e−01	4.76e−12	0.00e+00	1.09e+04	1.85e−01	0.00e+00	4.61e−01	1.60e−14	2.25e+03	1.27e−01	0.00e+00	4.87e−08	1.76e+03	1.36e−01	0.00e+00
	200	2.53e−14	2.00e+00	1.03e+03	2.43e−14	1.73e−01	2.95e−12	0.00e+00	5.23e+04	1.67e+00	0.00e+00	5.66e−01	4.64e−02	1.17e+04	7.96e−02	0.00e+00	5.40e−01	2.08e+04	1.50e−01	0.00e+00
	500	2.65e−14	1.67e+01	1.13e+03	2.44e−14	2.54e−01	3.14e−12	8.90e−16	3.00e+05	4.85e+00	0.00e+00	4.88e+00	1.32e−08	1.33e+03	1.29e+00	0.00e+00	2.12e+00	5.72e+02	2.47e+00	0.00e+00
Tuned IPSOLS	50	0.00e+00	2.56e−014	0.00e+00	0.00e+00	6.72e−03	0.00e+00	4.98e−012	4.78e−009	4.95e−006	0.00e+00	8.19e−002	1.17e−011	2.65e−010	1.18e+00	2.62e−011	2.80e+00	3.10e+00	1.24e+00	1.19e−011
	100	0.00e+00	3.42e−014	0.00e+00	0.00e+00	1.31e−02	0.00e+00	3.00e−012	7.18e−003	1.09e+00	0.00e+00	1.48e−003	2.58e−011	1.59e−001	4.34e+00	2.89e−011	9.09e−012	8.16e−003	1.68e+00	7.33e−012
	200	0.00e+00	3.23e−014	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.14e−011	2.47e+00	4.09e+00	0.00e+00	6.98e+00	6.65e−012	2.55e+00	4.27e+00	1.08e−010	1.65e+00	1.15e+00	4.34e+00	1.11e−011
	500	0.00e+00	8.07e−014	0.00e+00	2.82e−003	0.00e+00	0.00e+00	1.06e−011	8.64e+03	1.13e+01	0.00e+00	1.14e+001	1.93e−011	4.14e+00	1.30e+00	2.83e−010	1.33e+00	1.33e+00	1.33e+001	2.73e−011
EvoPROpt	50	1.22e−02	3.71e−01	1.12e+02	4.96e−02	5.13e−02	6.85e−03	2.63e−02	2.08e+02	8.02e+00	4.80e−02	9.68e+00	2.27e+00	4.22e+01	9.97e−01	6.38e−02	5.63e+00	6.77e+01	1.62e+00	5.03e−02
	100	4.34e−02	3.30e+00	3.98e+02	1.07e−01	3.92e−02	2.50e−04	9.17e−02	2.27e+03	2.91e+01	2.05e−01	2.60e+01	5.01e+00	1.40e+02	1.24e+00	6.56e−02	8.29e+00	1.97e+02	3.34e+00	1.43e−01
	200	8.03e−02	8.03e+00	2.91e+02	3.52e−01	2.68e−02	6.22e−01	3.82e−02	1.34e+04	6.22e+01	1.04e+00	5.93e+01	1.00e+01	1.71e+02	3.75e+00	3.80e−01	1.74e+01	1.56e+02	8.85e+00	2.15e+00
	500	0.00e+00	2.04e+01	5.97e+02	1.45e+00	3.03e−02	1.21e+00	8.06e−03	7.05e+04	1.75e+02	3.29e+01	1.77e+02	1.73e+01	5.75e+02	9.00e+00	2.25e+00	4.87e+01	3.94e+02	3.28e+01	5.00e+01
EM323	50	4.80e−13	4.08e−09	6.12e+01	5.34e−13	3.05e−13	5.66e−13	0.00e+00	2.08e+02	0.00e+00	0.00e+00	2.16e−07	9.88e−08	1.96e+01	1.32e−07	0.00e+00	2.52e−07	8.58e+01	3.12e−07	0.00e+00
	100	9.91e−13	3.42e−04	2.10e+02	1.15e−12	5.91e−13	1.14e−12	0.00e+00	6.31e+02	3.29e−08	0.00e+00	9.99e−09	1.51e−02	5.97e+01	3.74e−07	0.00e+00	4.57e−07	9.60e+01	5.02e−02	0.00e+00
	200	2.15e−12	1.92e−01	4.47e+02	2.23e−12	3.95e−04	2.42e−12	0.00e+00	3.37e+04	1.31e−08	0.00e+00	1.92e−06	4.13e−07	2.97e+01	2.10e−01	0.00e+00	9.40e−07	5.73e+01	2.63e−03	0.00e+00
	500	5.80e−12	2.04e+01	1.25e+03	7.08e−12	1.77e−03	6.50e−12	0.00e+00	3.91e+05	2.21e−06	0.00e+00	1.10e−02	3.41e−01	1.19e+03	1.51e−01	0.00e+00	4.71e−03	2.14e+02	2.28e−01	0.00e+00
VXQR1	50	0.00e+00	2.36e+00	3.69e−09	0.00e+00	4.55e−14	3.07e−13	0.00e+00	0.00e+00	9.62e+00	0.00e+00	1.09e+01	1.49e+00	8.39e+00	2.58e−01	3.72e−05	4.48e+00	1.73e+01	6.94e−01	0.00e+00
	100	0.00e+00	5.12e+01	3.19e−01	0.00e+00	6.03e−14	9.44e−13	0.00e+00	2.15e+01	0.00e+00	1.89e+01	5.24e+00	1.78e+01	6.41e−01	7.14e−04	1.18e+01	1.14e+02	1.26e+00	1.49e−06	
	200	0.00e+00	8.50e+01	3.60e+01	2.50e−14	1.18e−03	3.10e−12	0.00e+00	0.00e+00	4.54e+01	0.00e+00	4.37e+01	4.00e+01	9.55e+01	1.27e+00	1.09e−01	3.00e+01	8.79e+01	8.48e+00	3.56e−05
	500	0.00e+00	9.38e+01	2.14e+02	1.86e−13	2.96e−04	9.10e−12	0.00e+00	0.00e+00	8.68e+01	0.00e+00	8.52e+01	1.21e+02	2.21e+02	4.31e+00	2.24e−01	6.32e+01	1.65e+02	5.52e+01	1.14e−03
GODE	50	0.00e+000	2.57e−001	3.06e+001	1.05e−013	0.00e+000	1.24e−014	0.00e+000	1.67e−001	7.77e−006	0.00e+000	6.44e−006	1.33e−013	2.55e+001	6.24e−009	0.00e+000	1.57e−010	1.17e+000	2.97e−007	0.00e+000
	100	0.00e+000	3.65e+000	8.14e+001	8.32e−014	0.00e+000	2.60e−014	0.00e+000	7.53e+001	1.46e−005	0.00e+000	1.58e−005	7.57e−012	6.32e+001	4.13e−008	0.00e+000	3.75e−010	1.11e+001	1.11e−006	0.00e+000
	200	0.00e+000	1.53e+001	1.80e+002	4.17e−013	0.00e+000	5.45e−014	0.00e+000	2.10e+003	3.23e−005	0.00e+000	3.12e−005	1.20e−010	1.38e+002	8.17e−002	0.00e+000	9.54e−010	3.74e+001	1.91e−006	0.00e+000
	500	0.00e+000	5.81e+001	4.76e+002	1.62e−003	0.00e+000	1.43e−013	0.00e+000	3.93e+004	7.84e−005	0.00e+000	8.25e−005	7.39e−010	3.59e+002	7.67e−002	0.00e+000	2.24e−009	1.12e+002	5.06e−006	0.00e+000

Table A.5: Average errors and standard deviations of the DEGPOA, eDEGPOA, and the base algorithms in the SOCO suite, for dimension $n = 50$ and 100.

Problem	DEGPOA		eDEGPOA		DE_{bin}		DE_{exp}		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
50-dimensional												
$F1$	$9.09e-14$	$3.67e-14$	$1.30e-13$	$3.86e-14$	$3.00e-17$	$7.69e-18$	$2.78e-17$	$6.29e-33$	$2.90e+02$	$5.69e+02$	$2.78e-17$	$6.29e-33$
$F2$	$6.25e+00$	$5.56e+00$	$7.06e+00$	$7.05e+00$	$3.87e+01$	$8.90e+00$	$3.31e-01$	$5.90e-02$	$7.72e+01$	$1.23e+01$	$7.69e-11$	$4.83e-11$
$F3$	$4.65e+01$	$2.12e+01$	$4.90e+01$	$1.97e+01$	$6.99e+01$	$3.58e+01$	$3.10e+01$	$8.65e+00$	$5.64e+07$	$1.42e+08$	$6.38e-01$	$1.49e+00$
$F4$	$4.78e-01$	$9.58e-01$	$1.99e-01$	$8.12e-01$	$3.21e+01$	$1.38e+01$	$4.79e-02$	$2.01e-01$	$1.12e+02$	$2.74e+01$	$3.72e+02$	$8.68e+01$
$F5$	$2.96e-04$	$1.48e-03$	$2.96e-04$	$1.48e-03$	$9.86e-04$	$2.76e-03$	$0.00e+00$	$0.00e+00$	$9.02e-01$	$1.82e+00$	$2.16e-01$	$5.64e-01$
$F6$	$1.49e-13$	$3.78e-14$	$2.58e-13$	$1.19e-13$	$7.16e-14$	$1.86e-14$	$1.39e-13$	$9.43e-15$	$3.23e+00$	$2.44e+00$	$1.90e+01$	$1.02e+00$
$F7$	$1.05e-15$	$3.21e-15$	$2.70e-11$	$1.33e-10$	$2.22e-15$	$1.17e-15$	$8.88e-17$	$1.96e-16$	$1.23e-09$	$1.45e-09$	$2.10e+01$	$1.38e+01$
$F8$	$1.64e+01$	$3.72e+01$	$1.55e+02$	$3.91e+02$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$9.02e+06$	$9.03e+10$	$9.39e+07$
$F9$	$5.35e-02$	$9.50e-02$	$4.40e-03$	$8.08e-03$	$2.85e+02$	$5.30e+00$	$2.73e+02$	$7.40e-01$	$3.11e+02$	$4.98e+00$	$3.16e+02$	$7.03e+00$
$F10$	$9.67e-32$	$4.73e-31$	$1.78e-22$	$6.97e-22$	$1.53e+00$	$1.29e+00$	$6.50e-29$	$3.60e-29$	$7.72e+00$	$2.93e+00$	$9.25e+00$	$2.82e+00$
$F11$	$2.81e-02$	$5.00e-02$	$2.65e-02$	$1.01e-01$	$9.65e-01$	$2.02e+00$	$6.26e-05$	$1.30e-05$	$1.01e-02$	$1.26e-02$	$1.95e+02$	$3.65e+01$
$F12$	$8.34e-05$	$4.00e-04$	$1.27e-05$	$4.66e-05$	$5.82e+00$	$1.03e+01$	$5.26e-13$	$1.64e-13$	$8.23e+01$	$1.53e+02$	$1.14e+02$	$1.01e+01$
$F13$	$3.06e+01$	$1.99e+01$	$2.75e+01$	$5.77e+00$	$5.97e+01$	$2.22e+01$	$2.48e+01$	$1.31e+00$	$1.43e+07$	$3.29e+07$	$1.16e+02$	$1.43e+01$
$F14$	$2.79e-01$	$6.75e-01$	$1.99e-01$	$8.12e-01$	$3.35e+01$	$1.86e+01$	$3.55e-08$	$2.26e-08$	$6.76e+01$	$1.30e+01$	$2.71e+02$	$7.30e+01$
$F15$	$9.68e-15$	$3.43e-14$	$1.08e-12$	$3.78e-12$	$2.29e-01$	$6.07e-01$	$1.99e-24$	$3.22e-24$	$3.07e+00$	$5.32e+00$	$3.94e+01$	$1.25e+02$
$F16$	$3.77e-03$	$1.13e-02$	$8.22e-05$	$4.09e-04$	$5.64e+00$	$8.47e+00$	$1.56e-09$	$2.81e-10$	$5.60e+01$	$5.16e+01$	$2.23e+02$	$1.50e+01$
$F17$	$4.86e+00$	$6.32e+00$	$3.33e+00$	$4.82e+00$	$1.51e+01$	$1.43e+01$	$8.52e-01$	$4.92e-01$	$7.61e+06$	$2.44e+07$	$3.47e+02$	$2.18e+01$
$F18$	$8.59e-02$	$2.76e-01$	$4.39e-02$	$1.99e-01$	$5.73e+00$	$5.26e+00$	$1.28e-04$	$4.63e-05$	$6.76e+01$	$3.46e+01$	$3.59e+02$	$8.45e+01$
$F19$	$1.33e-23$	$5.99e-23$	$1.64e-16$	$7.87e-16$	$1.23e+00$	$9.26e-01$	$2.00e-24$	$1.50e-24$	$1.95e+02$	$5.01e+02$	$1.71e+03$	$5.84e+03$
100-dimensional												
$F1$	$2.43e-13$	$5.57e-14$	$3.34e-13$	$9.61e-14$	$1.12e-16$	$4.28e-17$	$7.77e-17$	$1.13e-17$	$4.67e+02$	$7.02e+02$	$5.55e-17$	$1.26e-32$
$F2$	$2.12e+01$	$9.36e+00$	$2.27e+01$	$8.64e+00$	$7.74e+01$	$7.77e+00$	$4.60e+00$	$4.24e-01$	$9.96e+01$	$1.16e+01$	$2.61e-03$	$1.30e-02$
$F3$	$1.10e+02$	$2.66e+01$	$1.03e+02$	$3.06e+01$	$4.43e+02$	$3.63e+02$	$8.01e+01$	$1.03e+01$	$1.52e+08$	$2.69e+08$	$1.23e+01$	$1.80e+01$
$F4$	$1.07e+00$	$2.07e+00$	$3.58e-01$	$1.25e+00$	$1.01e+02$	$2.25e+01$	$9.53e-03$	$4.76e-02$	$2.92e+02$	$5.16e+01$	$8.38e+02$	$1.39e+02$
$F5$	$2.96e-04$	$1.48e-03$	$1.65e-13$	$3.58e-14$	$2.93e-02$	$5.32e-02$	$2.55e-17$	$5.19e-18$	$5.95e+00$	$1.29e+01$	$2.68e+00$	$1.05e+01$
$F6$	$4.14e-13$	$9.95e-14$	$4.41e-12$	$1.27e-11$	$1.55e+00$	$3.88e-01$	$3.10e-13$	$1.62e-14$	$4.79e+00$	$1.87e+00$	$1.86e+01$	$2.45e+00$
$F7$	$2.26e-15$	$1.02e-14$	$2.94e-05$	$1.47e-04$	$1.39e-14$	$7.12e-15$	$3.80e-17$	$5.29e-17$	$8.67e-02$	$3.70e-01$	$6.35e+01$	$2.36e+01$
$F8$	$8.60e+02$	$2.18e+03$	$2.24e+03$	$2.93e+03$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$1.92e+07$	$1.80e+11$	$3.54e+08$
$F9$	$4.80e-02$	$1.09e-01$	$2.04e-03$	$5.99e-03$	$5.43e+02$	$1.36e+01$	$5.06e+02$	$9.16e-01$	$5.87e+02$	$1.01e+01$	$6.08e+02$	$1.07e+01$
$F10$	$4.84e-28$	$2.40e-27$	$2.08e-23$	$6.17e-23$	$1.54e+01$	$3.31e+00$	$1.35e-28$	$3.86e-29$	$2.89e+01$	$1.01e+01$	$1.93e+01$	$5.10e+00$
$F11$	$7.29e-02$	$1.69e-01$	$4.44e-03$	$1.34e-02$	$4.31e+01$	$2.09e+01$	$1.25e-04$	$1.43e-05$	$2.80e+01$	$3.02e+01$	$4.82e+02$	$4.27e+01$
$F12$	$2.66e-03$	$9.35e-03$	$1.17e-03$	$3.72e-03$	$7.21e+01$	$3.21e+01$	$6.44e-11$	$1.52e-11$	$8.72e+02$	$2.55e+03$	$2.41e+02$	$1.23e+01$
$F13$	$6.74e+01$	$2.55e+01$	$6.30e+01$	$2.20e+01$	$2.76e+02$	$6.18e+01$	$6.13e+01$	$1.00e+00$	$9.37e+07$	$4.02e+08$	$2.59e+02$	$2.16e+01$
$F14$	$7.56e-01$	$2.12e+00$	$1.20e-01$	$3.30e-01$	$9.37e+01$	$1.56e+01$	$4.48e-02$	$2.24e-01$	$2.25e+02$	$4.59e+01$	$6.19e+02$	$9.25e+01$
$F15$	$6.36e-14$	$3.09e-13$	$7.16e-14$	$3.50e-13$	$3.67e+00$	$1.76e+00$	$7.10e-23$	$7.00e-23$	$5.99e+00$	$1.19e+01$	$5.57e+01$	$5.22e+01$
$F16$	$3.03e-03$	$9.53e-03$	$1.96e-03$	$9.43e-03$	$1.10e+02$	$3.80e+01$	$1.94e-02$	$9.70e-02$	$2.08e+02$	$1.49e+02$	$4.84e+02$	$2.08e+01$
$F17$	$1.92e+01$	$2.29e+01$	$1.30e+01$	$1.81e+01$	$1.78e+02$	$5.49e+01$	$1.19e+01$	$2.62e+00$	$4.36e+07$	$7.09e+07$	$7.04e+02$	$3.92e+01$
$F18$	$4.71e-01$	$1.01e+00$	$3.99e-02$	$1.99e-01$	$1.04e+02$	$4.39e+01$	$2.92e-04$	$6.77e-05$	$2.37e+02$	$7.02e+01$	$1.09e+03$	$4.15e+02$
$F19$	$5.54e-23$	$2.77e-22$	$5.86e-17$	$2.93e-16$	$1.17e+01$	$2.61e+00$	$4.79e-23$	$2.65e-23$	$4.70e+02$	$1.84e+03$	$5.83e+03$	$9.85e+03$

Table A.6: Average errors and standard deviations of the DEGPOA, eDEGPOA, and the base algorithms in the SOCO suite, for dimension $n = 200$ and 500.

Problem	DEGPOA		eDEGPOA		DE_{bin}		DE_{exp}		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
200-dimensional												
$F1$	$5.82e-13$	$1.41e-13$	$7.12e-13$	$1.20e-13$	$6.39e-16$	$5.32e-16$	$1.78e-16$	$1.60e-17$	$9.61e+02$	$1.65e+03$	$1.17e-16$	$1.13e-17$
$F2$	$4.65e+01$	$9.30e+00$	$5.24e+01$	$1.21e+01$	$1.01e+02$	$5.90e+00$	$1.89e+01$	$1.05e+00$	$1.17e+02$	$7.60e+00$	$7.47e-02$	$2.44e-01$
$F3$	$2.04e+02$	$3.48e+01$	$2.12e+02$	$3.38e+01$	$6.38e+02$	$3.80e+02$	$1.79e+02$	$8.89e+00$	$2.54e+08$	$3.97e+08$	$1.24e+02$	$8.85e+01$
$F4$	$1.43e+00$	$3.11e+00$	$3.98e-01$	$1.32e+00$	$4.21e+02$	$5.63e+01$	$8.52e-02$	$3.98e-01$	$6.32e+02$	$8.43e+01$	$1.57e+03$	$1.54e+02$
$F5$	$4.45e-12$	$2.07e-11$	$3.57e-13$	$6.72e-14$	$3.00e-01$	$7.73e-01$	$7.49e-17$	$6.94e-18$	$1.02e+01$	$1.59e+01$	$1.13e+00$	$2.84e+00$
$F6$	$9.15e-13$	$2.21e-13$	$1.85e-12$	$1.76e-12$	$5.28e+00$	$9.65e-01$	$6.46e-13$	$2.53e-14$	$8.14e+00$	$2.26e+00$	$1.93e+01$	$7.39e-01$
$F7$	$6.34e-15$	$1.15e-14$	$6.41e-12$	$3.16e-11$	$1.78e-11$	$4.65e-11$	$2.25e-16$	$1.92e-16$	$3.95e-01$	$1.21e+00$	$1.25e+02$	$1.92e+01$
$F8$	$5.92e+03$	$7.34e+03$	$1.56e+04$	$1.62e+04$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$3.09e+08$	$8.56e+11$	$3.36e+09$
$F9$	$3.79e-02$	$8.17e-02$	$5.54e-03$	$1.55e-02$	$1.13e+03$	$1.87e+01$	$1.01e+03$	$1.30e+00$	$1.18e+03$	$8.29e+00$	$1.22e+03$	$1.79e+01$
$F10$	$8.23e-28$	$4.11e-27$	$2.03e-21$	$1.02e-20$	$5.52e+01$	$1.02e+01$	$2.77e-28$	$5.31e-29$	$7.34e+01$	$6.25e+01$	$3.76e+01$	$2.78e+01$
$F11$	$5.20e-02$	$1.15e-01$	$3.68e-03$	$1.00e-02$	$3.95e+02$	$6.11e+01$	$2.55e-04$	$3.20e-05$	$4.03e+02$	$8.45e+01$	$1.08e+03$	$8.25e+01$
$F12$	$1.51e-03$	$4.22e-03$	$1.46e-03$	$7.27e-03$	$2.84e+02$	$4.97e+01$	$9.97e-10$	$2.01e-10$	$8.11e+02$	$1.58e+03$	$5.87e+02$	$4.52e+02$
$F13$	$1.47e+02$	$3.46e+01$	$1.37e+02$	$3.81e+01$	$7.52e+02$	$2.71e+02$	$1.40e+02$	$1.26e+01$	$2.06e+08$	$3.51e+08$	$5.92e+02$	$1.08e+02$
$F14$	$5.17e-01$	$1.41e+00$	$1.19e-01$	$5.97e-01$	$3.11e+02$	$3.66e+01$	$8.08e-03$	$4.04e-02$	$4.90e+02$	$5.23e+01$	$1.26e+03$	$1.81e+02$
$F15$	$7.95e-14$	$2.05e-13$	$1.29e-13$	$4.57e-13$	$1.17e+01$	$2.85e+00$	$3.71e-24$	$2.32e-24$	$1.40e+01$	$9.80e+00$	$1.95e+02$	$1.66e+02$
$F16$	$1.62e-02$	$4.13e-02$	$2.25e-05$	$6.82e-05$	$5.58e+02$	$7.96e+01$	$7.85e-09$	$1.11e-09$	$6.77e+02$	$6.04e+02$	$9.56e+02$	$3.33e+01$
$F17$	$5.99e+01$	$3.05e+01$	$4.71e+01$	$2.84e+01$	$1.03e+03$	$1.11e+02$	$3.71e+01$	$8.30e-01$	$1.17e+07$	$1.70e+07$	$1.49e+03$	$8.00e+01$
$F18$	$5.48e-02$	$2.06e-01$	$4.00e-02$	$1.99e-01$	$7.53e+02$	$6.55e+01$	$5.10e-04$	$9.97e-05$	$7.67e+02$	$2.14e+02$	$3.94e+03$	$3.91e+03$
$F19$	$2.43e-16$	$1.03e-15$	$1.72e-15$	$6.24e-15$	$4.04e+01$	$7.71e+00$	$1.67e-22$	$7.58e-23$	$7.51e+02$	$1.76e+03$	$2.53e+04$	$2.45e+04$
500-dimensional												
$F1$	$1.57e-12$	$2.76e-13$	$1.61e-12$	$4.01e-13$	$3.88e-05$	$7.93e-05$	$5.17e-16$	$1.36e-17$	$9.25e+02$	$1.27e+03$	n/a	n/a
$F2$	$8.93e+01$	$1.20e+01$	$8.76e+01$	$1.17e+01$	$1.25e+02$	$5.37e+00$	$5.38e+01$	$1.21e+00$	$1.35e+02$	$5.52e+00$	n/a	n/a
$F3$	$5.28e+02$	$8.52e+01$	$5.00e+02$	$2.83e+01$	$3.44e+04$	$1.62e+05$	$4.74e+02$	$1.48e+00$	$6.93e+08$	$1.72e+09$	n/a	n/a
$F4$	$1.87e+00$	$5.67e+00$	$3.98e-02$	$1.99e-01$	$2.35e+03$	$1.60e+02$	$7.12e-01$	$9.64e-01$	$2.11e+03$	$1.66e+02$	n/a	n/a
$F5$	$8.48e-13$	$1.51e-13$	$8.61e-13$	$1.54e-13$	$3.11e-01$	$5.07e-01$	$2.38e-16$	$1.18e-17$	$1.45e+01$	$2.52e+01$	n/a	n/a
$F6$	$3.28e-12$	$1.39e-12$	$2.91e-12$	$3.78e-13$	$1.49e+01$	$8.38e-01$	$1.64e-12$	$4.85e-14$	$1.27e+01$	$1.26e+00$	n/a	n/a
$F7$	$6.21e-13$	$2.14e-12$	$2.74e-13$	$1.36e-12$	$2.74e-03$	$6.49e-03$	$7.29e-16$	$3.58e-16$	$3.33e-05$	$1.12e-04$	n/a	n/a
$F8$	$9.60e+04$	$9.04e+04$	$1.11e+05$	$1.18e+05$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$1.42e+08$	n/a	n/a
$F9$	$5.47e-02$	$1.37e-01$	$3.58e-03$	$9.41e-03$	$2.97e+03$	$3.17e+01$	$2.52e+03$	$2.10e+00$	$3.00e+03$	$1.64e+01$	n/a	n/a
$F10$	$1.41e-31$	$4.85e-31$	$1.51e-30$	$3.54e-30$	$1.36e+02$	$2.08e+01$	$9.79e-28$	$1.43e-28$	$1.64e+02$	$5.62e+01$	n/a	n/a
$F11$	$1.31e-01$	$3.27e-01$	$3.88e-03$	$9.22e-03$	$2.34e+03$	$9.22e+01$	$6.78e-04$	$3.60e-05$	$1.67e+03$	$1.44e+02$	n/a	n/a
$F12$	$1.55e-02$	$4.42e-02$	$1.25e-05$	$4.65e-05$	$1.02e+03$	$6.68e+01$	$6.80e-09$	$8.58e-10$	$1.62e+03$	$1.83e+03$	n/a	n/a
$F13$	$3.91e+02$	$5.28e+01$	$3.54e+02$	$3.01e+01$	$2.49e+03$	$3.13e+02$	$3.60e+02$	$9.23e+00$	$3.41e+08$	$4.29e+08$	n/a	n/a
$F14$	$7.57e-01$	$2.73e+00$	$1.23e-04$	$6.13e-04$	$1.67e+03$	$1.51e+02$	$3.93e-01$	$1.05e+00$	$1.59e+03$	$1.57e+02$	n/a	n/a
$F15$	$9.20e-13$	$3.33e-12$	$8.21e-13$	$3.77e-12$	$4.44e+01$	$5.59e+00$	$2.93e-18$	$7.16e-18$	$3.50e+01$	$1.20e+01$	n/a	n/a
$F16$	$1.04e-02$	$4.78e-02$	$2.74e-06$	$7.42e-06$	$2.02e+03$	$8.60e+01$	$2.05e-08$	$1.64e-09$	$1.92e+03$	$1.44e+03$	n/a	n/a
$F17$	$1.41e+02$	$4.51e+01$	$1.39e+02$	$3.57e+01$	$3.83e+03$	$1.41e+02$	$1.12e+02$	$1.02e+00$	$6.64e+08$	$1.64e+09$	n/a	n/a
$F18$	$3.24e-01$	$8.12e-01$	$4.02e-02$	$1.99e-01$	$3.37e+03$	$4.34e+02$	$1.25e-03$	$1.87e-04$	$2.74e+03$	$3.59e+02$	n/a	n/a
$F19$	$7.64e-16$	$2.69e-15$	$2.29e-21$	$1.13e-20$	$1.29e+02$	$2.34e+01$	$3.35e-21$	$2.15e-21$	$2.05e+03$	$4.03e+03$	n/a	n/a

Table A.7: Average errors of the DEGPOA, eDEGPOA and other algorithms provided in the CEC-2013 for f_1-f_9 test problems.

		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
DEGPOA$_{0.5}$	30	$0.00e+00$	$8.47e+06$	$2.49e+08$	$2.56e+04$	$0.00e+00$	$2.57e+01$	$1.00e+02$	$2.10e+01$	$2.65e+01$
	50	$0.00e+00$	$1.89e+07$	$2.66e+09$	$5.58e+04$	$2.86e-07$	$4.66e+01$	$1.41e+02$	$2.12e+01$	$5.25e+01$
eDEGPOA$_{0.5}$	30	$0.00e+00$	$1.20e+07$	$5.97e+08$	$3.00e+04$	$0.00e+00$	$2.48e+01$	$9.41e+01$	$2.10e+01$	$2.62e+01$
	50	$0.00e+00$	$3.06e+07$	$3.86e+09$	$5.53e+04$	$4.22e-10$	$4.64e+01$	$1.41e+02$	$2.12e+01$	$5.29e+01$
SMADE	30	$0.00e+00$	$0.00e+00$	$9.82e+03$	$0.00e+00$	$0.00e+00$	$2.67e+00$	$3.25e+01$	$2.10e+01$	$2.23e+01$
	50	$0.00e+00$	$0.00e+00$	$3.81e+05$	$0.00e+00$	$0.00e+00$	$4.30e+01$	$4.32e+01$	$2.11e+01$	$4.36e+01$
TLBSaDE	30	$0.00e+00$	$6.73e+03$	$3.91e+01$	$2.34e+00$	$0.00e+00$	$1.04e-02$	$1.54e+01$	$2.08e+01$	$2.69e+01$
	50	$0.00e+00$	$1.68e+05$	$7.08e+05$	$6.58e+02$	$0.00e+00$	$4.07e+01$	$4.96e+01$	$2.11e+01$	$6.09e+01$
JANDE	30	$0.00e+00$	$1.29e+05$	$9.84e+06$	$1.97e+04$	$1.26e-08$	$7.93e+00$	$9.82e+00$	$2.09e+01$	$2.10e+01$
	50	$2.76e-08$	$6.05e+05$	$4.78e+07$	$8.34e+04$	$2.43e-06$	$4.30e+01$	$2.94e+01$	$2.11e+01$	$5.33e+01$
DE_APC	30	$0.00e+00$	$1.75e+05$	$3.21e+06$	$2.20e-01$	$0.00e+00$	$9.35e+00$	$2.18e+01$	$2.09e+01$	$3.07e+01$
	50	$0.00e+00$	$3.60e+05$	$6.98e+06$	$1.53e+00$	$0.00e+00$	$3.90e+01$	$3.66e+01$	$2.11e+01$	$6.09e+01$
TPC-GA	30	$0.00e+00$	$2.44e+05$	$3.80e+07$	$1.38e+01$	$0.00e+00$	$2.43e+01$	$2.91e+01$	$2.10e+01$	$3.61e+01$
	50	$0.00e+00$	$4.76e+05$	$1.06e+08$	$3.33e+00$	$0.00e+00$	$4.72e+01$	$4.17e+01$	$2.12e+01$	$7.30e+01$
PVADE	30	$0.00e+00$	$2.12e+06$	$1.65e+03$	$1.70e+04$	$1.40e-07$	$8.29e+00$	$1.29e+00$	$2.09e+01$	$6.30e+00$
	50	$0.00e+00$	$2.04e+05$	$7.48e+06$	$2.20e+02$	$1.39e-03$	$7.36e+01$	$2.07e+01$	$2.11e+01$	$2.60e+01$
CDASA	30	$0.00e+00$	$9.52e+05$	$4.54e+07$	$1.83e-01$	$8.19e-06$	$3.54e+01$	$6.95e+01$	$2.09e+01$	$2.35e+01$
	50	$0.00e+00$	$1.93e+06$	$2.18e+08$	$1.58e-02$	$8.40e-06$	$4.80e+01$	$1.04e+02$	$2.11e+01$	$4.67e+01$
PLES	30	$0.00e+00$	$1.34e+07$	$1.94e+09$	$4.47e+04$	$0.00e+00$	$7.77e+01$	$1.18e+02$	$2.09e+01$	$3.31e+01$
	50	$0.00e+00$	$1.59e+07$	$5.06e+09$	$5.43e+04$	$3.19e-10$	$9.70e+01$	$1.28e+02$	$2.11e+01$	$6.15e+01$

Table A.8: Average errors of the DEGPOA, eDEGPOA and other algorithms provided in the CEC-2013 for f_{10}-f_{19} test problems.

		f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}
$DEGPOA_{0.5}$	30	$1.79e+00$	$3.32e-01$	$8.90e+01$	$1.40e+02$	$1.80e+01$	$3.79e+03$	$1.33e+00$	$3.05e+01$	$1.41e+02$	$9.96e-01$
	50	$6.39e+00$	$2.93e-01$	$2.70e+02$	$3.66e+02$	$1.15e+01$	$7.73e+03$	$1.79e+00$	$5.08e+01$	$3.25e+02$	$1.82e+00$
$eDEGPOA_{0.5}$	30	$2.01e+00$	$7.80e-02$	$1.05e+02$	$1.57e+02$	$1.73e+01$	$3.79e+03$	$1.46e+00$	$3.05e+01$	$1.48e+02$	$9.63e-01$
	50	$7.08e+00$	$2.15e-01$	$3.10e+02$	$4.03e+02$	$2.22e+01$	$7.90e+03$	$1.82e+00$	$5.09e+01$	$3.33e+02$	$1.80e+00$
SMADE	30	$1.84e-02$	$1.09e+01$	$5.72e+01$	$1.28e+02$	$1.33e+02$	$4.10e+03$	$1.31e-01$	$3.48e+01$	$8.33e+01$	$2.55e+00$
	50	$2.47e-02$	$4.81e+01$	$1.57e+02$	$3.35e+02$	$3.41e+02$	$8.54e+03$	$8.96e-02$	$6.57e+01$	$1.93e+02$	$5.43e+00$
TLBSaDE	30	$1.62e-02$	$0.00e+00$	$4.99e+01$	$8.58e+01$	$3.07e+01$	$3.61e+03$	$1.48e+00$	$3.25e+01$	$7.68e+01$	$2.67e+00$
	50	$1.76e-02$	$0.00e+00$	$1.20e+02$	$2.19e+02$	$8.25e+02$	$7.69e+03$	$1.80e+00$	$7.95e+01$	$1.81e+02$	$7.57e+00$
JANDE	30	$7.91e-02$	$0.00e+00$	$4.28e+01$	$7.08e+01$	$1.33e+00$	$4.83e+03$	$2.28e+00$	$3.04e+01$	$1.23e+02$	$1.10e+00$
	50	$1.47e-01$	$1.95e-02$	$9.72e+01$	$1.76e+02$	$8.01e+00$	$9.48e+03$	$3.13e+00$	$5.08e+01$	$2.18e+02$	$2.24e+00$
DE_APC	30	$6.42e-02$	$3.08e+00$	$3.17e+01$	$7.55e+01$	$3.84e+03$	$4.14e+03$	$2.46e+00$	$5.92e+01$	$6.04e+01$	$2.30e+00$
	50	$6.71e-02$	$3.44e+01$	$5.96e+01$	$1.55e+02$	$9.96e+03$	$9.34e+03$	$3.24e+00$	$1.72e+02$	$1.05e+02$	$5.08e+00$
TPC-GA	30	$8.68e-02$	$2.39e+01$	$4.14e+01$	$8.41e+01$	$9.25e+02$	$3.97e+03$	$2.50e+00$	$5.44e+01$	$6.96e+01$	$3.28e+00$
	50	$1.05e-01$	$5.57e+01$	$9.62e+01$	$1.92e+02$	$2.55e+03$	$9.40e+03$	$3.38e+00$	$1.15e+02$	$1.68e+02$	$8.92e+00$
PVADE	30	$2.16e-02$	$5.84e+01$	$1.15e+02$	$1.31e+02$	$3.20e+03$	$5.61e+03$	$2.39e+00$	$1.02e+02$	$1.82e+02$	$5.40e+00$
	50	$5.99e-01$	$1.68e+02$	$2.57e+02$	$3.06e+02$	$7.34e+03$	$1.25e+04$	$3.39e+00$	$2.38e+02$	$3.87e+02$	$2.12e+01$
CDASA	30	$3.55e-02$	$1.17e+00$	$1.17e+02$	$1.86e+02$	$6.64e+02$	$3.87e+03$	$3.26e-01$	$3.40e+01$	$1.96e+02$	$2.10e+00$
	50	$4.66e-02$	$2.15e+00$	$2.67e+02$	$4.11e+02$	$1.08e+03$	$7.33e+03$	$4.97e-01$	$5.82e+01$	$4.43e+02$	$3.69e+00$
PLES	30	$1.18e+01$	$1.65e+02$	$2.15e+02$	$3.29e+02$	$2.61e+03$	$4.39e+03$	$1.32e+00$	$2.43e+02$	$2.57e+02$	$2.41e+01$
	50	$2.99e+01$	$3.53e+02$	$4.41e+02$	$6.38e+02$	$5.06e+03$	$8.51e+03$	$2.07e+00$	$6.01e+02$	$6.33e+02$	$1.28e+02$

Table A.9: Average errors of the DEGPOA, eDEGPOA and other algorithms provided in the CEC-2013 for f_{20}-f_{28} test problems .

		f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}	f_{28}
DEGPOA$_{0.5}$	30	$1.30e+01$	$2.87e+02$	$6.93e+01$	$4.34e+03$	$2.67e+02$	$2.67e+02$	$2.01e+02$	$9.46e+02$	$3.00e+02$
	50	$2.33e+01$	$5.02e+02$	$2.24e+01$	$9.07e+03$	$3.38e+02$	$3.36e+02$	$2.04e+02$	$1.57e+03$	$4.00e+02$
eDEGPOA$_{0.5}$	30	$1.30e+01$	$2.92e+02$	$1.01e+02$	$4.66e+03$	$2.69e+02$	$2.69e+02$	$2.01e+02$	$8.31e+02$	$3.00e+02$
	50	$2.34e+01$	$5.39e+02$	$4.99e+01$	$9.17e+03$	$3.39e+02$	$3.39e+02$	$2.04e+02$	$1.67e+03$	$4.00e+02$
SMADE	30	$1.05e+01$	$3.27e+02$	$1.79e+02$	$4.22e+03$	$2.32e+02$	$2.78e+02$	$2.15e+02$	$6.47e+02$	$3.88e+02$
	50	$1.92e+01$	$8.46e+02$	$3.39e+02$	$9.89e+03$	$3.00e+02$	$3.68e+02$	$2.91e+02$	$1.18e+03$	$1.07e+03$
TLBSaDE	30	$1.06e+01$	$2.67e+02$	$2.90e+02$	$4.34e+03$	$3.03e+02$	$2.96e+02$	$2.00e+02$	$1.19e+03$	$2.96e+02$
	50	$1.93e+01$	$3.12e+02$	$2.59e+03$	$9.68e+03$	$3.98e+02$	$3.79e+02$	$2.01e+02$	$2.17e+03$	$4.00e+02$
JANDE	30	$1.16e+01$	$2.94e+02$	$5.16e+01$	$4.61e+03$	$2.48e+02$	$2.60e+02$	$2.58e+02$	$7.22e+02$	$3.00e+02$
	50	$2.15e+01$	$8.24e+02$	$3.10e+01$	$9.48e+03$	$2.89e+02$	$3.17e+02$	$3.97e+02$	$1.16e+03$	$9.43e+02$
DE_APC	30	$1.26e+01$	$2.67e+02$	$4.56e+03$	$4.18e+03$	$2.92e+02$	$2.99e+02$	$3.28e+02$	$1.19e+03$	$3.00e+02$
	50	$2.23e+01$	$6.81e+02$	$1.06e+04$	$9.09e+03$	$3.84e+02$	$3.83e+02$	$4.09e+02$	$2.14e+03$	$6.97e+02$
TPC-GA	30	$1.37e+01$	$2.92e+02$	$1.27e+03$	$4.33e+03$	$2.74e+02$	$2.98e+02$	$3.25e+02$	$1.03e+03$	$3.00e+02$
	50	$2.34e+01$	$7.93e+02$	$3.51e+03$	$9.93e+03$	$3.77e+02$	$3.86e+02$	$4.22e+02$	$2.03e+03$	$4.59e+02$
PVADE	30	$1.13e+01$	$3.19e+02$	$2.50e+03$	$5.81e+03$	$2.02e+02$	$2.30e+02$	$2.18e+02$	$3.26e+02$	$3.00e+02$
	50	$2.07e+01$	$9.65e+02$	$7.72e+03$	$1.18e+04$	$2.78e+02$	$3.54e+02$	$3.47e+02$	$1.11e+03$	$4.62e+02$
CDASA	30	$1.48e+01$	$2.77e+02$	$4.89e+02$	$5.41e+03$	$2.98e+02$	$3.15e+02$	$2.91e+02$	$1.08e+03$	$3.87e+02$
	50	$2.43e+01$	$6.86e+02$	$7.32e+02$	$1.01e+04$	$3.74e+02$	$4.04e+02$	$3.44e+02$	$1.60e+03$	$1.04e+03$
PLES	30	$1.43e+01$	$3.30e+02$	$3.25e+03$	$5.00e+03$	$2.97e+02$	$3.27e+02$	$2.46e+02$	$1.15e+03$	$2.08e+03$
	50	$2.37e+01$	$7.58e+02$	$6.50e+03$	$1.02e+04$	$3.83e+02$	$4.44e+02$	$4.26e+02$	$2.03e+03$	$4.02e+03$

Table A.10: Average errors and standard deviations of the PSOPNA and the base algorithms in the SOCO test suite, for dimension $n = 50$ and 100.

Prob.	$\text{PSOPNA}_{0.5/1.5}$		DE_{bin}		DE_{exp}		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
50-dimensional										
$F1$	$2.32e-13$	$2.80e-13$	$3.00e-17$	$7.69e-18$	$2.78e-17$	$6.29e-33$	$2.90e+02$	$5.69e+02$	$2.78e-17$	$6.29e-33$
$F2$	$1.59e+01$	$9.16e+00$	$3.87e+01$	$8.90e+00$	$3.31e-01$	$5.90e-02$	$7.72e+01$	$1.23e+01$	$7.69e-11$	$4.83e-11$
$F3$	$4.09e+01$	$3.79e+01$	$6.99e+01$	$3.58e+01$	$3.10e+01$	$8.65e+00$	$5.64e+07$	$1.42e+08$	$6.38e-01$	$1.49e+00$
$F4$	$1.74e+02$	$4.29e+01$	$3.21e+01$	$1.38e+01$	$4.79e-02$	$2.01e-01$	$1.12e+02$	$2.74e+01$	$3.72e+02$	$8.68e+01$
$F5$	$6.39e-03$	$1.18e-02$	$9.86e-04$	$2.76e-03$	$0.00e+00$	$0.00e+00$	$9.02e-01$	$1.82e+00$	$2.16e-01$	$5.64e-01$
$F6$	$2.30e+00$	$1.54e+00$	$7.16e-14$	$1.86e-14$	$1.39e-13$	$9.43e-15$	$3.23e+00$	$2.44e+00$	$1.90e+01$	$1.02e+00$
$F7$	$2.80e-16$	$5.35e-16$	$2.22e-15$	$1.17e-15$	$8.88e-17$	$1.96e-16$	$1.23e-09$	$1.45e-09$	$2.10e+01$	$1.38e+01$
$F8$	$1.31e-01$	$2.64e-01$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$9.02e+06$	$9.03e+10$	$9.39e+07$
$F9$	$1.44e+02$	$4.12e+01$	$2.85e+02$	$5.30e+00$	$2.73e+02$	$7.40e-01$	$3.11e+02$	$4.98e+00$	$3.16e+02$	$7.03e+00$
$F10$	$6.63e+00$	$5.57e+00$	$1.53e+00$	$1.29e+00$	$6.50e-29$	$3.60e-29$	$7.72e+00$	$2.93e+00$	$9.25e+00$	$2.82e+00$
$F11$	$1.39e+02$	$3.81e+01$	$9.65e-01$	$2.02e+00$	$6.26e-05$	$1.30e-05$	$1.01e-02$	$1.26e-02$	$1.95e+02$	$3.65e+01$
$F12$	$3.19e+01$	$3.65e+01$	$5.82e+00$	$1.03e+01$	$5.26e-13$	$1.64e-13$	$8.23e+01$	$1.53e+02$	$1.14e+02$	$1.01e+01$
$F13$	$1.13e+02$	$4.70e+01$	$5.97e+01$	$2.22e+01$	$2.48e+01$	$1.31e+00$	$1.43e+07$	$3.29e+07$	$1.16e+02$	$1.43e+01$
$F14$	$1.25e+02$	$2.49e+01$	$3.35e+01$	$1.86e+01$	$3.55e-08$	$2.26e-08$	$6.76e+01$	$1.30e+01$	$2.71e+02$	$7.30e+01$
$F15$	$4.20e-02$	$2.10e-01$	$2.29e-01$	$6.07e-01$	$1.99e-24$	$3.22e-24$	$3.07e+00$	$5.32e+00$	$3.94e+01$	$1.25e+02$
$F16$	$9.35e+01$	$4.98e+01$	$5.64e+00$	$8.47e+00$	$1.56e-09$	$2.81e-10$	$5.60e+01$	$5.16e+01$	$2.23e+02$	$1.50e+01$
$F17$	$2.98e+02$	$3.84e+01$	$1.51e+01$	$1.43e+01$	$8.52e-01$	$4.92e-01$	$7.61e+06$	$2.44e+07$	$3.47e+02$	$2.18e+01$
$F18$	$7.28e+01$	$7.22e+00$	$5.73e+00$	$5.26e+00$	$1.28e-04$	$4.63e-05$	$6.76e+01$	$3.46e+01$	$3.59e+02$	$8.45e+01$
$F19$	$2.76e+00$	$3.04e+00$	$1.23e+00$	$9.26e-01$	$2.00e-24$	$1.50e-24$	$1.95e+02$	$5.01e+02$	$1.71e+03$	$5.84e+03$
100-dimensional										
$F1$	$2.76e-12$	$5.37e-12$	$1.12e-16$	$4.28e-17$	$7.77e-17$	$1.13e-17$	$4.67e+02$	$7.02e+02$	$5.55e-17$	$1.26e-32$
$F2$	$3.64e+01$	$1.08e+01$	$7.74e+01$	$7.77e+00$	$4.60e+00$	$4.24e-01$	$9.96e+01$	$1.16e+01$	$2.61e-03$	$1.30e-02$
$F3$	$1.32e+02$	$8.18e+01$	$4.43e+02$	$3.63e+02$	$8.01e+01$	$1.03e+01$	$1.52e+08$	$2.69e+08$	$1.23e+01$	$1.80e+01$
$F4$	$4.86e+02$	$8.67e+01$	$1.01e+02$	$2.25e+01$	$9.53e-03$	$4.76e-02$	$2.92e+02$	$5.16e+01$	$8.38e+02$	$1.39e+02$
$F5$	$3.15e-03$	$7.81e-03$	$2.93e-02$	$5.32e-02$	$2.55e-17$	$5.19e-18$	$5.95e+00$	$1.29e+01$	$2.68e+00$	$1.05e+01$
$F6$	$5.14e+00$	$2.28e+00$	$1.55e+00$	$3.88e-01$	$3.10e-13$	$1.62e-14$	$4.79e+00$	$1.87e+00$	$1.86e+01$	$2.45e+00$
$F7$	$7.69e-13$	$2.72e-12$	$1.39e-14$	$7.12e-15$	$3.80e-17$	$5.29e-17$	$8.67e-02$	$3.70e-01$	$6.35e+01$	$2.36e+01$
$F8$	$2.71e+02$	$3.44e+02$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$1.92e+07$	$1.80e+11$	$3.54e+08$
$F9$	$4.46e+02$	$5.74e+01$	$5.43e+02$	$1.36e+01$	$5.06e+02$	$9.16e-01$	$5.87e+02$	$1.01e+01$	$6.08e+02$	$1.07e+01$
$F10$	$2.50e+01$	$6.38e+00$	$1.54e+01$	$3.31e+00$	$1.35e-28$	$3.86e-29$	$2.89e+01$	$1.01e+01$	$1.93e+01$	$5.10e+00$
$F11$	$4.78e+02$	$6.32e+01$	$4.31e+01$	$2.09e+01$	$1.25e-04$	$1.43e-05$	$2.80e+01$	$3.02e+01$	$4.82e+02$	$4.27e+01$
$F12$	$1.43e+02$	$5.57e+01$	$7.21e+01$	$3.21e+01$	$6.44e-11$	$1.52e-11$	$8.72e+02$	$2.55e+03$	$2.41e+02$	$1.23e+01$
$F13$	$2.61e+02$	$5.82e+01$	$2.76e+02$	$6.18e+01$	$6.13e+01$	$1.00e+00$	$9.37e+07$	$4.02e+08$	$2.59e+02$	$2.16e+01$
$F14$	$3.31e+02$	$4.86e+01$	$9.37e+01$	$1.56e+01$	$4.48e-02$	$2.24e-01$	$2.25e+02$	$4.59e+01$	$6.19e+02$	$9.25e+01$
$F15$	$1.83e+00$	$2.30e+00$	$3.67e+00$	$1.76e+00$	$7.10e-23$	$7.00e-23$	$5.99e+00$	$1.19e+01$	$5.57e+01$	$5.22e+01$
$F16$	$3.39e+02$	$4.56e+01$	$1.10e+02$	$3.80e+01$	$1.94e-02$	$9.70e-02$	$2.08e+02$	$1.49e+02$	$4.84e+02$	$2.08e+01$
$F17$	$6.26e+02$	$6.07e+01$	$1.78e+02$	$5.49e+01$	$1.19e+01$	$2.62e+00$	$4.36e+07$	$7.09e+07$	$7.04e+02$	$3.92e+01$
$F18$	$1.81e+02$	$2.17e+01$	$1.04e+02$	$4.39e+01$	$2.92e-04$	$6.77e-05$	$2.37e+02$	$7.02e+01$	$1.09e+03$	$4.15e+02$
$F19$	$1.38e+01$	$6.03e+00$	$1.17e+01$	$2.61e+00$	$4.79e-23$	$2.65e-23$	$4.70e+02$	$1.84e+03$	$5.83e+03$	$9.85e+03$

Table A.11: Average errors and standard deviations of the PSOPNA and the base algorithms in the SOCO test suite, for dimension $n = 200$ and 500.

Prob.	$\text{PSOPNA}_{0.5/1.5}$		DE_{bin}		DE_{exp}		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
200-dimensional										
$F1$	$6.10e-09$	$2.93e-08$	$6.39e-16$	$5.32e-16$	$1.78e-16$	$1.60e-17$	$9.61e+02$	$1.65e+03$	$1.17e-16$	$1.13e-17$
$F2$	$6.10e+01$	$8.44e+00$	$1.01e+02$	$5.90e+00$	$1.89e+01$	$1.05e+00$	$1.17e+02$	$7.60e+00$	$7.47e-02$	$2.44e-01$
$F3$	$3.07e+02$	$1.14e+02$	$6.38e+02$	$3.80e+02$	$1.79e+02$	$8.89e+00$	$2.54e+08$	$3.97e+08$	$1.24e+02$	$8.85e+01$
$F4$	$1.22e+03$	$1.34e+02$	$4.21e+02$	$5.63e+01$	$8.52e-02$	$3.98e-01$	$6.32e+02$	$8.43e+01$	$1.57e+03$	$1.54e+02$
$F5$	$4.20e-02$	$1.06e-01$	$3.00e-01$	$7.73e-01$	$7.49e-17$	$6.94e-18$	$1.02e+01$	$1.59e+01$	$1.13e+00$	$2.84e+00$
$F6$	$1.11e+01$	$4.22e+00$	$5.28e+00$	$9.65e-01$	$6.46e-13$	$2.53e-14$	$8.14e+00$	$2.26e+00$	$1.93e+01$	$7.39e-01$
$F7$	$2.77e-08$	$1.18e-07$	$1.78e-11$	$4.65e-11$	$2.25e-16$	$1.92e-16$	$3.95e-01$	$1.21e+00$	$1.25e+02$	$1.92e+01$
$F8$	$1.01e+04$	$1.30e+04$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$3.09e+08$	$8.56e+11$	$3.36e+09$
$F9$	$1.18e+03$	$7.39e+01$	$1.13e+03$	$1.87e+01$	$1.01e+03$	$1.30e+00$	$1.18e+03$	$8.29e+00$	$1.22e+03$	$1.79e+01$
$F10$	$5.64e+01$	$9.29e+00$	$5.52e+01$	$1.02e+01$	$2.77e-28$	$5.31e-29$	$7.34e+01$	$6.25e+01$	$3.76e+01$	$2.78e+01$
$F11$	$1.17e+03$	$6.83e+01$	$3.95e+02$	$6.11e+01$	$2.55e-04$	$3.20e-05$	$4.03e+02$	$8.45e+01$	$1.08e+03$	$8.25e+01$
$F12$	$3.56e+02$	$5.83e+01$	$2.84e+02$	$4.97e+01$	$9.97e-10$	$2.01e-10$	$8.11e+02$	$1.58e+03$	$5.87e+02$	$4.52e+02$
$F13$	$6.25e+02$	$8.59e+01$	$7.52e+02$	$2.71e+02$	$1.40e+02$	$1.26e+01$	$2.06e+08$	$3.51e+08$	$5.92e+02$	$1.08e+02$
$F14$	$8.96e+02$	$1.17e+02$	$3.11e+02$	$3.66e+01$	$8.08e-03$	$4.04e-02$	$4.90e+02$	$5.23e+01$	$1.26e+03$	$1.81e+02$
$F15$	$6.30e+00$	$3.07e+00$	$1.17e+01$	$2.85e+00$	$3.71e-24$	$2.32e-24$	$1.40e+01$	$9.80e+00$	$1.95e+02$	$1.66e+02$
$F16$	$7.60e+02$	$6.07e+01$	$5.58e+02$	$7.96e+01$	$7.85e-09$	$1.11e-09$	$6.77e+02$	$6.04e+02$	$9.56e+02$	$3.33e+01$
$F17$	$1.29e+03$	$8.24e+01$	$1.03e+03$	$1.11e+02$	$3.71e+01$	$8.30e-01$	$1.17e+07$	$1.70e+07$	$1.49e+03$	$8.00e+01$
$F18$	$4.26e+02$	$4.56e+01$	$7.53e+02$	$6.55e+01$	$5.10e-04$	$9.97e-05$	$7.67e+02$	$2.14e+02$	$3.94e+03$	$3.91e+03$
$F19$	$4.16e+01$	$6.67e+00$	$4.04e+01$	$7.71e+00$	$1.67e-22$	$7.58e-23$	$7.51e+02$	$1.76e+03$	$2.53e+04$	$2.45e+04$
500-dimensional										
$F1$	$1.15e+02$	$3.81e+02$	$3.88e-05$	$7.93e-05$	$5.17e-16$	$1.36e-17$	$9.25e+02$	$1.27e+03$	n/a	n/a
$F2$	$9.65e+01$	$1.43e+01$	$1.25e+02$	$5.37e+00$	$5.38e+01$	$1.21e+00$	$1.35e+02$	$5.52e+00$	n/a	n/a
$F3$	$1.24e+07$	$5.97e+07$	$3.44e+04$	$1.62e+05$	$4.74e+02$	$1.48e+00$	$6.93e+08$	$1.72e+09$	n/a	n/a
$F4$	$3.81e+03$	$2.19e+02$	$2.35e+03$	$1.60e+02$	$7.12e-01$	$9.64e-01$	$2.11e+03$	$1.66e+02$	n/a	n/a
$F5$	$8.57e-01$	$1.82e+00$	$3.11e-01$	$5.07e-01$	$2.38e-16$	$1.18e-17$	$1.45e+01$	$2.52e+01$	n/a	n/a
$F6$	$1.91e+01$	$3.69e-01$	$1.49e+01$	$8.38e-01$	$1.64e-12$	$4.85e-14$	$1.27e+01$	$1.26e+00$	n/a	n/a
$F7$	$5.89e-02$	$1.96e-01$	$2.74e-03$	$6.49e-03$	$7.29e-16$	$3.58e-16$	$3.33e-05$	$1.12e-04$	n/a	n/a
$F8$	$1.66e+05$	$1.23e+05$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$1.42e+08$	n/a	n/a
$F9$	$3.29e+03$	$1.18e+02$	$2.97e+03$	$3.17e+01$	$2.52e+03$	$2.10e+00$	$3.00e+03$	$1.64e+01$	n/a	n/a
$F10$	$1.11e+02$	$2.90e+01$	$1.36e+02$	$2.08e+01$	$9.79e-28$	$1.43e-28$	$1.64e+02$	$5.62e+01$	n/a	n/a
$F11$	$3.35e+03$	$1.14e+02$	$2.34e+03$	$9.22e+01$	$6.78e-04$	$3.60e-05$	$1.67e+03$	$1.44e+02$	n/a	n/a
$F12$	$1.09e+03$	$2.33e+02$	$1.02e+03$	$6.68e+01$	$6.80e-09$	$8.58e-10$	$1.62e+03$	$1.83e+03$	n/a	n/a
$F13$	$3.39e+05$	$1.68e+06$	$2.49e+03$	$3.13e+02$	$3.60e+02$	$9.23e+00$	$3.41e+08$	$4.29e+08$	n/a	n/a
$F14$	$2.91e+03$	$1.84e+02$	$1.67e+03$	$1.51e+02$	$3.93e-01$	$1.05e+00$	$1.59e+03$	$1.57e+02$	n/a	n/a
$F15$	$2.47e+01$	$7.28e+00$	$4.44e+01$	$5.59e+00$	$2.93e-18$	$7.16e-18$	$3.50e+01$	$1.20e+01$	n/a	n/a
$F16$	$2.03e+03$	$2.84e+02$	$2.02e+03$	$8.60e+01$	$2.05e-08$	$1.64e-09$	$1.92e+03$	$1.44e+03$	n/a	n/a
$F17$	$3.53e+03$	$2.64e+02$	$3.83e+03$	$1.41e+02$	$1.12e+02$	$1.02e+00$	$6.64e+08$	$1.64e+09$	n/a	n/a
$F18$	$1.31e+03$	$1.38e+02$	$3.37e+03$	$4.34e+02$	$1.25e-03$	$1.87e-04$	$2.74e+03$	$3.59e+02$	n/a	n/a
$F19$	$9.13e+01$	$2.01e+01$	$1.29e+02$	$2.34e+01$	$3.35e-21$	$2.15e-21$	$2.05e+03$	$4.03e+03$	n/a	n/a

Appendix B

Average errors of GPALS

Average errors of GPALS for the SOCO and CEC-2013 test suites.

Table B.1: Average errors and standard deviations of the three GPALS-DE variants and the base algorithms in the SOCO test suite, for dimension $n = 50$.

Problem	GPALS-DE$_{0.5}$		GPALS-DE$_{0.2}$		GPALS-DE$_{0.8}$		DE$_{\text{bin}}$		DE$_{\text{exp}}$		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
f_1	$6.59e-14$	$2.13e-14$	$1.34e-13$	$3.23e-14$	$6.59e-14$	$2.13e-14$	$3.00e-17$	$7.69e-18$	$2.78e-17$	$6.29e-33$	$2.90e+02$	$5.69e+02$	$2.78e-17$	$6.29e-33$
f_2	$2.35e+00$	$2.49e+00$	$6.79e+00$	$2.78e+00$	$3.44e+00$	$2.55e+00$	$3.87e+01$	$8.90e+00$	$3.31e-01$	$5.90e-02$	$7.72e+01$	$1.23e+01$	$7.69e-11$	$4.83e-11$
f_3	$4.54e+01$	$1.18e+01$	$1.80e+01$	$2.49e+01$	$4.34e+01$	$1.89e+01$	$6.99e+01$	$3.58e+01$	$3.10e+01$	$8.65e+00$	$5.64e+07$	$1.42e+08$	$6.38e-01$	$1.49e+00$
f_4	$9.78e-14$	$3.08e-14$	$2.39e-01$	$5.94e-01$	$1.98e-13$	$3.07e-13$	$3.21e+01$	$1.38e+01$	$4.79e-02$	$2.01e-01$	$1.12e+02$	$2.74e+01$	$3.72e+02$	$8.68e+01$
f_5	$2.96e-14$	$5.68e-15$	$5.12e-14$	$2.59e-14$	$2.96e-14$	$5.68e-15$	$9.86e-04$	$2.76e-03$	$0.00e+00$	$0.00e+00$	$9.02e-01$	$1.82e+00$	$2.16e-01$	$5.64e-01$
f_6	$1.32e-13$	$4.70e-14$	$4.38e-09$	$2.19e-08$	$7.37e-13$	$1.76e-12$	$7.16e-14$	$1.86e-14$	$1.39e-13$	$9.43e-15$	$3.23e+00$	$2.44e+00$	$1.90e+01$	$1.02e+00$
f_7	$0.00e+00$	$0.00e+00$	$2.06e-15$	$9.49e-15$	$6.29e-14$	$2.76e-13$	$2.22e-15$	$1.17e-15$	$8.88e-17$	$1.96e-16$	$1.23e-09$	$1.45e-09$	$2.10e+01$	$1.38e+01$
f_8	$3.06e+02$	$6.25e+02$	$4.79e+02$	$4.43e+02$	$7.78e+01$	$1.10e+02$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$9.02e+06$	$9.03e+10$	$9.39e+07$
f_9	$1.94e-07$	$6.58e-07$	$9.12e-07$	$2.48e-06$	$5.83e-06$	$9.41e-06$	$2.85e+02$	$5.30e+00$	$2.73e+02$	$7.40e-01$	$3.11e+02$	$4.98e+00$	$3.16e+02$	$7.03e+00$
f_{10}	$1.01e-31$	$4.73e-31$	$2.01e-30$	$1.01e-29$	$9.83e-27$	$3.82e-26$	$1.53e+00$	$1.29e+00$	$6.50e-29$	$3.60e-29$	$7.72e+00$	$2.93e+00$	$9.25e+00$	$2.82e+00$
f_{11}	$1.10e-06$	$5.01e-06$	$7.38e-07$	$1.74e-06$	$1.51e-05$	$2.78e-05$	$9.65e-01$	$2.02e+00$	$6.26e-05$	$1.30e-05$	$1.01e-02$	$1.26e-02$	$1.95e+02$	$3.65e+01$
f_{12}	$8.23e-10$	$3.95e-09$	$1.30e-10$	$4.05e-10$	$2.27e-07$	$9.21e-07$	$5.82e+00$	$1.03e+01$	$5.26e-13$	$1.64e-13$	$8.23e+01$	$1.53e+02$	$1.14e+02$	$1.01e+01$
f_{13}	$3.32e+01$	$1.65e+01$	$4.46e+00$	$1.39e+01$	$4.04e+01$	$2.03e+01$	$5.97e+01$	$2.22e+01$	$2.48e+01$	$1.31e+00$	$1.43e+07$	$3.29e+07$	$1.16e+02$	$1.43e+01$
f_{14}	$7.19e-09$	$2.01e-08$	$1.76e-01$	$3.74e-01$	$2.32e-05$	$1.08e-04$	$3.35e+01$	$1.86e+01$	$3.55e-08$	$2.26e-08$	$6.76e+01$	$1.30e+01$	$2.71e+02$	$7.30e+01$
f_{15}	$1.17e-28$	$5.75e-28$	$6.37e-15$	$2.55e-14$	$7.47e-14$	$2.95e-13$	$2.29e-01$	$6.07e-01$	$1.99e-24$	$3.22e-24$	$3.07e+00$	$5.32e+00$	$3.94e+01$	$1.25e+02$
f_{16}	$1.80e-10$	$6.47e-10$	$1.54e-08$	$4.67e-08$	$4.12e-06$	$1.71e-05$	$5.64e+00$	$8.47e+00$	$1.56e-09$	$2.81e-10$	$5.60e+01$	$5.16e+01$	$2.23e+02$	$1.50e+01$
f_{17}	$1.83e+00$	$4.64e+00$	$9.21e+00$	$1.33e+01$	$8.36e+00$	$4.81e+00$	$1.51e+01$	$1.43e+01$	$8.52e-01$	$4.92e-01$	$7.61e+06$	$2.44e+07$	$3.47e+02$	$2.18e+01$
f_{18}	$9.71e-08$	$1.97e-07$	$1.20e-01$	$3.30e-01$	$1.15e-03$	$2.81e-03$	$5.73e+00$	$5.26e+00$	$1.28e-04$	$4.63e-05$	$6.76e+01$	$3.46e+01$	$3.59e+02$	$8.45e+01$
f_{19}	$1.07e-29$	$3.64e-29$	$1.38e-26$	$6.87e-26$	$5.09e-18$	$1.51e-17$	$1.23e+00$	$9.26e-01$	$2.00e-24$	$1.50e-24$	$1.95e+02$	$5.01e+02$	$1.71e+03$	$5.84e+03$

Table B.2: Average errors and standard deviations of the three GPALS-DE variants and the base algorithms in the SOCO test suite, for dimension $n = 100$.

Problem	GPALS-DE$_{0.5}$		GPALS-DE$_{0.2}$		GPALS-DE$_{0.8}$		DE$_{\text{bin}}$		DE$_{\text{exp}}$		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
f_1	$1.55e-13$	$3.86e-14$	$2.96e-13$	$6.14e-14$	$1.21e-13$	$2.99e-14$	$1.12e-16$	$4.28e-17$	$7.77e-17$	$1.13e-17$	$4.67e+02$	$7.02e+02$	$5.55e-17$	$1.26e-32$
f_2	$1.32e+01$	$7.03e+00$	$1.95e+01$	$3.63e+00$	$1.84e+01$	$8.97e+00$	$7.74e+01$	$7.77e+00$	$4.60e+00$	$4.24e-01$	$9.96e+01$	$1.16e+01$	$2.61e-03$	$1.30e-02$
f_3	$1.06e+02$	$2.51e+01$	$3.24e+01$	$3.93e+01$	$9.18e+01$	$1.91e+01$	$4.43e+02$	$3.63e+02$	$8.01e+01$	$1.03e+01$	$1.52e+08$	$2.69e+08$	$1.23e+01$	$1.80e+01$
f_4	$2.46e-13$	$5.37e-14$	$2.53e-01$	$6.61e-01$	$3.16e-13$	$1.20e-13$	$1.01e+02$	$2.25e+01$	$9.53e-03$	$4.76e-02$	$2.92e+02$	$5.16e+01$	$8.38e+02$	$1.39e+02$
f_5	$8.30e-14$	$2.71e-14$	$1.48e-13$	$4.42e-14$	$6.25e-14$	$2.17e-14$	$2.93e-02$	$5.32e-02$	$2.55e-17$	$5.19e-18$	$5.95e+00$	$1.29e+01$	$2.68e+00$	$1.05e+01$
f_6	$2.39e-13$	$8.99e-14$	$1.10e-10$	$5.43e-10$	$5.07e-13$	$1.11e-12$	$1.55e+00$	$3.88e-01$	$3.10e-13$	$1.62e-14$	$4.79e+00$	$1.87e+00$	$1.86e+01$	$2.45e+00$
f_7	$0.00e+00$	$0.00e+00$	$5.92e-14$	$2.86e-13$	$1.34e-15$	$6.72e-15$	$1.39e-14$	$7.12e-15$	$3.80e-17$	$5.29e-17$	$8.67e-02$	$3.70e-01$	$6.35e+01$	$2.36e+01$
f_8	$2.33e+03$	$2.44e+03$	$3.99e+03$	$2.92e+03$	$3.94e+03$	$3.51e+03$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$1.92e+07$	$1.80e+11$	$3.54e+08$
f_9	$9.76e-09$	$4.88e-08$	$4.31e-04$	$2.15e-03$	$1.67e-05$	$7.42e-05$	$5.43e+02$	$1.36e+01$	$5.06e+02$	$9.16e-01$	$5.87e+02$	$1.01e+01$	$6.08e+02$	$1.07e+01$
f_{10}	$1.60e-27$	$8.00e-27$	$2.36e-21$	$1.12e-20$	$2.19e-30$	$1.09e-29$	$1.54e+01$	$3.31e+00$	$1.35e-28$	$3.86e-29$	$2.89e+01$	$1.01e+01$	$1.93e+01$	$5.10e+00$
f_{11}	$2.05e-08$	$1.02e-07$	$1.60e-06$	$3.14e-06$	$2.84e-05$	$1.07e-04$	$4.31e+01$	$2.09e+01$	$1.25e-04$	$1.43e-05$	$2.80e+01$	$3.02e+01$	$4.82e+02$	$4.27e+01$
f_{12}	$6.31e-11$	$3.03e-10$	$1.52e-08$	$7.21e-08$	$6.74e-09$	$2.01e-08$	$7.21e+01$	$3.21e+01$	$6.44e-11$	$1.52e-11$	$8.72e+02$	$2.55e+03$	$2.41e+02$	$1.23e+01$
f_{13}	$6.54e+01$	$1.73e+01$	$7.91e+00$	$1.27e+01$	$7.35e+01$	$1.58e+01$	$2.76e+02$	$6.18e+01$	$6.13e+01$	$1.00e+00$	$9.37e+07$	$4.02e+08$	$2.59e+02$	$2.16e+01$
f_{14}	$5.15e-09$	$2.47e-08$	$1.59e-01$	$3.72e-01$	$1.14e-05$	$3.21e-05$	$9.37e+01$	$1.56e+01$	$4.48e-02$	$2.24e-01$	$2.25e+02$	$4.59e+01$	$6.19e+02$	$9.25e+01$
f_{15}	$7.26e-23$	$3.63e-22$	$2.98e-15$	$1.16e-14$	$3.11e-14$	$1.43e-13$	$3.67e+00$	$1.76e+00$	$7.10e-23$	$7.00e-23$	$5.99e+00$	$1.19e+01$	$5.57e+01$	$5.22e+01$
f_{16}	$4.48e-12$	$9.43e-12$	$4.60e-07$	$2.26e-06$	$6.60e-08$	$2.06e-07$	$1.10e+02$	$3.80e+01$	$1.94e-02$	$9.70e-02$	$2.08e+02$	$1.49e+02$	$4.84e+02$	$2.08e+01$
f_{17}	$1.52e+01$	$1.94e+01$	$1.28e+01$	$1.45e+01$	$3.82e+01$	$3.40e+01$	$1.78e+02$	$5.49e+01$	$1.19e+01$	$2.62e+00$	$4.36e+07$	$7.09e+07$	$7.04e+02$	$3.92e+01$
f_{18}	$9.68e-09$	$1.96e-08$	$4.28e-04$	$1.43e-03$	$1.24e-04$	$2.54e-04$	$1.04e+02$	$4.39e+01$	$2.92e-04$	$6.77e-05$	$2.37e+02$	$7.02e+01$	$1.09e+03$	$4.15e+02$
f_{19}	$1.40e-30$	$6.96e-30$	$8.44e-20$	$4.12e-19$	$6.86e-24$	$2.54e-23$	$1.17e+01$	$2.61e+00$	$4.79e-23$	$2.65e-23$	$4.70e+02$	$1.84e+03$	$5.83e+03$	$9.85e+03$

Table B.3: Average errors and standard deviations of the three GPALS-DE variants and the base algorithms in the SOCO test suite, for dimension $n = 200$.

Problem	GPALS-DE$_{0.5}$		GPALS-DE$_{0.2}$		GPALS-DE$_{0.8}$		DE$_{\text{bin}}$		DE$_{\text{exp}}$		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
f_1	$3.87e-13$	$1.10e-13$	$6.00e-13$	$1.98e-13$	$3.18e-13$	$8.84e-14$	$6.39e-16$	$5.32e-16$	$1.78e-16$	$1.60e-17$	$9.61e+02$	$1.65e+03$	$1.17e-16$	$1.13e-17$
f_2	$3.37e+01$	$8.45e+00$	$4.40e+01$	$6.26e+00$	$4.42e+01$	$1.19e+01$	$1.01e+02$	$5.90e+00$	$1.89e+01$	$1.05e+00$	$1.17e+02$	$7.60e+00$	$7.47e-02$	$2.44e-01$
f_3	$2.11e+02$	$3.13e+01$	$6.72e+01$	$7.09e+01$	$1.93e+02$	$2.66e+01$	$6.38e+02$	$3.80e+02$	$1.79e+02$	$8.89e+00$	$2.54e+08$	$3.97e+08$	$1.24e+02$	$8.85e+01$
f_4	$5.66e-13$	$1.76e-13$	$7.74e-01$	$8.91e-01$	$6.82e-13$	$1.93e-13$	$4.21e+02$	$5.63e+01$	$8.52e-02$	$3.98e-01$	$6.32e+02$	$8.43e+01$	$1.57e+03$	$1.54e+02$
f_5	$1.97e-13$	$4.99e-14$	$3.63e-13$	$1.23e-13$	$1.75e-13$	$6.75e-14$	$3.00e-01$	$7.73e-01$	$7.49e-17$	$6.94e-18$	$1.02e+01$	$1.59e+01$	$1.13e+00$	$2.84e+00$
f_6	$5.18e-13$	$9.65e-14$	$2.02e-03$	$1.01e-02$	$1.73e-12$	$6.46e-12$	$5.28e+00$	$9.65e-01$	$6.46e-13$	$2.53e-14$	$8.14e+00$	$2.26e+00$	$1.93e+01$	$7.39e-01$
f_7	$0.00e+00$	$0.00e+00$	$3.09e-16$	$1.13e-15$	$5.20e-15$	$2.60e-14$	$1.78e-11$	$4.65e-11$	$2.25e-16$	$1.92e-16$	$3.95e-01$	$1.21e+00$	$1.25e+02$	$1.92e+01$
f_8	$1.15e+04$	$1.19e+04$	$2.18e+04$	$1.33e+04$	$3.36e+04$	$1.72e+04$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$3.09e+08$	$8.56e+11$	$3.36e+09$
f_9	$2.04e-08$	$7.48e-08$	$9.41e-05$	$2.69e-04$	$2.79e-05$	$1.30e-04$	$1.13e+03$	$1.87e+01$	$1.01e+03$	$1.30e+00$	$1.18e+03$	$8.29e+00$	$1.22e+03$	$1.79e+01$
f_{10}	$2.13e-31$	$6.55e-31$	$1.53e-13$	$7.65e-13$	$2.67e-28$	$1.33e-27$	$5.52e+01$	$1.02e+01$	$2.77e-28$	$5.31e-29$	$7.34e+01$	$6.25e+01$	$3.76e+01$	$2.78e+01$
f_{11}	$1.40e-06$	$6.89e-06$	$1.08e-05$	$1.84e-05$	$2.21e-05$	$8.45e-05$	$3.95e+02$	$6.11e+01$	$2.55e-04$	$3.20e-05$	$4.03e+02$	$8.45e+01$	$1.08e+03$	$8.25e+01$
f_{12}	$2.83e-09$	$1.42e-08$	$4.66e-06$	$2.33e-05$	$3.22e-08$	$1.52e-07$	$2.84e+02$	$4.97e+01$	$9.97e-10$	$2.01e-10$	$8.11e+02$	$1.58e+03$	$5.87e+02$	$4.52e+02$
f_{13}	$1.41e+02$	$1.95e+01$	$4.51e+01$	$4.37e+01$	$1.46e+02$	$1.20e+01$	$7.52e+02$	$2.71e+02$	$1.40e+02$	$1.26e+01$	$2.06e+08$	$3.51e+08$	$5.92e+02$	$1.08e+02$
f_{14}	$1.53e-09$	$6.89e-09$	$2.14e-01$	$4.20e-01$	$1.54e-05$	$5.74e-05$	$3.11e+02$	$3.66e+01$	$8.08e-03$	$4.04e-02$	$4.90e+02$	$5.23e+01$	$1.26e+03$	$1.81e+02$
f_{15}	$2.12e-25$	$1.05e-24$	$9.41e-15$	$3.35e-14$	$4.90e-27$	$1.67e-26$	$1.17e+01$	$2.85e+00$	$3.71e-24$	$2.32e-24$	$1.40e+01$	$9.80e+00$	$1.95e+02$	$1.66e+02$
f_{16}	$2.97e-11$	$8.57e-11$	$2.50e-08$	$1.20e-07$	$1.19e-07$	$3.50e-07$	$5.58e+02$	$7.96e+01$	$7.85e-09$	$1.11e-09$	$6.77e+02$	$6.04e+02$	$9.56e+02$	$3.33e+01$
f_{17}	$4.97e+01$	$3.50e+01$	$8.93e+00$	$1.01e+01$	$5.68e+01$	$2.39e+01$	$1.03e+03$	$1.11e+02$	$3.71e+01$	$8.30e-01$	$1.17e+07$	$1.70e+07$	$1.49e+03$	$8.00e+01$
f_{18}	$1.21e-08$	$3.69e-08$	$8.93e-02$	$2.75e-01$	$2.20e-04$	$4.28e-04$	$7.53e+02$	$6.55e+01$	$5.10e-04$	$9.97e-05$	$7.67e+02$	$2.14e+02$	$3.94e+03$	$3.91e+03$
f_{19}	$0.00e+00$	$0.00e+00$	$5.05e-26$	$2.49e-25$	$6.41e-24$	$3.12e-23$	$4.04e+01$	$7.71e+00$	$1.67e-22$	$7.58e-23$	$7.51e+02$	$1.76e+03$	$2.53e+04$	$2.45e+04$

Table B.4: Average errors and standard deviations of the three GPALS-DE variants and the base algorithms in the SOCO test suite, for dimension $n = 500$.

Problem	GPALS-DE$_{0.5}$		GPALS-DE$_{0.2}$		GPALS-DE$_{0.8}$		DE$_{\text{bin}}$		DE$_{\text{exp}}$		CHC		GCMAES
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean StD
f_1	$1.04e-12$	$2.66e-13$	$1.86e-12$	$4.97e-13$	$9.75e-13$	$2.34e-13$	$3.88e-05$	$7.93e-05$	$5.17e-16$	$1.36e-17$	$9.25e+02$	$1.27e+03$	n/a n/a
f_2	$7.40e+01$	$8.28e+00$	$8.20e+01$	$9.53e+00$	$8.31e+01$	$1.04e+01$	$1.25e+02$	$5.37e+00$	$5.38e+01$	$1.21e+00$	$1.35e+02$	$5.52e+00$	n/a n/a
f_3	$4.99e+02$	$2.34e+01$	$1.92e+02$	$2.17e+02$	$4.88e+02$	$1.95e+01$	$3.44e+04$	$1.62e+05$	$4.74e+02$	$1.48e+00$	$6.93e+08$	$1.72e+09$	n/a n/a
f_4	$1.70e-12$	$4.47e-13$	$6.98e-01$	$8.85e-01$	$1.65e-12$	$3.37e-13$	$2.35e+03$	$1.60e+02$	$7.12e-01$	$9.64e-01$	$2.11e+03$	$1.66e+02$	n/a n/a
f_5	$5.35e-13$	$8.47e-14$	$1.01e-12$	$8.62e-13$	$5.89e-13$	$2.40e-13$	$3.11e-01$	$5.07e-01$	$2.38e-16$	$1.18e-17$	$1.45e+01$	$2.52e+01$	n/a n/a
f_6	$1.24e-12$	$1.93e-13$	$1.09e-02$	$3.51e-02$	$1.17e-12$	$3.40e-13$	$1.49e+01$	$8.38e-01$	$1.64e-12$	$4.85e-14$	$1.27e+01$	$1.26e+00$	n/a n/a
f_7	$0.00e+00$	$0.00e+00$	$6.31e-12$	$3.15e-11$	$0.00e+00$	$0.00e+00$	$2.74e-03$	$6.49e-03$	$7.29e-16$	$3.58e-16$	$3.33e-05$	$1.12e-04$	n/a n/a
f_8	$1.07e+05$	$7.42e+04$	$2.38e+05$	$1.36e+05$	$2.00e+05$	$7.54e+04$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$1.42e+08$	n/a n/a
f_9	$1.64e-08$	$4.60e-08$	$2.65e-04$	$4.69e-04$	$1.54e-04$	$7.10e-04$	$2.97e+03$	$3.17e+01$	$2.52e+03$	$2.10e+00$	$3.00e+03$	$1.64e+01$	n/a n/a
f_{10}	$1.42e-31$	$4.91e-31$	$4.77e-17$	$2.38e-16$	$4.01e-30$	$1.99e-29$	$1.36e+02$	$2.08e+01$	$9.79e-28$	$1.43e-28$	$1.64e+02$	$5.62e+01$	n/a n/a
f_{11}	$0.00e+00$	$0.00e+00$	$3.04e-03$	$1.26e-02$	$9.77e-07$	$3.47e-06$	$2.34e+03$	$9.22e+01$	$6.78e-04$	$3.60e-05$	$1.67e+03$	$1.44e+02$	n/a n/a
f_{12}	$9.80e-12$	$4.33e-11$	$1.76e-06$	$8.81e-06$	$4.32e-07$	$1.83e-06$	$1.02e+03$	$6.68e+01$	$6.80e-09$	$8.58e-10$	$1.62e+03$	$1.83e+03$	n/a n/a
f_{13}	$3.79e+02$	$3.79e+01$	$8.69e+01$	$1.00e+02$	$3.75e+02$	$2.01e+01$	$2.49e+03$	$3.13e+02$	$3.60e+02$	$9.23e+00$	$3.41e+08$	$4.29e+08$	n/a n/a
f_{14}	$9.10e-11$	$1.83e-10$	$8.93e-01$	$1.11e+00$	$1.29e-04$	$3.51e-04$	$1.67e+03$	$1.51e+02$	$3.93e-01$	$1.05e+00$	$1.59e+03$	$1.57e+02$	n/a n/a
f_{15}	$0.00e+00$	$0.00e+00$	$5.53e-16$	$2.58e-15$	$3.82e-25$	$1.89e-24$	$4.44e+01$	$5.59e+00$	$2.93e-18$	$7.16e-18$	$3.50e+01$	$1.20e+01$	n/a n/a
f_{16}	$1.46e-09$	$6.41e-09$	$6.06e-07$	$2.67e-06$	$2.17e-07$	$7.81e-07$	$2.02e+03$	$8.60e+01$	$2.05e-08$	$1.64e-09$	$1.92e+03$	$1.44e+03$	n/a n/a
f_{17}	$1.37e+02$	$2.99e+01$	$3.81e+00$	$4.97e+00$	$1.24e+02$	$1.31e+01$	$3.83e+03$	$1.41e+02$	$1.12e+02$	$1.02e+00$	$6.64e+08$	$1.64e+09$	n/a n/a
f_{18}	$4.89e-09$	$1.71e-08$	$2.92e-01$	$6.07e-01$	$7.81e-04$	$2.27e-03$	$3.37e+03$	$4.34e+02$	$1.25e-03$	$1.87e-04$	$2.74e+03$	$3.59e+02$	n/a n/a
f_{19}	$1.04e-12$	$2.66e-13$	$4.86e-08$	$2.43e-07$	$9.47e-22$	$4.74e-21$	$1.29e+02$	$2.34e+01$	$3.35e-21$	$2.15e-21$	$2.05e+03$	$4.03e+03$	n/a n/a

Table B.5: Average errors of the GPALS-DE, GPALS-PSO and other algorithms for the SOCO test problems $f_1 - f_9$.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
GPALS-DE$_{0.5}$									
50	$0.00e+00$	$2.35e+00$	$4.54e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.06e+02$	$1.94e-07$
100	$0.00e+00$	$1.32e+01$	$1.06e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$2.33e+03$	$0.00e+00$
200	$0.00e+00$	$3.37e+01$	$2.11e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.15e+04$	$2.04e-08$
500	$0.00e+00$	$7.40e+01$	$4.99e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.07e+05$	$1.64e-08$
GPALS-PSO$_{0.5}$									
50	$0.00e+00$	$1.66e+01$	$1.02e+02$	$2.17e+02$	$1.87e-03$	$5.52e-01$	$0.00e+00$	$1.87e+02$	$1.86e+02$
100	$1.51e-05$	$3.82e+01$	$2.80e+02$	$5.79e+02$	$4.90e-03$	$2.77e+00$	$0.00e+00$	$4.22e+03$	$5.41e+02$
200	$0.00e+00$	$5.81e+01$	$5.59e+02$	$1.40e+03$	$3.63e-02$	$1.31e+01$	$7.57e-07$	$3.41e+04$	$1.21e+03$
500	$9.91e+00$	$8.40e+01$	$1.57e+07$	$4.16e+03$	$4.44e-01$	$1.94e+01$	$1.50e-03$	$2.86e+05$	$3.36e+03$
SOUPDE									
50	$0.00e+00$	$1.18e+00$	$3.10e+01$	$3.98e-02$	$0.00e+00$	$1.47e-14$	$2.28e-14$	$9.69e-02$	$3.75e-06$
100	$0.00e+00$	$7.47e+00$	$7.92e+01$	$3.98e-02$	$0.00e+00$	$3.03e-14$	$3.88e-14$	$6.55e+01$	$7.82e-06$
200	$0.00e+00$	$2.38e+01$	$1.80e+02$	$1.19e-01$	$0.00e+00$	$6.40e-14$	$7.46e-14$	$2.46e+03$	$1.51e-05$
500	$0.00e+00$	$6.50e+01$	$4.71e+02$	$7.96e-02$	$0.00e+00$	$1.67e-13$	$1.78e-13$	$4.36e+04$	$3.59e-05$
DE-$D^{40}+M^m$									
50	$3.33e-18$	$1.67e-01$	$1.34e+01$	$1.99e-01$	$0.00e+00$	$4.55e-14$	$0.00e+00$	$6.11e-01$	$0.00e+00$
100	$2.78e-17$	$2.24e+00$	$7.61e+01$	$1.99e-01$	$1.39e-17$	$1.01e-13$	$5.33e-17$	$4.75e+05$	$4.29e-04$
200	$6.66e-17$	$9.58e+00$	$1.69e+02$	$2.39e-01$	$2.78e-17$	$2.51e-13$	$0.00e+00$	$2.19e+08$	$0.00e+00$
500	$2.23e-16$	$3.72e+01$	$4.54e+02$	$9.15e-01$	$1.03e-16$	$7.14e-13$	$0.00e+00$	$1.41e+10$	$6.78e-09$
GaDE									
50	$0.00e+00$	$1.46e+01$	$1.18e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.08e-08$	$6.24e-07$
100	$0.00e+00$	$3.88e+01$	$5.89e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.23e-03$	$3.87e-07$
200	$0.00e+00$	$5.76e+01$	$1.61e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.02e+00$	$4.53e-09$
500	$0.00e+00$	$7.42e+01$	$4.40e+02$	$0.00e+00$	$0.00e+00$	$1.46e-14$	$0.00e+00$	$1.33e+03$	$0.00e+00$
jDElscop									
50	$0.00e+00$	$3.15e-02$	$2.28e+01$	$0.00e+00$	$0.00e+00$	$9.55e-14$	$0.00e+00$	$9.97e-03$	$0.00e+00$
100	$0.00e+00$	$1.21e+00$	$6.13e+01$	$0.00e+00$	$0.00e+00$	$2.00e-13$	$0.00e+00$	$5.57e+00$	$7.18e-09$
200	$0.00e+00$	$7.54e+00$	$1.40e+02$	$0.00e+00$	$0.00e+00$	$4.52e-13$	$0.00e+00$	$2.52e+02$	$4.30e-08$
500	$0.00e+00$	$3.06e+01$	$4.06e+02$	$1.59e-01$	$0.00e+00$	$1.18e-12$	$0.00e+00$	$5.66e+03$	$6.10e-08$
SaDE-MMTS									
50	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$4.13e-09$	$1.35e-01$
100	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.05e-04$	$3.18e-01$
200	$0.00e+00$	$1.34e+00$	$0.00e+00$	$8.08e-02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$2.67e+01$	$1.24e+00$
500	$0.00e+00$	$1.25e+01$	$0.00e+00$	$3.85e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.01e+02$	$2.81e+01$
MOS									
50	$0.00e+00$	$4.64e-13$	$9.61e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.54e-08$	$0.00e+00$
100	$0.00e+00$	$2.94e-12$	$2.03e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.17e-02$	$0.00e+00$
200	$0.00e+00$	$1.24e-11$	$4.01e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.16e+02$	$0.00e+00$
500	$0.00e+00$	$5.51e-04$	$4.57e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.28e+04$	$0.00e+00$
MA-SSW-Chains									
50	$1.67e-17$	$7.61e-02$	$4.79e+01$	$1.19e-01$	$0.00e+00$	$4.89e-14$	$9.33e-17$	$3.06e-01$	$2.94e+02$
100	$2.78e-17$	$7.01e+00$	$1.38e+02$	$1.19e-01$	$1.39e-17$	$6.03e-14$	$8.17e-16$	$3.48e+01$	$5.63e+02$
200	$5.33e-17$	$3.36e+01$	$2.50e+02$	$4.43e+00$	$2.72e-17$	$1.19e-13$	$6.96e-15$	$7.23e+02$	$1.17e+03$
500	$1.01e-16$	$7.86e+01$	$6.07e+02$	$1.78e+02$	$7.70e-17$	$2.63e-13$	$4.69e-14$	$1.32e+04$	$2.53e+03$

Table B.6: Average errors of the GPALS-DE, GPALS-PSO and other algorithms for the SOCO test problems $f_{10} - f_{19}$.

	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}
GPALS-DE$_{0.5}$										
50	$0.00e+00$	$1.10e-06$	$0.00e+00$	$3.32e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.83e+00$	$9.71e-08$	$0.00e+00$
100	$0.00e+00$	$2.05e-08$	$0.00e+00$	$6.54e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.52e+01$	$0.00e+00$	$0.00e+00$
200	$0.00e+00$	$1.40e-06$	$0.00e+00$	$1.41e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$4.97e+01$	$1.21e-08$	$0.00e+00$
500	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.79e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.37e+02$	$0.00e+00$	$0.00e+00$
GPALS-PSO$_{0.5}$										
50	$2.79e+00$	$1.79e+02$	$8.88e+00$	$1.33e+02$	$1.41e+02$	$1.34e-01$	$8.37e+01$	$2.77e+02$	$7.14e+01$	$7.75e-01$
100	$2.30e+01$	$5.14e+02$	$9.92e+01$	$2.99e+02$	$4.26e+02$	$1.18e+00$	$3.04e+02$	$6.42e+02$	$1.85e+02$	$9.66e+00$
200	$6.05e+01$	$1.25e+03$	$2.94e+02$	$6.63e+02$	$1.08e+03$	$7.01e+00$	$6.40e+02$	$1.30e+03$	$4.80e+02$	$4.22e+01$
500	$1.38e+02$	$3.38e+03$	$7.72e+02$	$2.50e+06$	$3.12e+03$	$2.11e+01$	$1.61e+03$	$3.28e+03$	$1.37e+03$	$9.51e+01$
SOUPDE										
50	$0.00e+00$	$3.09e-06$	$0.00e+00$	$2.06e+01$	$0.00e+00$	$1.38e-14$	$0.00e+00$	$2.53e-01$	$0.00e+00$	$0.00e+00$
100	$0.00e+00$	$6.75e-06$	$0.00e+00$	$5.85e+01$	$9.09e-15$	$2.79e-14$	$0.00e+00$	$8.55e+00$	$0.00e+00$	$0.00e+00$
200	$0.00e+00$	$1.43e-05$	$0.00e+00$	$1.35e+02$	$3.98e-02$	$5.79e-14$	$0.00e+00$	$3.31e+01$	$0.00e+00$	$1.91e-14$
500	$0.00e+00$	$4.66e-04$	$0.00e+00$	$3.58e+02$	$1.31e-12$	$1.39e-13$	$0.00e+00$	$1.09e+02$	$2.82e-13$	$4.95e-14$
DE-$D^{40}+M^n$										
50	$1.89e-31$	$6.06e-04$	$1.58e-21$	$1.39e+01$	$1.19e-01$	$0.00e+00$	$1.76e-16$	$4.31e-02$	$3.98e-02$	$0.00e+00$
100	$0.00e+00$	$0.00e+00$	$4.62e-17$	$5.33e+01$	$1.19e-01$	$0.00e+00$	$8.99e-15$	$3.22e+00$	$8.11e-10$	$0.00e+00$
200	$3.51e+01$	$0.00e+00$	$5.45e-15$	$1.23e+02$	$3.98e-02$	$0.00e+00$	$2.26e-13$	$2.72e+01$	$3.98e-02$	$9.47e-32$
500	$2.43e-31$	$0.00e+00$	$5.05e-13$	$3.48e+02$	$2.39e-01$	$0.00e+00$	$4.17e-12$	$1.02e+02$	$9.97e-08$	$0.00e+00$
GaDE										
50	$0.00e+00$	$1.31e-06$	$0.00e+00$	$1.19e+01$	$9.78e-13$	$0.00e+00$	$4.78e-12$	$4.97e-01$	$4.82e-08$	$0.00e+00$
100	$0.00e+00$	$4.34e-07$	$0.00e+00$	$4.99e+01$	$7.90e-13$	$0.00e+00$	$2.45e-12$	$3.28e+00$	$1.96e-08$	$0.00e+00$
200	$4.20e-02$	$1.85e-07$	$4.92e-14$	$1.24e+02$	$2.87e-12$	$0.00e+00$	$1.58e-12$	$2.45e+01$	$2.53e-08$	$0.00e+00$
500	$3.78e-01$	$0.00e+00$	$1.07e-12$	$3.34e+02$	$2.79e-11$	$0.00e+00$	$1.67e-12$	$9.26e+01$	$5.59e-08$	$4.20e-02$
jDElscop										
50	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.36e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$7.43e-03$	$2.41e-14$	$0.00e+00$
100	$0.00e+00$	$8.17e-09$	$0.00e+00$	$5.11e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.21e-01$	$6.33e-14$	$0.00e+00$
200	$0.00e+00$	$9.58e-09$	$0.00e+00$	$1.10e+02$	$4.11e-16$	$0.00e+00$	$0.00e+00$	$2.39e+01$	$2.04e-13$	$0.00e+00$
500	$0.00e+00$	$4.40e-08$	$0.00e+00$	$3.14e+02$	$8.00e-02$	$0.00e+00$	$0.00e+00$	$7.65e+01$	$1.11e-12$	$0.00e+00$
SaDE-MMTS										
50	$0.00e+00$	$5.19e-05$	$0.00e+00$	$4.23e+00$	$3.93e-08$	$0.00e+00$	$0.00e+00$	$4.78e-01$	$9.38e-03$	$0.00e+00$
100	$0.00e+00$	$2.00e-04$	$0.00e+00$	$3.30e+01$	$1.02e-02$	$0.00e+00$	$0.00e+00$	$1.17e+01$	$4.70e-02$	$0.00e+00$
200	$0.00e+00$	$2.39e-04$	$0.00e+00$	$8.89e+01$	$1.57e-02$	$0.00e+00$	$0.00e+00$	$3.50e+01$	$3.35e-01$	$0.00e+00$
500	$0.00e+00$	$2.53e+01$	$0.00e+00$	$3.27e+02$	$4.01e-01$	$0.00e+00$	$0.00e+00$	$9.80e+01$	$1.18e+00$	$0.00e+00$
MOS										
50	$0.00e+00$	$0.00e+00$	$0.00e+00$	$4.55e-01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.40e+01$	$0.00e+00$	$0.00e+00$
100	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.75e+01$	$1.68e-11$	$0.00e+00$	$0.00e+00$	$1.43e+01$	$0.00e+00$	$0.00e+00$
200	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.03e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$5.03e+00$	$0.00e+00$	$0.00e+00$
500	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.78e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.21e+01$	$0.00e+00$	$0.00e+00$
MA-SSW-Chains										
50	$1.67e-30$	$4.49e-03$	$6.27e-41$	$3.02e+01$	$1.37e-17$	$3.91e-16$	$4.06e-03$	$2.60e+01$	$3.88e-19$	$4.02e-31$
100	$1.05e-29$	$1.09e-01$	$3.28e-03$	$8.35e+01$	$2.21e-16$	$1.59e-15$	$1.61e-02$	$9.92e+01$	$2.71e-18$	$3.15e-30$
200	$5.41e-29$	$3.50e-01$	$1.75e-02$	$1.68e+02$	$9.76e-01$	$5.32e-15$	$6.02e-02$	$7.55e+01$	$4.29e-04$	$1.51e-16$
500	$2.80e-01$	$4.21e+01$	$2.55e+01$	$4.00e+02$	$5.65e+01$	$5.53e+00$	$1.08e-01$	$1.38e+02$	$2.41e-03$	$7.84e-17$

Table B.7: Average error of the SHADE and L-SHADE variants for the SOCO test problems $f_1 - f_9$.

		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
SHADE_{60}	50	$0.00e+00$	$1.22e+01$	$7.81e+02$	$3.98e-02$	$4.13e-03$	$2.46e-01$	$0.00e+00$	$0.00e+00$	$9.00e-01$
	100	$0.00e+00$	$5.86e+01$	$8.27e+02$	$0.00e+00$	$1.70e-02$	$2.49e+00$	$0.00e+00$	$9.06e-05$	$2.81e+01$
	200	$0.00e+00$	$8.33e+01$	$1.07e+03$	$1.43e+00$	$5.44e-02$	$5.51e+00$	$0.00e+00$	$2.41e+00$	$4.68e+01$
	500	$2.01e-09$	$1.05e+02$	$1.79e+03$	$1.06e+02$	$4.90e-01$	$1.37e+01$	$6.05e-05$	$2.38e+03$	$3.73e+01$
SHADE_{100}	50	$0.00e+00$	$2.24e-01$	$7.81e+02$	$0.00e+00$	$6.90e-04$	$0.00e+00$	$0.00e+00$	$4.22e-09$	$9.85e-02$
	100	$0.00e+00$	$4.41e+01$	$8.36e+02$	$0.00e+00$	$3.83e-03$	$1.45e+00$	$0.00e+00$	$5.40e-04$	$1.42e+01$
	200	$0.00e+00$	$7.13e+01$	$1.07e+03$	$3.98e-02$	$2.47e-02$	$3.43e+00$	$0.00e+00$	$1.85e+00$	$1.43e+02$
	500	$0.00e+00$	$9.55e+01$	$1.79e+03$	$1.48e+01$	$2.56e-01$	$1.09e+01$	$1.10e-05$	$1.66e+03$	$2.56e+02$
mSHADE_{60}	50	$0.00e+00$	$2.15e+01$	$7.98e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.02e-07$	$4.30e-03$
	100	$0.00e+00$	$6.82e+01$	$8.48e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$6.56e-03$	$1.10e-02$
	200	$0.00e+00$	$9.51e+01$	$9.90e+02$	$0.00e+00$	$6.90e-04$	$0.00e+00$	$0.00e+00$	$1.06e+01$	$3.08e-02$
	500	$4.93e-06$	$1.17e+02$	$1.97e+03$	$5.66e-09$	$1.09e-01$	$9.75e-01$	$1.26e-03$	$3.81e+03$	$7.49e-02$
mSHADE_{100}	50	$0.00e+00$	$5.60e+00$	$8.13e+02$	$0.00e+00$	$0.00e+00$	$2.63e-06$	$5.75e-07$	$6.45e-03$	$6.98e-01$
	100	$0.00e+00$	$4.37e+01$	$8.57e+02$	$0.00e+00$	$0.00e+00$	$3.23e-06$	$1.55e-06$	$1.90e-01$	$1.70e+00$
	200	$0.00e+00$	$8.06e+01$	$9.71e+02$	$0.00e+00$	$0.00e+00$	$3.05e-06$	$3.55e-06$	$3.04e+01$	$4.07e+00$
	500	$0.00e+00$	$1.07e+02$	$1.76e+03$	$0.00e+00$	$5.39e-03$	$5.93e-07$	$3.22e-04$	$3.64e+03$	$1.22e+01$
L-SHADE	50	$0.00e+00$	$5.32e-07$	$8.03e+02$	$1.03e-01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$5.41e-10$	$1.39e-01$
	100	$0.00e+00$	$2.40e-03$	$8.62e+02$	$2.81e+01$	$2.96e-04$	$0.00e+00$	$0.00e+00$	$5.43e-02$	$4.09e+00$
	200	$0.00e+00$	$3.64e-01$	$9.71e+02$	$2.26e+02$	$1.09e-03$	$0.00e+00$	$1.59e-06$	$9.52e+01$	$5.55e+01$
	500	-	-	-	-	-	-	-	-	-
mL-SHADE	50	$6.45e-05$	$5.91e+00$	$8.27e+02$	$1.75e-02$	$6.43e-05$	$2.34e-03$	$1.41e-03$	$4.59e+00$	$4.16e+00$
	100	$3.89e-01$	$2.55e+01$	$1.17e+03$	$1.52e+01$	$2.36e-01$	$1.87e-01$	$3.43e-01$	$1.01e+03$	$6.02e+01$
	200	$1.20e+02$	$5.24e+01$	$7.93e+04$	$2.91e+02$	$1.83e+00$	$1.87e+00$	$9.68e+00$	$1.47e+04$	$4.04e+02$
	500	-	-	-	-	-	-	-	-	-

Table B.8: Average errors of the SHADE and L-SHADE variants for the SOCO test problems $f_{10} - f_{19}$.

	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}
SHADE_{60}										
50	$3.47e+00$	$4.52e-01$	$4.05e+00$	$4.20e+01$	$0.00e+00$	$1.00e-01$	$7.18e-01$	$4.49e+00$	$5.79e-09$	$1.45e+00$
100	$3.10e+01$	$1.87e+01$	$7.11e+01$	$1.72e+02$	$0.00e+00$	$2.88e+00$	$5.65e+01$	$1.52e+02$	$2.11e-06$	$1.88e+01$
200	$8.45e+01$	$4.46e+01$	$2.85e+02$	$6.35e+02$	$5.33e-01$	$1.26e+01$	$5.49e+02$	$1.05e+03$	$4.61e-02$	$5.88e+01$
500	$2.26e+02$	$4.51e+01$	$1.00e+03$	$2.65e+03$	$4.84e+01$	$5.68e+01$	$1.84e+03$	$3.50e+03$	$8.13e+00$	$1.76e+02$
SHADE_{100}										
50	$2.81e-01$	$5.99e-02$	$0.00e+00$	$1.31e+01$	$2.93e-06$	$0.00e+00$	$8.58e-04$	$1.19e+00$	$2.61e-01$	$0.00e+00$
100	$2.45e+01$	$1.62e+01$	$3.37e+01$	$1.16e+02$	$0.00e+00$	$8.81e-01$	$3.38e+01$	$5.35e+01$	$1.07e+00$	$1.05e+01$
200	$8.14e+01$	$1.24e+02$	$2.28e+02$	$4.83e+02$	$4.29e-04$	$1.05e+01$	$3.89e+02$	$8.32e+02$	$5.43e+00$	$5.75e+01$
500	$2.47e+02$	$2.54e+02$	$7.90e+02$	$1.96e+03$	$8.24e+00$	$5.10e+01$	$1.61e+03$	$3.03e+03$	$3.36e+01$	$1.91e+02$
mSHADE_{60}										
50	$0.00e+00$	$4.56e-03$	$0.00e+00$	$1.41e+01$	$0.00e+00$	$0.00e+00$	$3.09e-06$	$4.82e-01$	$9.20e-05$	$0.00e+00$
100	$0.00e+00$	$1.26e-02$	$0.00e+00$	$6.59e+01$	$0.00e+00$	$0.00e+00$	$5.93e-06$	$7.60e+00$	$1.96e-04$	$0.00e+00$
200	$0.00e+00$	$2.96e-02$	$6.16e-01$	$1.98e+02$	$4.79e-09$	$1.41e+00$	$4.50e-06$	$8.98e+01$	$4.66e-04$	$3.36e+00$
500	$3.27e+01$	$7.59e-02$	$3.07e+02$	$1.69e+03$	$8.51e-03$	$3.02e+01$	$9.76e+01$	$7.09e+02$	$1.19e-02$	$9.95e+01$
mSHADE_{100}										
50	$0.00e+00$	$7.06e-01$	$8.99e-03$	$3.10e+01$	$8.83e-03$	$2.98e-07$	$6.48e-02$	$9.18e+00$	$1.19e-01$	$1.82e-09$
100	$0.00e+00$	$1.70e+00$	$2.75e-02$	$6.31e+01$	$2.17e-02$	$8.88e-07$	$1.35e-01$	$2.92e+01$	$2.59e-01$	$3.15e-08$
200	$0.00e+00$	$4.12e+00$	$7.41e-02$	$1.73e+02$	$4.20e-02$	$2.12e-06$	$2.83e-01$	$6.58e+01$	$5.48e-01$	$2.10e-01$
500	$7.99e+00$	$1.23e+01$	$2.04e+01$	$1.07e+03$	$5.78e-02$	$2.08e+01$	$3.31e+00$	$3.81e+02$	$1.41e+00$	$1.91e+01$
L-SHADE										
50	$1.65e-02$	$1.20e-01$	$4.29e-04$	$5.61e+01$	$2.68e+00$	$0.00e+00$	$2.54e-01$	$1.04e+01$	$4.49e+00$	$0.00e+00$
100	$8.21e-01$	$7.03e+00$	$2.34e+01$	$1.52e+02$	$3.49e+01$	$7.46e-09$	$1.41e+01$	$4.80e+01$	$3.97e+01$	$2.92e-01$
200	$1.47e+01$	$5.74e+01$	$2.05e+02$	$3.32e+02$	$2.01e+02$	$8.27e-02$	$1.84e+02$	$4.66e+02$	$1.36e+02$	$4.74e+00$
500	-	-	-	-	-	-	-	-	-	-
mL-SHADE										
50	$2.17e-04$	$4.30e+00$	$2.18e-01$	$3.61e+01$	$3.27e-01$	$1.54e-03$	$9.16e-01$	$1.46e+01$	$9.88e-01$	$3.40e-04$
100	$1.27e+00$	$5.86e+01$	$8.04e+00$	$9.37e+01$	$1.83e+01$	$6.79e-01$	$2.61e+01$	$8.54e+01$	$1.74e+01$	$1.03e+00$
200	$3.17e+01$	$3.98e+02$	$8.83e+01$	$6.68e+03$	$2.34e+02$	$1.45e+01$	$2.06e+02$	$3.17e+02$	$1.40e+02$	$2.80e+01$
500	-	-	-	-	-	-	-	-	-	-

Table B.9: Average errors and standard deviations of the GPALS-PSO and the base algorithms in the SOCO test suite, for dimension $n = 50$.

Problem	GPALS-PSO$_{0.5}$		GPALS-PSO$_{0.2}$		GPALS-PSO$_{0.8}$		DE$_{\text{bin}}$		DE$_{\text{exp}}$		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
$F1$	$1.03e-12$	$2.46e-12$	$7.11e-07$	$3.56e-06$	$3.66e+03$	$7.48e+03$	$3.00e-17$	$7.69e-18$	$2.78e-17$	$6.29e-33$	$2.90e+02$	$5.69e+02$	$2.78e-17$	$6.29e-33$
f_2	$1.66e+01$	$1.45e+01$	$4.93e+01$	$2.38e+01$	$1.53e+01$	$9.07e+00$	$3.87e+01$	$8.90e+00$	$3.31e-01$	$5.90e-02$	$7.72e+01$	$1.23e+01$	$7.69e-11$	$4.83e-11$
f_3	$1.02e+02$	$8.10e+01$	$1.89e+02$	$1.31e+02$	$8.64e+07$	$3.10e+08$	$6.99e+01$	$3.58e+01$	$3.10e+01$	$8.65e+00$	$5.64e+07$	$1.42e+08$	$6.38e-01$	$1.49e+00$
f_4	$2.17e+02$	$5.77e+01$	$3.77e+02$	$3.87e+01$	$1.53e+02$	$1.05e+02$	$3.21e+01$	$1.38e+01$	$4.79e-02$	$2.01e-01$	$1.12e+02$	$2.74e+01$	$3.72e+02$	$8.68e+01$
f_5	$1.87e-03$	$4.04e-03$	$1.00e+00$	$4.72e+00$	$1.19e+01$	$3.34e+01$	$9.86e-04$	$2.76e-03$	$0.00e+00$	$0.00e+00$	$9.02e-01$	$1.82e+00$	$2.16e-01$	$5.64e-01$
f_6	$5.52e-01$	$8.22e-01$	$1.63e+01$	$5.09e+00$	$2.60e+00$	$3.58e+00$	$7.16e-14$	$1.86e-14$	$1.39e-13$	$9.43e-15$	$3.23e+00$	$2.44e+00$	$1.90e+01$	$1.02e+00$
f_7	$3.24e-16$	$9.56e-16$	$2.36e-05$	$8.04e-05$	$1.80e+01$	$3.69e+01$	$2.22e-15$	$1.17e-15$	$8.88e-17$	$1.96e-16$	$1.23e-09$	$1.45e-09$	$2.10e+01$	$1.38e+01$
f_8	$1.87e+02$	$1.23e+02$	$4.86e+02$	$5.07e+02$	$5.31e+03$	$5.85e+03$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$9.02e+06$	$9.03e+10$	$9.39e+07$
f_9	$1.86e+02$	$2.89e+01$	$2.63e+02$	$2.19e+01$	$1.66e+02$	$5.45e+01$	$2.85e+02$	$5.30e+00$	$2.73e+02$	$7.40e-01$	$3.11e+02$	$4.98e+00$	$3.16e+02$	$7.03e+00$
f_{10}	$2.79e+00$	$2.64e+00$	$1.43e+01$	$5.73e+00$	$3.36e-01$	$7.25e-01$	$1.53e+00$	$1.29e+00$	$6.50e-29$	$3.60e-29$	$7.72e+00$	$2.93e+00$	$9.25e+00$	$2.82e+00$
f_{11}	$1.79e+02$	$3.51e+01$	$2.51e+02$	$3.06e+01$	$1.84e+02$	$6.24e+01$	$9.65e-01$	$2.02e+00$	$6.26e-05$	$1.30e-05$	$1.01e-02$	$1.26e-02$	$1.95e+02$	$3.65e+01$
f_{12}	$8.88e+00$	$1.84e+01$	$6.33e+01$	$1.85e+01$	$1.51e+03$	$2.93e+03$	$5.82e+00$	$1.03e+01$	$5.26e-13$	$1.64e-13$	$8.23e+01$	$1.53e+02$	$1.14e+02$	$1.01e+01$
f_{13}	$1.33e+02$	$4.07e+01$	$1.60e+02$	$4.84e+01$	$2.03e+07$	$7.39e+07$	$5.97e+01$	$2.22e+01$	$2.48e+01$	$1.31e+00$	$1.43e+07$	$3.29e+07$	$1.16e+02$	$1.43e+01$
f_{14}	$1.41e+02$	$3.36e+01$	$2.44e+02$	$4.01e+01$	$1.11e+02$	$5.97e+01$	$3.35e+01$	$1.86e+01$	$3.55e-08$	$2.26e-08$	$6.76e+01$	$1.30e+01$	$2.71e+02$	$7.30e+01$
f_{15}	$1.34e-01$	$3.62e-01$	$4.85e+00$	$3.04e+00$	$9.06e+00$	$2.85e+01$	$2.29e-01$	$6.07e-01$	$1.99e-24$	$3.22e-24$	$3.07e+00$	$5.32e+00$	$3.94e+01$	$1.25e+02$
f_{16}	$8.37e+01$	$4.15e+01$	$1.47e+02$	$2.65e+01$	$1.76e+03$	$2.70e+03$	$5.64e+00$	$8.47e+00$	$1.56e-09$	$2.81e-10$	$5.60e+01$	$5.16e+01$	$2.23e+02$	$1.50e+01$
f_{17}	$2.77e+02$	$3.74e+01$	$4.16e+02$	$1.86e+02$	$1.46e+04$	$5.05e+04$	$1.51e+01$	$1.43e+01$	$8.52e-01$	$4.92e-01$	$7.61e+06$	$2.44e+07$	$3.47e+02$	$2.18e+01$
f_{18}	$7.14e+01$	$8.98e+00$	$9.17e+01$	$1.05e+01$	$6.91e+01$	$2.34e+01$	$5.73e+00$	$5.26e+00$	$1.28e-04$	$4.63e-05$	$6.76e+01$	$3.46e+01$	$3.59e+02$	$8.45e+01$
f_{19}	$7.75e-01$	$1.28e+00$	$1.11e+01$	$4.43e+00$	$4.95e+01$	$1.57e+02$	$1.23e+00$	$9.26e-01$	$2.00e-24$	$1.50e-24$	$1.95e+02$	$5.01e+02$	$1.71e+03$	$5.84e+03$

Table B.10: Average errors and standard deviations of the GPALS-PSO and the base algorithms in the SOCO test suite, for dimension $n = 100$.

Problem	GPALS-PSO$_{0.5}$		GPALS-PSO$_{0.2}$		GPALS-PSO$_{0.8}$		DE$_{\text{bin}}$		DE$_{\text{exp}}$		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
f_1	$1.51e-05$	$7.57e-05$	$2.99e-05$	$1.49e-04$	$7.26e+03$	$1.41e+04$	$1.12e-16$	$4.28e-17$	$7.77e-17$	$1.13e-17$	$4.67e+02$	$7.02e+02$	$5.55e-17$	$1.26e-32$
f_2	$3.82e+01$	$2.12e+01$	$8.39e+01$	$2.66e+01$	$4.12e+01$	$1.39e+01$	$7.74e+01$	$7.77e+00$	$4.60e+00$	$4.24e-01$	$9.96e+01$	$1.16e+01$	$2.61e-03$	$1.30e-02$
f_3	$2.80e+02$	$1.38e+02$	$7.66e+02$	$1.32e+03$	$4.17e+08$	$1.79e+09$	$4.43e+02$	$3.63e+02$	$8.01e+01$	$1.03e+01$	$1.52e+08$	$2.69e+08$	$1.23e+01$	$1.80e+01$
f_4	$5.79e+02$	$8.88e+01$	$8.14e+02$	$9.58e+01$	$3.71e+02$	$1.82e+02$	$1.01e+02$	$2.25e+01$	$9.53e-03$	$4.76e-02$	$2.92e+02$	$5.16e+01$	$8.38e+02$	$1.39e+02$
f_5	$4.90e-03$	$1.28e-02$	$8.04e+00$	$3.77e+01$	$3.64e+01$	$6.97e+01$	$2.93e-02$	$5.32e-02$	$2.55e-17$	$5.19e-18$	$5.95e+00$	$1.29e+01$	$2.68e+00$	$1.05e+01$
f_6	$2.77e+00$	$2.37e+00$	$1.88e+01$	$5.02e-01$	$5.20e+00$	$5.23e+00$	$1.55e+00$	$3.88e-01$	$3.10e-13$	$1.62e-14$	$4.79e+00$	$1.87e+00$	$1.86e+01$	$2.45e+00$
f_7	$2.50e-13$	$8.83e-13$	$1.06e+02$	$1.20e+02$	$5.15e+01$	$7.42e+01$	$1.39e-14$	$7.12e-15$	$3.80e-17$	$5.29e-17$	$8.67e-02$	$3.70e-01$	$6.35e+01$	$2.36e+01$
f_8	$4.22e+03$	$1.87e+03$	$4.68e+03$	$2.94e+03$	$2.75e+04$	$1.85e+04$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$1.92e+07$	$1.80e+11$	$3.54e+08$
f_9	$5.41e+02$	$4.43e+01$	$5.89e+02$	$4.30e+01$	$5.08e+02$	$8.02e+01$	$5.43e+02$	$1.36e+01$	$5.06e+02$	$9.16e-01$	$5.87e+02$	$1.01e+01$	$6.08e+02$	$1.07e+01$
f_{10}	$2.30e+01$	$7.14e+00$	$1.79e+01$	$1.29e+01$	$1.78e+02$	$8.12e+02$	$1.54e+01$	$3.31e+00$	$1.35e-28$	$3.86e-29$	$2.89e+01$	$1.01e+01$	$1.93e+01$	$5.10e+00$
f_{11}	$5.14e+02$	$3.96e+01$	$5.88e+02$	$3.27e+01$	$5.09e+02$	$6.20e+01$	$4.31e+01$	$2.09e+01$	$1.25e-04$	$1.43e-05$	$2.80e+01$	$3.02e+01$	$4.82e+02$	$4.27e+01$
f_{12}	$9.92e+01$	$3.57e+01$	$6.71e+02$	$2.21e+03$	$6.70e+03$	$1.02e+04$	$7.21e+01$	$3.21e+01$	$6.44e-11$	$1.52e-11$	$8.72e+02$	$2.55e+03$	$2.41e+02$	$1.23e+01$
f_{13}	$2.99e+02$	$7.68e+01$	$3.75e+02$	$8.89e+01$	$1.80e+08$	$7.54e+08$	$2.76e+02$	$6.18e+01$	$6.13e+01$	$1.00e+00$	$9.37e+07$	$4.02e+08$	$2.59e+02$	$2.16e+01$
f_{14}	$4.26e+02$	$7.16e+01$	$6.05e+02$	$9.65e+01$	$2.62e+02$	$1.14e+02$	$9.37e+01$	$1.56e+01$	$4.48e-02$	$2.24e-01$	$2.25e+02$	$4.59e+01$	$6.19e+02$	$9.25e+01$
f_{15}	$1.18e+00$	$1.60e+00$	$1.15e+02$	$3.65e+02$	$2.40e+02$	$5.34e+02$	$3.67e+00$	$1.76e+00$	$7.10e-23$	$7.00e-23$	$5.99e+00$	$1.19e+01$	$5.57e+01$	$5.22e+01$
f_{16}	$3.04e+02$	$3.94e+01$	$4.02e+02$	$5.43e+02$	$6.75e+03$	$6.98e+03$	$1.10e+02$	$3.80e+01$	$1.94e-02$	$9.70e-02$	$2.08e+02$	$1.49e+02$	$4.84e+02$	$2.08e+01$
f_{17}	$6.42e+02$	$7.15e+01$	$6.99e+02$	$2.25e+02$	$1.04e+04$	$4.69e+04$	$1.78e+02$	$5.49e+01$	$1.19e+01$	$2.62e+00$	$4.36e+07$	$7.09e+07$	$7.04e+02$	$3.92e+01$
f_{18}	$1.85e+02$	$2.23e+01$	$2.38e+02$	$2.89e+01$	$1.56e+02$	$3.62e+01$	$1.04e+02$	$4.39e+01$	$2.92e-04$	$6.77e-05$	$2.37e+02$	$7.02e+01$	$1.09e+03$	$4.15e+02$
f_{19}	$9.66e+00$	$6.01e+00$	$1.60e+01$	$8.21e+00$	$1.30e+02$	$4.68e+02$	$1.17e+01$	$2.61e+00$	$4.79e-23$	$2.65e-23$	$4.70e+02$	$1.84e+03$	$5.83e+03$	$9.85e+03$

Table B.11: Average errors and standard deviations of the GPALS-PSO and the base algorithms in the SOCO test suite, for dimension $n = 200$.

Problem	GPALS-PSO$_{0.5}$		GPALS-PSO$_{0.2}$		GPALS-PSO$_{0.8}$		DE$_{\text{bin}}$		DE$_{\text{exp}}$		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
f_1	$3.75e-11$	$9.67e-11$	$2.12e+04$	$1.01e+05$	$2.12e+04$	$2.62e+04$	$6.39e-16$	$5.32e-16$	$1.78e-16$	$1.60e-17$	$9.61e+02$	$1.65e+03$	$1.17e-16$	$1.13e-17$
f_2	$5.81e+01$	$1.54e+01$	$9.69e+01$	$1.95e+01$	$7.10e+01$	$2.16e+01$	$1.01e+02$	$5.90e+00$	$1.89e+01$	$1.05e+00$	$1.17e+02$	$7.60e+00$	$7.47e-02$	$2.44e-01$
f_3	$5.59e+02$	$5.50e+02$	$1.06e+10$	$3.67e+10$	$7.47e+08$	$2.83e+09$	$6.38e+02$	$3.80e+02$	$1.79e+02$	$8.89e+00$	$2.54e+08$	$3.97e+08$	$1.24e+02$	$8.85e+01$
f_4	$1.40e+03$	$1.33e+02$	$1.95e+03$	$5.41e+02$	$1.23e+03$	$6.48e+02$	$4.21e+02$	$5.63e+01$	$8.52e-02$	$3.98e-01$	$6.32e+02$	$8.43e+01$	$1.57e+03$	$1.54e+02$
f_5	$3.63e-02$	$9.39e-02$	$1.54e+00$	$4.23e+00$	$2.97e+02$	$4.73e+02$	$3.00e-01$	$7.73e-01$	$7.49e-17$	$6.94e-18$	$1.02e+01$	$1.59e+01$	$1.13e+00$	$2.84e+00$
f_6	$1.31e+01$	$7.55e+00$	$1.95e+01$	$5.74e-01$	$9.48e+00$	$6.24e+00$	$5.28e+00$	$9.65e-01$	$6.46e-13$	$2.53e-14$	$8.14e+00$	$2.26e+00$	$1.93e+01$	$7.39e-01$
f_7	$7.57e-07$	$3.73e-06$	$4.08e+02$	$1.73e+02$	$2.12e+28$	$1.06e+29$	$1.78e-11$	$4.65e-11$	$2.25e-16$	$1.92e-16$	$3.95e-01$	$1.21e+00$	$1.25e+02$	$1.92e+01$
f_8	$3.41e+04$	$9.69e+03$	$2.79e+04$	$2.24e+04$	$1.54e+05$	$6.83e+04$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$3.09e+08$	$8.56e+11$	$3.36e+09$
f_9	$1.21e+03$	$4.44e+01$	$1.27e+03$	$1.29e+02$	$1.37e+03$	$2.45e+02$	$1.13e+03$	$1.87e+01$	$1.01e+03$	$1.30e+00$	$1.18e+03$	$8.29e+00$	$1.22e+03$	$1.79e+01$
f_{10}	$6.05e+01$	$9.70e+00$	$2.56e+02$	$1.07e+03$	$6.17e+01$	$6.93e+01$	$5.52e+01$	$1.02e+01$	$2.77e-28$	$5.31e-29$	$7.34e+01$	$6.25e+01$	$3.76e+01$	$2.78e+01$
f_{11}	$1.25e+03$	$7.25e+01$	$1.26e+03$	$1.01e+02$	$1.26e+03$	$1.74e+02$	$3.95e+02$	$6.11e+01$	$2.55e-04$	$3.20e-05$	$4.03e+02$	$8.45e+01$	$1.08e+03$	$8.25e+01$
f_{12}	$2.94e+02$	$4.74e+01$	$1.46e+04$	$6.96e+04$	$2.23e+04$	$1.90e+04$	$2.84e+02$	$4.97e+01$	$9.97e-10$	$2.01e-10$	$8.11e+02$	$1.58e+03$	$5.87e+02$	$4.52e+02$
f_{13}	$6.63e+02$	$1.20e+02$	$1.01e+09$	$5.07e+09$	$2.58e+09$	$6.45e+09$	$7.52e+02$	$2.71e+02$	$1.40e+02$	$1.26e+01$	$2.06e+08$	$3.51e+08$	$5.92e+02$	$1.08e+02$
f_{14}	$1.08e+03$	$1.29e+02$	$1.33e+03$	$2.10e+02$	$1.03e+03$	$5.28e+02$	$3.11e+02$	$3.66e+01$	$8.08e-03$	$4.04e-02$	$4.90e+02$	$5.23e+01$	$1.26e+03$	$1.81e+02$
f_{15}	$7.01e+00$	$3.83e+00$	$1.58e+03$	$1.07e+03$	$1.33e+16$	$6.63e+16$	$1.17e+01$	$2.85e+00$	$3.71e-24$	$2.32e-24$	$1.40e+01$	$9.80e+00$	$1.95e+02$	$1.66e+02$
f_{16}	$6.40e+02$	$4.28e+01$	$9.66e+03$	$4.01e+04$	$1.48e+04$	$1.75e+04$	$5.58e+02$	$7.96e+01$	$7.85e-09$	$1.11e-09$	$6.77e+02$	$6.04e+02$	$9.56e+02$	$3.33e+01$
f_{17}	$1.30e+03$	$1.25e+02$	$1.11e+09$	$3.47e+09$	$5.28e+07$	$1.81e+08$	$1.03e+03$	$1.11e+02$	$3.71e+01$	$8.30e-01$	$1.17e+07$	$1.70e+07$	$1.49e+03$	$8.00e+01$
f_{18}	$4.80e+02$	$4.57e+01$	$6.15e+02$	$1.03e+02$	$4.17e+02$	$1.08e+02$	$7.53e+02$	$6.55e+01$	$5.10e-04$	$9.97e-05$	$7.67e+02$	$2.14e+02$	$3.94e+03$	$3.91e+03$
f_{19}	$4.22e+01$	$9.23e+00$	$3.46e+01$	$4.54e+01$	$9.73e+02$	$2.48e+03$	$4.04e+01$	$7.71e+00$	$1.67e-22$	$7.58e-23$	$7.51e+02$	$1.76e+03$	$2.53e+04$	$2.45e+04$

Table B.12: Average errors and standard deviations of the GPALS-PSO and the base algorithms in the SOCO test suite, for dimension $n = 500$.

Problem	GPALS-PSO$_{0.5}$		GPALS-PSO$_{0.2}$		GPALS-PSO$_{0.8}$		DE$_{\text{bin}}$		DE$_{\text{exp}}$		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
f_1	$9.91e+00$	$4.83e+01$	$7.22e+01$	$3.61e+02$	$7.22e+01$	$3.61e+02$	$3.88e-05$	$7.93e-05$	$5.17e-16$	$1.36e-17$	$9.25e+02$	$1.27e+03$	n/a	n/a
f_2	$8.40e+01$	$1.09e+01$	$9.17e+01$	$1.53e+01$	$9.17e+01$	$1.53e+01$	$1.25e+02$	$5.37e+00$	$5.38e+01$	$1.21e+00$	$1.35e+02$	$5.52e+00$	n/a	n/a
f_3	$1.57e+07$	$7.85e+07$	$2.64e+06$	$1.20e+07$	$2.64e+06$	$1.20e+07$	$3.44e+04$	$1.62e+05$	$4.74e+02$	$1.48e+00$	$6.93e+08$	$1.72e+09$	n/a	n/a
f_4	$4.16e+03$	$2.36e+02$	$4.04e+03$	$3.74e+02$	$4.04e+03$	$3.74e+02$	$2.35e+03$	$1.60e+02$	$7.12e-01$	$9.64e-01$	$2.11e+03$	$1.66e+02$	n/a	n/a
f_5	$4.44e-01$	$5.61e-01$	$4.06e-01$	$6.32e-01$	$4.06e-01$	$6.32e-01$	$3.11e-01$	$5.07e-01$	$2.38e-16$	$1.18e-17$	$1.45e+01$	$2.52e+01$	n/a	n/a
f_6	$1.94e+01$	$1.34e-01$	$1.94e+01$	$1.12e-01$	$1.94e+01$	$1.12e-01$	$1.49e+01$	$8.38e-01$	$1.64e-12$	$4.85e-14$	$1.27e+01$	$1.26e+00$	n/a	n/a
f_7	$1.50e-03$	$5.90e-03$	$6.28e-02$	$3.11e-01$	$6.28e-02$	$3.11e-01$	$2.74e-03$	$6.49e-03$	$7.29e-16$	$3.58e-16$	$3.33e-05$	$1.12e-04$	n/a	n/a
f_8	$2.86e+05$	$1.24e+05$	$2.50e+05$	$9.06e+04$	$2.50e+05$	$9.06e+04$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$1.42e+08$	n/a	n/a
f_9	$3.36e+03$	$1.17e+02$	$3.35e+03$	$1.34e+02$	$3.35e+03$	$1.34e+02$	$2.97e+03$	$3.17e+01$	$2.52e+03$	$2.10e+00$	$3.00e+03$	$1.64e+01$	n/a	n/a
f_{10}	$1.38e+02$	$3.92e+01$	$1.41e+02$	$2.68e+01$	$1.41e+02$	$2.68e+01$	$1.36e+02$	$2.08e+01$	$9.79e-28$	$1.43e-28$	$1.64e+02$	$5.62e+01$	n/a	n/a
f_{11}	$3.38e+03$	$1.23e+02$	$3.36e+03$	$1.12e+02$	$3.36e+03$	$1.12e+02$	$2.34e+03$	$9.22e+01$	$6.78e-04$	$3.60e-05$	$1.67e+03$	$1.44e+02$	n/a	n/a
f_{12}	$7.72e+02$	$8.66e+01$	$7.93e+02$	$9.52e+01$	$7.93e+02$	$9.52e+01$	$1.02e+03$	$6.68e+01$	$6.80e-09$	$8.58e-10$	$1.62e+03$	$1.83e+03$	n/a	n/a
f_{13}	$2.50e+06$	$1.25e+07$	$1.96e+03$	$6.20e+02$	$1.96e+03$	$6.20e+02$	$2.49e+03$	$3.13e+02$	$3.60e+02$	$9.23e+00$	$3.41e+08$	$4.29e+08$	n/a	n/a
f_{14}	$3.12e+03$	$2.07e+02$	$3.18e+03$	$2.10e+02$	$3.18e+03$	$2.10e+02$	$1.67e+03$	$1.51e+02$	$3.93e-01$	$1.05e+00$	$1.59e+03$	$1.57e+02$	n/a	n/a
f_{15}	$2.11e+01$	$7.92e+00$	$2.30e+01$	$7.36e+00$	$2.30e+01$	$7.36e+00$	$4.44e+01$	$5.59e+00$	$2.93e-18$	$7.16e-18$	$3.50e+01$	$1.20e+01$	n/a	n/a
f_{16}	$1.61e+03$	$1.37e+02$	$1.59e+03$	$1.69e+02$	$1.59e+03$	$1.59e+02$	$2.02e+03$	$8.60e+01$	$2.05e-08$	$1.64e-09$	$1.92e+03$	$1.44e+03$	n/a	n/a
f_{17}	$3.28e+03$	$2.30e+02$	$3.25e+03$	$1.71e+02$	$3.25e+03$	$1.71e+02$	$3.83e+03$	$1.41e+02$	$1.12e+02$	$1.02e+00$	$6.64e+08$	$1.64e+09$	n/a	n/a
f_{18}	$1.37e+03$	$1.05e+02$	$1.33e+03$	$9.12e+01$	$1.33e+03$	$9.12e+01$	$3.37e+03$	$4.34e+02$	$1.25e-03$	$1.87e-04$	$2.74e+03$	$3.59e+02$	n/a	n/a
f_{19}	$9.51e+01$	$1.67e+01$	$9.57e+01$	$1.86e+01$	$9.57e+01$	$1.86e+01$	$1.29e+02$	$2.34e+01$	$3.35e-21$	$2.15e-21$	$2.05e+03$	$4.03e+03$	n/a	n/a

Table B.13: Average errors of the GPALS-DE, GPALS-PSO and other algorithms for the CEC-2013 test problems $f_1 - f_9$.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
mGPALS-DE									
30	$0.00e+00$	$1.40e+05$	$2.83e+07$	$2.49e+03$	$0.00e+00$	$4.40e+00$	$1.03e+01$	$2.09e+01$	$2.19e+01$
50	$0.00e+00$	$2.78e+05$	$4.89e+08$	$6.43e+03$	$0.00e+00$	$4.36e+01$	$3.17e+01$	$2.11e+01$	$4.10e+01$
GPALS-DE									
30	$0.00e+00$	$1.37e+07$	$1.58e+09$	$2.42e+04$	$0.00e+00$	$2.51e+01$	$1.08e+02$	$2.10e+01$	$2.71e+01$
50	$0.00e+00$	$3.28e+07$	$1.08e+10$	$4.46e+04$	$0.00e+00$	$4.60e+01$	$1.48e+02$	$2.11e+01$	$5.30e+01$
mSHADE									
30	$0.00e+00$	$1.97e+07$	$1.50e+09$	$2.91e+01$	$0.00e+00$	$2.48e+01$	$6.66e+01$	$2.08e+01$	$2.80e+01$
50	$0.00e+00$	$5.08e+07$	$1.59e+10$	$3.85e+01$	$0.00e+00$	$4.49e+01$	$1.39e+02$	$2.09e+01$	$5.56e+01$
SHADE									
30	$0.00e+00$	$9.00e+03$	$4.02e+01$	$1.92e-04$	$0.00e+00$	$5.96e-01$	$4.60e+00$	$2.07e+01$	$2.75e+01$
50	$0.00e+00$	$2.66e+04$	$8.80e+05$	$1.61e-03$	$0.00e+00$	$4.28e+01$	$2.33e+01$	$2.09e+01$	$5.54e+01$
mL-SHADE									
30	$0.00e+00$	$1.11e+07$	$1.17e+08$	$3.22e+03$	$0.00e+00$	$1.69e+01$	$6.96e+01$	$2.08e+01$	$2.65e+01$
50	$0.00e+00$	$2.00e+07$	$1.28e+09$	$1.00e+05$	$0.00e+00$	$4.34e+01$	$1.11e+02$	$2.10e+01$	$5.45e+01$
L-SHADE									
30	$0.00e+00$	$0.00e+00$	$1.64e+00$	$0.00e+00$	$0.00e+00$	$1.17e-06$	$7.79e-01$	$2.08e+01$	$2.64e+01$
50	$0.00e+00$	$1.96e+03$	$1.50e+04$	$0.00e+00$	$0.00e+00$	$4.34e+01$	$1.93e+00$	$2.10e+01$	$5.37e+01$
DEcfbLS									
30	$0.00e+00$	$1.99e+05$	$2.11e+06$	$3.82e+02$	$0.00e+00$	$7.08e+00$	$5.68e+01$	$2.09e+01$	$2.40e+01$
50	$0.00e+00$	$6.55e+05$	$2.20e+08$	$1.21e+03$	$0.00e+00$	$4.34e+01$	$1.05e+02$	$2.11e+01$	$4.71e+01$
jande									
30	$0.00e+00$	$1.29e+05$	$9.84e+06$	$1.97e+04$	$1.26e-08$	$7.93e+00$	$9.82e+00$	$2.09e+01$	$2.10e+01$
50	$2.76e-08$	$6.05e+05$	$4.78e+07$	$8.34e+04$	$2.43e-06$	$4.30e+01$	$2.94e+01$	$2.11e+01$	$5.33e+01$
DE_APC									
30	$0.00e+00$	$1.75e+05$	$3.21e+06$	$2.20e-01$	$0.00e+00$	$9.35e+00$	$2.18e+01$	$2.09e+01$	$3.07e+01$
50	$0.00e+00$	$3.60e+05$	$6.98e+06$	$1.53e+00$	$0.00e+00$	$3.90e+01$	$3.66e+01$	$2.11e+01$	$6.09e+01$
PVADE									
30	$0.00e+00$	$2.12e+06$	$1.65e+03$	$1.70e+04$	$1.40e-07$	$8.29e+00$	$1.29e+00$	$2.09e+01$	$6.30e+00$
50	$0.00e+00$	$2.04e+05$	$7.48e+06$	$2.20e+02$	$1.39e-03$	$7.36e+01$	$2.07e+01$	$2.11e+01$	$2.60e+01$
GPALS-PSO									
30	$0.00e+00$	$1.40e+05$	$2.83e+07$	$2.49e+03$	$0.00e+00$	$4.40e+00$	$1.03e+01$	$2.09e+01$	$2.19e+01$
50	$0.00e+00$	$9.18e+06$	$6.62e+08$	$3.93e+04$	$0.00e+00$	$4.72e+01$	$9.45e+01$	$2.11e+01$	$6.00e+01$
SPSO2011									
30	$0.00e+00$	$3.38e+05$	$2.88e+08$	$3.86e+04$	$5.42e-04$	$3.79e+01$	$8.79e+01$	$2.09e+01$	$2.88e+01$
50	$0.00e+00$	$6.93e+05$	$7.50e+08$	$5.17e+04$	$9.89e-04$	$5.38e+01$	$8.92e+01$	$2.11e+01$	$5.59e+01$

Table B.14: Average errors of the GPALS-DE, GPALS-PSO and other algorithms for the CEC-2013 test problems $f_{10} - f_{19}$.

	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}
mGPALS-DE										
30	$5.26e-02$	$8.71e+00$	$3.46e+01$	$7.42e+01$	$4.27e+02$	$5.38e+03$	$2.56e+00$	$3.99e+01$	$1.38e+02$	$2.72e+00$
50	$3.61e-02$	$4.24e+01$	$8.39e+01$	$2.01e+02$	$8.79e+02$	$1.04e+04$	$3.27e+00$	$9.34e+01$	$3.10e+02$	$1.44e+01$
GPALS-DE										
30	$6.67e+00$	$0.00e+00$	$1.14e+02$	$1.57e+02$	$2.08e-01$	$4.28e+03$	$1.51e+00$	$3.04e+01$	$1.65e+02$	$1.05e+00$
50	$2.13e+01$	$0.00e+00$	$2.97e+02$	$3.97e+02$	$2.32e-01$	$8.52e+03$	$1.89e+00$	$5.08e+01$	$3.68e+02$	$1.94e+00$
mSHADE										
30	$1.99e+01$	$0.00e+00$	$1.30e+02$	$1.59e+02$	$2.86e-03$	$4.79e+03$	$1.50e+00$	$3.04e+01$	$1.05e+02$	$1.56e+00$
50	$6.11e+01$	$0.00e+00$	$3.43e+02$	$3.10e+02$	$8.08e-03$	$9.32e+03$	$1.71e+00$	$5.08e+01$	$2.02e+02$	$2.87e+00$
SHADE										
30	$7.69e-02$	$0.00e+00$	$2.30e+01$	$5.03e+01$	$3.18e-02$	$3.22e+03$	$9.13e-01$	$3.04e+01$	$7.25e+01$	$1.36e+00$
50	$7.37e-02$	$0.00e+00$	$5.86e+01$	$1.45e+02$	$3.45e-02$	$6.82e+03$	$1.28e+00$	$5.08e+01$	$1.37e+02$	$2.64e+00$
mL-SHADE										
30	$2.15e-02$	$0.00e+00$	$9.46e+01$	$1.34e+02$	$1.27e-02$	$4.36e+03$	$1.32e+00$	$3.04e+01$	$1.65e+02$	$1.62e+00$
50	$7.94e-02$	$0.00e+00$	$1.41e+02$	$2.35e+02$	$3.45e-02$	$7.56e+03$	$1.17e+00$	$5.08e+01$	$2.17e+02$	$1.63e+00$
L-SHADE										
30	$9.67e-04$	$0.00e+00$	$5.48e+00$	$6.34e+00$	$7.63e-02$	$2.68e+03$	$6.54e-01$	$3.04e+01$	$5.23e+01$	$1.21e+00$
50	$1.27e-02$	$4.66e-09$	$1.41e+01$	$2.12e+01$	$4.02e-01$	$6.36e+03$	$1.26e+00$	$5.08e+01$	$1.04e+02$	$2.56e+00$
DEcfbLS										
30	$2.01e-02$	$5.85e-02$	$5.42e+01$	$1.01e+02$	$3.29e+01$	$3.43e+03$	$7.27e-01$	$3.53e+01$	$7.94e+01$	$1.50e+00$
50	$3.21e-02$	$4.64e+00$	$1.34e+02$	$2.47e+02$	$2.31e+02$	$6.25e+03$	$1.63e+00$	$6.58e+01$	$1.57e+02$	$2.95e+00$
jande										
30	$7.91e-02$	$0.00e+00$	$4.28e+01$	$7.08e+01$	$1.33e+00$	$4.83e+03$	$2.28e+00$	$3.04e+01$	$1.23e+02$	$1.10e+00$
50	$1.47e-01$	$1.95e-02$	$9.72e+01$	$1.76e+02$	$8.01e+00$	$9.48e+03$	$3.13e+00$	$5.08e+01$	$2.18e+02$	$2.24e+00$
DE_APC										
30	$6.42e-02$	$3.08e+00$	$3.17e+01$	$7.55e+01$	$3.84e+03$	$4.14e+03$	$2.46e+00$	$5.92e+01$	$6.04e+01$	$2.30e+00$
50	$6.71e-02$	$3.44e+01$	$5.96e+01$	$1.55e+02$	$9.96e+03$	$9.34e+03$	$3.24e+00$	$1.72e+02$	$1.05e+02$	$5.08e+00$
PVADE										
30	$2.16e-02$	$5.84e+01$	$1.15e+02$	$1.31e+02$	$3.20e+03$	$5.61e+03$	$2.39e+00$	$1.02e+02$	$1.82e+02$	$5.40e+00$
50	$5.99e-01$	$1.68e+02$	$2.57e+02$	$3.06e+02$	$7.34e+03$	$1.25e+04$	$3.39e+00$	$2.38e+02$	$3.87e+02$	$2.12e+01$
GPALS-PSO										
30	$5.26e-02$	$8.71e+00$	$3.46e+01$	$7.42e+01$	$4.27e+02$	$5.38e+03$	$2.56e+00$	$3.99e+01$	$1.38e+02$	$2.72e+00$
50	$3.66e-01$	$1.94e+02$	$2.06e+02$	$3.32e+02$	$5.42e+03$	$8.33e+03$	$1.91e+00$	$2.76e+02$	$2.66e+02$	$2.25e+01$
SPSO2011										
30	$3.40e-01$	$1.05e+02$	$1.04e+02$	$1.94e+02$	$3.99e+03$	$3.81e+03$	$1.31e+00$	$1.16e+02$	$1.21e+02$	$9.51e+00$
50	$4.06e-01$	$2.36e+02$	$2.40e+02$	$4.30e+02$	$7.29e+03$	$7.99e+03$	$1.99e+00$	$3.07e+02$	$2.98e+02$	$3.69e+01$

Table B.15: Average errors of the GPALS-DE, GPALS-PSO and other algorithms for the CEC-2013 test problems $f_{20} - f_{28}$.

		f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}	f_{28}
mGPALS-DE	30	$1.13e+01$	$3.15e+02$	$3.07e+02$	$4.89e+03$	$2.15e+02$	$2.48e+02$	$2.20e+02$	$5.19e+02$	$3.01e+02$
	50	$2.08e+01$	$8.45e+02$	$6.13e+02$	$1.04e+04$	$2.51e+02$	$2.92e+02$	$2.95e+02$	$9.11e+02$	$5.90e+02$
GPALS-DE	30	$1.32e+01$	$2.70e+02$	$4.76e+01$	$4.87e+03$	$2.71e+02$	$2.70e+02$	$2.01e+02$	$8.00e+02$	$3.00e+02$
	50	$2.34e+01$	$3.65e+02$	$2.25e+01$	$9.88e+03$	$3.41e+02$	$3.37e+02$	$2.03e+02$	$1.59e+03$	$4.00e+02$
mSHADE	30	$1.12e+01$	$2.26e+02$	$4.62e+01$	$5.57e+03$	$2.75e+02$	$2.93e+02$	$2.01e+02$	$6.07e+02$	$3.00e+02$
	50	$2.12e+01$	$2.74e+02$	$3.83e+01$	$1.10e+04$	$3.50e+02$	$3.88e+02$	$2.02e+02$	$1.57e+03$	$4.00e+02$
SHADE	30	$1.05e+01$	$3.09e+02$	$9.81e+01$	$3.51e+03$	$2.05e+02$	$2.59e+02$	$2.02e+02$	$3.88e+02$	$3.00e+02$
	50	$1.93e+01$	$8.45e+02$	$1.33e+01$	$7.63e+03$	$2.34e+02$	$3.40e+02$	$2.58e+02$	$9.36e+02$	$4.58e+02$
mL-SHADE	30	$1.20e+01$	$4.23e+02$	$1.11e+02$	$4.96e+03$	$2.68e+02$	$2.87e+02$	$2.00e+02$	$1.00e+03$	$3.00e+02$
	50	$2.24e+01$	$3.39e+02$	$1.39e+01$	$9.31e+03$	$3.47e+02$	$3.82e+02$	$2.06e+02$	$1.74e+03$	$4.00e+02$
L-SHADE	30	$9.48e+00$	$2.96e+02$	$1.09e+02$	$2.51e+03$	$2.00e+02$	$2.40e+02$	$2.00e+02$	$3.02e+02$	$3.00e+02$
	50	$1.78e+01$	$8.27e+02$	$1.47e+01$	$5.60e+03$	$2.12e+02$	$2.77e+02$	$2.33e+02$	$4.19e+02$	$4.00e+02$
DEcfbLS	30	$1.17e+01$	$3.36e+02$	$2.56e+02$	$3.59e+03$	$2.64e+02$	$2.83e+02$	$2.00e+02$	$9.38e+02$	$3.00e+02$
	50	$2.17e+01$	$5.24e+02$	$6.89e+02$	$7.77e+03$	$3.31e+02$	$3.60e+02$	$2.00e+02$	$1.55e+03$	$4.00e+02$
jande	30	$1.16e+01$	$2.94e+02$	$5.16e+01$	$4.61e+03$	$2.48e+02$	$2.60e+02$	$2.58e+02$	$7.22e+02$	$3.00e+02$
	50	$2.15e+01$	$8.24e+02$	$3.10e+01$	$9.48e+03$	$2.89e+02$	$3.17e+02$	$3.97e+02$	$1.16e+03$	$9.43e+02$
DE_APC	30	$1.26e+01$	$2.67e+02$	$4.56e+03$	$4.18e+03$	$2.92e+02$	$2.99e+02$	$3.28e+02$	$1.19e+03$	$3.00e+02$
	50	$2.23e+01$	$6.81e+02$	$1.06e+04$	$9.09e+03$	$3.84e+02$	$3.83e+02$	$4.09e+02$	$2.14e+03$	$6.97e+02$
PVADE	30	$1.13e+01$	$3.19e+02$	$2.50e+03$	$5.81e+03$	$2.02e+02$	$2.30e+02$	$2.18e+02$	$3.26e+02$	$3.00e+02$
	50	$2.07e+01$	$9.65e+02$	$7.72e+03$	$1.18e+04$	$2.78e+02$	$3.54e+02$	$3.47e+02$	$1.11e+03$	$4.62e+02$
GPALS-PSO	30	$1.13e+01$	$3.15e+02$	$3.07e+02$	$4.89e+03$	$2.15e+02$	$2.48e+02$	$2.20e+02$	$5.19e+02$	$3.01e+02$
	50	$2.22e+01$	$4.03e+02$	$7.03e+03$	$1.00e+04$	$3.52e+02$	$4.00e+02$	$2.16e+02$	$1.88e+03$	$4.67e+02$
SPSO2011	30	$1.35e+01$	$3.09e+02$	$4.30e+03$	$4.83e+03$	$2.67e+02$	$2.99e+02$	$2.86e+02$	$1.00e+03$	$4.01e+02$
	50	$2.25e+01$	$8.78e+02$	$9.10e+03$	$1.04e+04$	$3.43e+02$	$4.03e+02$	$3.92e+02$	$1.69e+03$	$9.13e+02$

Author's Publications

- Tatsis, V.A., Parsopoulos, K.E., Differential Evolution with Grid-Based Parameter Adaptation, Soft Computing, 21 (8), pp. 2105-2127, Springer, 2017.
- Tatsis, V.A., Parsopoulos, K.E., Experimental Assessment of Differential Evolution with Grid-Based Parameter Adaptation, International Journal on Artificial Intelligence Tools, 27 (4), pp. 1-20, World Scientific, 2018.
- Tatsis, V.A., Parsopoulos, K.E., Dynamic Parameter Adaptation in Metaheuristics Using Gradient Approximation and Line Search, Applied Soft Computing, 74, pp. 368-384, Elsevier, 2019.
- Tatsis, V.A., Parsopoulos, K.E., Skouri, K., Konstantaras, I.,An Ant-Based Optimization Approach for Inventory Routing, Advances in Intelligent Systems and Computing, Vol. 227, M. Emmerich et al. (eds.), pp. 107-121, 2013, Springer.
- Piperagkas, G.S., Voglis, C., Tatsis, V.A., Parsopoulos, K.E., Skouri, K., Applying PSO and DE on Multi-Item Inventory Problem with Supplier Selection, The 9th Metaheuristics International Conference (MIC 2011), Udine, Italy, pp. 359-368, 2011.
- Tatsis, V.A., Parsopoulos, K.E., Grid Search for Operator and Parameter Control in Differential Evolution, 9th Hellenic Conference on Artificial Intelligence (SETN 2016), Thessaloniki, Greece, pp. 1-9, 2016, ACM International Conference Proceeding Series.
- Tatsis, V.A., Parsopoulos, K.E., Grid-Based Parameter Adaptation in Particle Swarm Optimization, 12th Metaheuristics International Conference (MIC 2017), Barcelona, Spain, pp. 217-226, 2017.

- Tatsis, V.A., Parsopoulos, K.E., On the Sensitivity of Grid-Based Parameter Adaptation Method, 7th International Conference on Metaheuristics and Nature Inspired Computing (META 2018), Marrakech, Morocco, pp. 86-94, 2018.

Short Biography

Vasileios Tatsis was born in Ioannina in 1990. He received his BSc and MSc Degrees from the Department of Computer Science and Engineering of the University of Ioannina, Greece in 2011 and 2013, respectively. In 2013 he was admitted to the PhD program of the same Department. He received scholarships regarding his entrance rank in the Department as well as a distinction as the second best student in his graduation year. His research focuses on Computational Optimization and, specifically, on automatic parameterization of metaheuristics.