Improving Differential Evolution with Adaptive and Local Search Methods

Miguel Leon
IMPROVING DIFFERENTIAL EVOLUTION WITH ADAPTIVE AND LOCAL SEARCH METHODS

Miguel Leon

2019

School of Innovation, Design and Engineering
IMPROVING DIFFERENTIAL EVOLUTION WITH
ADAPTIVE AND LOCAL SEARCH METHODS

Miguel Leon

Akademisk avhandling

som för avläggande av teknologie doktorsexamen i datavetenskap vid
Akademin för innovation, design och teknik kommer att offentligen försvaras
onsdagen den 18 december 2019, 13.15 i Delta, Mälardalens högskola, Västerås.

Fakultetsopponent: Adjungerad professor Kristian Sandström, RISE, Västerås
Abstract

Differential Evolution (DE) is a population-based algorithm that belongs to the Evolutionary algorithm family. During recent years, DE has become a popular algorithm in optimization due to its strength solving different types of optimization problems and due to its easy usage and implementation.

However, how to choose proper mutation strategy and control parameters for DE presents a major difficulty in many real applications. Since both mutation strategy and DE control parameters are highly problem dependent, they have to be adapted to suite different search spaces and different problems. Failure in proper assignment for them will cause slow convergence in search or stagnation with a local optimum.

Many researches have been conducted to tackle the above issues. The major efforts have been made in the following three directions. First, some works have been proposed that adapt the selection between various mutation strategies. But the choice of strategies in these methods has not considered the difference of quality of individuals in the population, which means that all individuals will acquire the same probability to select a mutation strategy from the candidates. This does not seem a very desired practice since solutions of large difference would require different mutation operators to reach improvement. Second, many works have been focusing on the adaptation of the control parameters of DE (mutation factor (F) and crossover rate (CR)). They mainly rely on previous successful F and CR values to update the probability functions that are used to generate new F and CR values. By doing this, they ignore the stochastic nature of the operators in DE such that weak F and CR values can also get success in producing better trial solutions. The use of such imprecise experiences of success would prevent the DE parameters from being adapted towards the most effective values in coming generations. Third, various local search methods have been incorporated into DE to enhance exploitation in promising regions so as to speed up the convergence to optima. It is important to properly adjust the characteristics of the local search in DE to achieve well balanced exploratory/exploitative behavior to solve complex optimization problems.

This thesis aims to further improve the performance of DE by new adaptation and local search methods. The main results can be summarized in the following three aspects:

1) Proposal of a new rank-based mutation adaptation method, which takes into account the quality of solutions in the population when adapting the selection probabilities of mutation strategies. This makes possible to treat solutions with distinct ranks (in quality) differently by using different selection probabilities for mutation operators.

2) Development of improved parameter adaptation methods for DE, which emphasizes more reliable and fair evaluation of candidates (F and CR assignments) during the search process. It is suggested that greedy search being used as a fast and cheap technique to look for better parameter assignment for F and CR respectively in the neighborhood of a current candidate. Further, a joint parameter adaptation method is proposed that enables continuous update of the selection probabilities for F and CR pairs based on feedback acquired during the search.

3) Proposal of new methods for better incorporation of local search into a DE algorithm. The Eager Random Search method is investigated as local search inside DE, which exhibits different exploratory-exploitative characteristics by using different probability density functions. More importantly, we propose a novel memetic framework in which Alopex local search (ALS) is performed in collaboration with a DE algorithm. The framework favors seamless connection between exploration and exploitation in the sense that the behavior of exploitation by ALS can be controlled by the status of global exploration by DE.

The proposed methods and algorithms have been tested in a number of benchmark problems, obtaining competitive results compared with the state-of-the-art algorithms. Additionally, the Greedy Adaptive DE (GADE) algorithm (developed based on greedy search for DE parameters) has been tested in a real industrial problem, i.e., finding best component parameters to optimize the performance of harmonic filters for power transmission. GADE is shown to produce better harmonic filter systems with lower harmonic distortion than the standard DE.
Abstract

Differential Evolution (DE) is a population-based algorithm that belongs to the Evolutionary algorithm family. During recent years, DE has become a popular algorithm in optimization due to its strength solving different types of optimization problems and due to its easy usage and implementation.

However, how to choose proper mutation strategy and control parameters for DE presents a major difficulty in many real applications. Since both mutation strategy and DE control parameters are highly problem dependent, they have to be adapted to suite different search spaces and different problems. Failure in proper assignment for them will cause slow convergence in search or stagnation with a local optimum.

Many researches have been conducted to tackle the above issues. The major efforts have been made in the following three directions. First, some works have been proposed that adapt the selection between various mutation strategies. But the choice of strategies in these methods has not considered the difference of quality of individuals in the population, which means that all individuals will acquire the same probability to select a mutation strategy from the candidates. This does not seem a very desired practice since solutions of large difference would require different mutation operators to reach improvement. Second, many works have been focusing on the adaptation of the control parameters of DE (mutation factor (F) and crossover rate (CR)). They mainly rely on previous successful F and CR values to update the probability functions that are used to generate new F and CR values. By doing this, they ignore the stochastic nature of the operators in DE such that weak F and CR values can also get success in producing better trial solutions. The use of such imprecise experiences of success would prevent the DE parameters from being adapted towards the most effective values in coming generations. Third, various local search methods have been incorporated into DE to enhance exploitation in promising regions so as to speed up the convergence to optima. It is important to properly adjust the characteristics of the local search in DE to achieve well balanced exploratory/exploitative behavior to solve complex optimization
problems.

This thesis aims to further improve the performance of DE by new adaptation and local search methods. The main results can be summarized in the following three aspects:

1) Proposal of a new rank-based mutation adaptation method, which takes into account the quality of solutions in the population when adapting the selection probabilities of mutation strategies. This makes possible to treat solutions with distinct ranks (in quality) differently by using different selection probabilities for mutation operators.

2) Development of improved parameter adaptation methods for DE, which emphasizes more reliable and fair evaluation of candidates (F and CR assignments) during the search process. It is suggested that greedy search being used as a fast and cheap technique to look for better parameter assignment for F and CR respectively in the neighborhood of a current candidate. Further, a joint parameter adaptation method is proposed that enables continuous update of the selection probabilities for F and CR pairs based on feedback acquired during the search.

3) Proposal of new methods for better incorporation of local search into a DE algorithm. The Eager Random Search method is investigated as local search inside DE, which exhibits different exploratory-exploitative characteristics by using different probability density functions. More importantly, we propose a novel memetic framework in which Alopex local search (ALS) is performed in collaboration with a DE algorithm. The framework favors seamless connection between exploration and exploitation in the sense that the behavior of exploitation by ALS can be controlled by the status of global exploration by DE.

The proposed methods and algorithms have been tested in a number of benchmark problems, obtaining competitive results compared with the state-of-the-art algorithms. Additionally, the Greedy Adaptive DE (GADE) algorithm (developed based on greedy search for DE parameters) has been tested in a real industrial problem, i.e., finding best component parameters to optimize the performance of harmonic filters for power transmission. GADE is shown to produce better harmonic filter systems with lower harmonic distortion than the standard DE.
Sammanfattning

Differential Evolution (DE) är en populationsbaserad algoritm som tillhör familjen för evolutionära algoritmer. På senare år har DE blivit en populär algoritm för optimering på grund av sin framgång vid lösning av olika typer av optimeringsproblem och tack vare sin enkla användning och implementation.

Trots detta medför stora svårigheter vid val av lämplig strategi för mutationer och val av kontrollparametrar när DE tillämpas på verkliga problem. Då både val av strategi för mutation och kontrollparametrar är beroende på problemen, med huvudsaklig anpassning av dessa parametrar kan utfallet innebära långsam konvergens eller stagnation på grund av lokalt optimum.

på ett lämpligt sätt så att balansen mellan exploatering och utforskning av nya områden passar väl för lösning av komplexa optimeringsproblem.

Denna avhandling fokuserar på att förbättra prestandan hos DE genom nya anpassningar och nya lokala sökmetoder. Resultaten kan i huvudsak sammanfattas i följande tre aspekter:

1) Förslag på en ny metod för rankbaserad anpassning av mutation, som tar hänsyn till kvaliteten på lösningar i populationen vid anpassning av sannolikheter för mutationsstrategier. Detta möjliggör hantering av lösningar av olika rang (med avseende på kvalitet) på olika sätt genom användande av olika urval och sannolikheter för mutationsoperatorer.

2) Utveckling av förbättrade metoder för anpassning av DE parametrar, med tonvikt på mer tillförlitlig och rättvis evaluering av kandidater (tilldelning av F och CR) i sökprocessen. Det föreslås att greedy search kan användas som en snabb och kostnadseffektiv teknik för att hitta bättre tilldelning av parametrarna F och CR i närheten av en kandidat. Vidare föreslås en metod för gemensam anpassning av parametrar som möjliggör kontinuerlig uppdatering av sannolikheter för urval för F och CR par, baserat på återkoppling inom sökprocessen.


De föreslagna metoderna och algoritmerna har testats på ett flertal referensproblem och visat konkurrenskraftiga resultat jämfört med state-of-the-art algoritmerna. Vidare har Greedy Adaptive DE (GADE) algoritmen (som utvecklats baserat på greedy search för DE parametrar) testats i verkliga industri problem, till exempel för att finna de bästa komponent parametrarna för att optimera prestanda hos harmoniska filter för överföring av elkraft. GADE har påvisats producera bättre harmoniska filtersystem med lägre harmoniska störningar jämfört med vanlig DE.
To my family
Acknowledgements

First and foremost, I will like to express my sincere gratitude to my main supervisor Ning Xiong (Mälardalen University) and my two co-supervisors Peter Funk (Mälardalen University) and Yigen Zenlander (ABB FACTS). I have learned a lot from their advices, feedback and guidance, without their support this thesis would not have been possible. I also would like to express the deepest appreciation to my co-authors Magnus Evestedt (Prevas), Yigen Zenlander (ABB FACTS), Francisco Herrera (Granada University), Daniel Molina (Granada University), Joaquin Ballesteros, Elaine Astrand, Fredrik Ekstrand, Carl Ahlberg, Mikael Ekstrom, Jonatan Tidare and Javier Ramos. for their cooperation in the papers. I would like to thank Baran Çürükülu for helping me with the lectures and nice discussions.

I would like to thank all my current and past roommates, Arash Ghareh, Mobyen Uddin Ahmed, Hamidur Rahman, Mohammad Loni, Henrik Falk, Fredrik Ekstrand, Nikola Petrovic, Carl Ahlberg, Lennie Carlén Eriksson, Jonatan Tidare, Martin Ekström, Gregory Koshmak, Jimmie Hagblad and Henrik Johansson. It is a pleasure to share office with you.

I thank the lecturers and professors whom I learned a lot from during meetings, lectures, seminars and PhD courses including Thomas Nolte, Gordana Dodig-Crnkovic, Jan Gustafsson and Mikael Sjödin. You all have been a source of inspiration for me. I thank the administrative staff in particular, Malin Åshuvud, Sofia Jäderén, Carola Ryttersson and Jenny Hägglund for making my live easier.

I would like to thank my friends and colleagues at the division of IFT: Adnan, Alessandro, Ankica, Branko, Farid, Fredrik, Giacomo, Håkan, Hamid, Ivan, Johan, José-Fernán, LanAnh, Maria Ehn, Maria Lindén, Masoud, Mia, Mir Riyanul, Mirgita, Mirko, Mohammad, Per, Per Olov, Shahina, Shaibal, Sharmin, Sima and others. Also, I would like to thank all the other people from IDT.

Last but not least, a huge thank to my family for their support even from the long distance. Also I would like to thank my girlfriend and best friend Laura,
without your support this would be impossible.

Miguel Leon, Västerås, 2019
List of Publications

Papers included in the Ph.D. thesis:


**Paper D: Adaptive Differential Evolution with a New Joint Parameter Adaptation Method.** Miguel Leon, Ning Xiong, Submitted to a journal.


Part of this thesis (Paper E) was previously included in the licentiate thesis "Enhancing Differential Evolution Algorithm for Solving Continuous Optimization Problems" (Malardalaen University Licentiate thesis 246, Miguel Leon, 2016), as well as some figures.

Additional papers, not included in the Ph.D. thesis:

- **Discriminating EEG spectral power related to mental imagery of closing and opening of hand**: Jonatan Tidare, Miguel Leon, Ning Xiong, Elaine Astrand, THe 9th International IEEE EMBS Conference of Neural Engineering (IEEE NER’19), 2019.


- **Alopex-based mutation strategy in differential evolution**: Miguel Leon, Ning Xiong, IEEE Congress on Evolutionary Computation (CEC), 2017.


- **A New Differential Evolution Algorithm with Alopex-Based Local Search**: Miguel Leon, Ning Xiong, International Conference on Artificial Intelligence and Soft Computing, 2016.

- **Differential evolution based on decomposition for solving multi-objective optimization problems**: Ning Xiong, Miguel Leon, 8th International Conference on Agents and Artificial Intelligence (ICAART), 2016.


- **A Walk into Metaheuristics for Engineering Optimization: Principles, Methods and Recent Trends**: Ning Xiong, Daniel Molina, Miguel

- **Greedy adaptation of control parameters in differential evolution for global optimization problems**: Miguel Leon, Ning Xiong, Evolutionary Computation (CEC), 2015 IEEE Congress on, p. 385-392.

- **Application of adaptive differential evolution for model identification in furnace optimized control system**: Miguel Leon, Ning Xiong, International Conference on Evolutionary Computation Theory and Applications (ECTA), 2015.


- **Principles and state-of-the-art of engineering optimization techniques**: Ning Xiong, Miguel Leon, Proc. The 7th International Conference on Advanced Engineering Computing and Applications in Sciences, 2013, p. 36-42.
List of Figures

1.1 Categories of metaheuristic optimization algorithms ........ 4
2.1 Differential Evolution process ........................................ 9
2.2 Two-dimensional example of 6 different mutation strategies. . 11
2.3 Example of binomial crossover ....................................... 12
2.4 Example of exponential crossover (k=3,L=3) .................... 12
4.1 Overview of the relation between the Research Contributions and DE. G represent the maximum number of generations . . . 27
5.1 Overview of the followed research methodology .................. 35
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALS</td>
<td>Alopex Local Search</td>
</tr>
<tr>
<td>CLS</td>
<td>Cauchy Local Search</td>
</tr>
<tr>
<td>CR</td>
<td>Crossover Rate</td>
</tr>
<tr>
<td>DE</td>
<td>Differential Evolution</td>
</tr>
<tr>
<td>EA</td>
<td>Evolutionary Algorithm</td>
</tr>
<tr>
<td>ERS</td>
<td>Eager Random Search</td>
</tr>
<tr>
<td>F</td>
<td>mutation Factor</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GADE</td>
<td>Greedy Adaptive Differential Evolution</td>
</tr>
<tr>
<td>JAPDE</td>
<td>Joint Adaptation of Parameters in Differential Evolution</td>
</tr>
<tr>
<td>LIP</td>
<td>Local Improvement Process</td>
</tr>
<tr>
<td>LLS</td>
<td>Local Improvement process oriented Local Search</td>
</tr>
<tr>
<td>LS</td>
<td>Local Search</td>
</tr>
<tr>
<td>MA</td>
<td>Memetic Algorithm</td>
</tr>
<tr>
<td>MFDEALS</td>
<td>Memetic Framework for enhancing Differential Evolution algorithms with Alopex Local Search</td>
</tr>
<tr>
<td>NLS</td>
<td>Normal Local Search</td>
</tr>
<tr>
<td>PS</td>
<td>Population Size</td>
</tr>
<tr>
<td>RAM</td>
<td>Ranked-based Adaptation of Mutation strategies</td>
</tr>
<tr>
<td>RC</td>
<td>Research Contribution</td>
</tr>
<tr>
<td>RLS</td>
<td>Random Local Search</td>
</tr>
<tr>
<td>RQ</td>
<td>Research Question</td>
</tr>
<tr>
<td>XLS</td>
<td>Crossover-based Local Search</td>
</tr>
</tbody>
</table>
# Contents

I Thesis 1

1 Introduction 3
   1.1 Thesis contribution .................................. 6
   1.2 Thesis outline ........................................ 6

2 Differential Evolution 9
   2.1 Mutation .............................................. 10
   2.2 Crossover .............................................. 11
   2.3 Selection .............................................. 12

3 Related Work 15
   3.1 Adaptation of mutation strategies in Differential Evolution ........................................ 15
   3.2 Parameter adaptation in Differential Evolution ........................................ 17
   3.3 Hybridization between local search methods and Differential Evolution ........................................ 19
       3.3.1 Strategies to assign resources to local search ........................................ 19
       3.3.2 Local search methods in DE ........................................ 20

4 Research Questions and Contributions 23
   4.1 Research Questions ...................................... 23
   4.2 Research Contribution .................................... 24
   4.3 Overview of the Papers .................................... 27

5 Research Methodology 33

6 Conclusion and Future Work 37

Bibliography 41
II  Included Papers  47
I

Thesis
Chapter 1

Introduction

Optimization is an important part of industrial development and manufacturing [1, 2, 3, 4]. Industries need to optimize different problems such as, logistics (reducing the time of shipments, better usage of employers’ working hours, etc.), economics (reducing the manufacturing cost) or performance (e.g. Finding the best parameters in circuit design, furnaces, etc., so that they obtain the best performance while reducing losses). Optimization can play an important role in industry by finding good solutions to the different optimization tasks. However, all these are not a trivial task due to its complexity and the huge number of possible combinations (huge search space).

For the majority of these problems, the derivative information is not available, making impossible the usage of classical optimization algorithms as gradient descent [5] or Newton’s method [6]. A different solution would be to do an exhaustive search, applying strategies like grid-search or tree-search. However, due to the high number of possibilities (in most cases, it tends to infinity), these algorithms are impractical. For this reason, an approximate algorithm has to be applied, which basically tries to combine basic heuristic methods in higher level frameworks [7]. These heuristics methods are called metaheuristics.

The usage of Metaheuristics does not guarantee to find the global optimum, but instead, a sufficiently good solution is found. These methods aim to efficiently and effectively explore the search space to find the best possible solution [7]. Metaheuristic can be divided into two categories: trajectory-based optimization approaches and population-based optimization approaches (see Fig. 1.1). The former can be divided into single trajectory-based and multi trajectory-based. Single trajectory-based optimization algorithms, as hill-climbing [8] or simulated annealing [9], will move from one solution to a different one, keeping only one solution at a time. Multi trajectory-based optimization algorithms, such as iterative local search [10], randomized adaptive
search procedure [11] or iterative greedy search [12], are strategies that will re-start the search in order to try to find a better solution. In population-based approaches, a set of solutions, also called population, are manipulated with a new population being generated from an old one (in order to find an optimal solution). Population-based approaches have a nice quality compared to trajectory-based ones when solving any optimization problem. Since a number of points of the search space are considered at the same time, population-based approaches performs parallel beam search and thus have stronger global exploration capability. For this reason, if a problem with many local optima is considered, trajectory-based approaches will be more likely to get stuck into a local optimum compared to population-based approaches.

As an important family of population-based methods, Evolutionary Algorithms (EAs) [13] have been proven to be a powerful tool when dealing with complex and high dimensional optimization problems. Their successful applications include calibration in furnace optimized control system [14], process modeling for greenhouses [15], enhancing the contrast and brightness of satellite images [16], as well as controlling the reactive power and voltage of an electric network [2], to mention a few. EA is a big family of algorithms, among which well known techniques are Genetic Algorithm (GA) [17, 18] and Differential Evolution (DE) [19].

DE, as other algorithms in the EA family, is based on the theory of evolution, in which the fittest individuals will survive through the generations. Differently from other EAs, DE explores the search space by calculating differences between random members of the population. DE is easy to use and implement and it is particularly competent to deal with real-parameter optimization problems. Moreover, some researchers have indicated that DE was
more efficient and more accurate than some other EAs, such as GAs [19, 20].

However, how to choose proper mutation strategy and control parameters for DE presents a major difficulty in solving real application problems. Since both mutation strategy and DE parameters are highly problem dependent, they have to be adapted to suite different landscapes of different problems. Manual selection of mutation strategy and tuning of control parameters for DE is a time consuming task. Failure to assign appropriate mutation operator and control parameters will cause imbalanced relation between exploration and exploitation during the search, leading to slow convergence or stagnation into a local optimum [21].

Many researchers have dealt with the above issue by developing methods and algorithms which can be injected into DE to achieve a good trade-off between exploration and exploitation automatically with the DE behavior. The research efforts have been made in the following three directions:

- There exist a variety of mutation strategies, each one with a different exploration/exploitation behavior. Some researchers have proposed to adapt the selection among different mutation strategies [22, 23]. However, these methods do not distinguish different quality of individuals of the population, which means that all individuals get the same probability to select a mutation strategy from the candidates. This would not be very consistent with some situations in which the best and worst solutions would need different operators of mutation to reach improvement.

- Parameter adaptation has attracted hot attention where the majority of works have focused on adjusting the mutation factor (F) and crossover rate (CR) parameters while fixing the population size (PS) at a constant [22, 24, 25]. A common property of the proposed methods is that they rely on the previous successful F and CR values to update the probability functions that are used to generate F and CR values. By doing this, they ignore the stochastic nature of operators in DE such that weak F and CR values can also get success in producing better trial solutions. The use of imprecise experience of success would prevent DE parameters from being adapted to the most effective values in the following generations.

- A number of works were conducted to hybridize DE with a local search (LS) mechanism [26, 21, 27]. The combined methods, called memetic algorithms (MAs), apply local refinements after evolutionary operators in order to exploit promising regions of the search space. LS can enhance the exploitation of DE so as to accelerate the speed of convergence to an optimal solution. On the other hand, the fitness evaluations required in
LS will also decrease the computing resources for the global exploration by DE.

1.1 Thesis contribution

This thesis aim to further improve the performance of DE by proposal and development of new adaptation and LS methods. The main results can be summarized in the following three aspects:

- Regarding mutation strategy adaptation, a new rank-based adaptation method is proposed. The merit of this new method is that it takes into account the quality of solutions in the population when adapting the selection probabilities of mutation strategies. Consequently two solutions with big differences (in quality) may be treated differently following different selection probabilities for mutation operators.

- Regarding DE parameters, improved adaptation methods are developed which emphasizes more reliable and fair evaluation of candidates (F and CR values ) during the search process. It is suggested that greedy search being used as a fast and cheap technique to look for better parameter assignment for F and CR respectively in the neighborhood of the current candidate. Further, in consideration of the combined effect of F and CR values, we propose the joint parameter adaptation method that enables continuous update of the selection probabilities for F and CR combinations based on feedback acquired during the search.

- Regarding LS embedded in DE, the Eager Random Search method is investigated which enables different exploratory/exploitative behavior by using different probability distributions. Further, we propose a novel memetic framework in which Alopex LS is performed in collaboration with a DE algorithm. The framework favors seamless connection between exploration and exploitation in the sense that the behavior of exploitation by LS can be controlled by the status of global exploration by DE.

1.2 Thesis outline

The rest of the thesis is structured as follows. In Chapter 2, the basic knowledge of DE is described. In chapter 3, a review of the most well-known DE algorithms is presented. The research questions and contributions of the thesis are
disseminated in Chapter 4. The followed research methodology is described in Chapter 5. The conclusion and future work are given in Chapter 6. Finally, the papers, in which this thesis is based are shown in Chapters 7-12.
Chapter 2

Differential Evolution

Differential Evolution (DE) is a population-based optimization algorithm, in which each individual from the population \((X)\) is a solution to the optimization problem that we are solving. The \(i\)th individual in the population, at generation \(g\), is expressed as \(X_{i,g} = \{x_1, x_2, \ldots, x_n\}\), where \(n\) is the dimension of the problem. Then, all the \(PS\) individuals from the population will undergo through the DE cycle, three steps denominated mutation, crossover and selection. One of these cycles is equal to one generation. The DE cycle is presented in Fig. 2.1.

![Figure 2.1: Differential Evolution process](image)

There are several versions of DE, depending on the selected mutation and crossover strategy. A version of DE is denoted as \(DE/x/y/z\) where \(x\) stands for the selected mutation strategy, \(y\) is a number representing the number of vector’s differences and \(z\) stands for the crossover strategy applied. These three
steps are broken down in the following Subsections.

### 2.1 Mutation

In mutation, $PS$ different mutated vectors ($V_g$), also called donor vectors, are generated at generation $g$. As previously mentioned, there are several ways of performing mutation. Some of them are shown below:

- **DE/rand/1**:
  
  $$V_{i,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g})$$  \hfill (2.1)

- **DE/rand/2**:
  
  $$V_{i,g} = X_{r1,g} + F \cdot (X_{r2,g} - X_{r3,g} + X_{r4,g} - X_{r5,g})$$  \hfill (2.2)

- **DE/best/1**:
  
  $$V_{i,g} = X_{best,g} + F \cdot (X_{r1,g} - X_{r2,g})$$  \hfill (2.3)

- **DE/best/2**:
  
  $$V_{i,g} = X_{best,g} + F \cdot (X_{r1,g} - X_{r2,g} + X_{r3,g} - X_{r4,g})$$  \hfill (2.4)

- **DE/current-to-rand/1**
  
  $$V_{i,g} = X_{i,g} + F \cdot (X_{r1,g} - X_{i,g} + X_{r2,g} - X_{r3,g})$$  \hfill (2.5)

- **DE/current-to-best/1**
  
  $$V_{i,g} = X_{i,g} + F \cdot (X_{best,g} - X_{i,g} + X_{r1,g} - X_{r2,g})$$  \hfill (2.6)

where $X_{r1,g}$, $X_{r2,g}$, $X_{r3,g}$, $X_{r4,g}$ and $X_{r5,g}$ are random individuals from the population, $X_{best,g}$ stands for the best individual in the population and $X_{i,g}$ is the $i$th individual from the population. Additionally, $F$ stands for the mutation factor, a value between 0 and 2 [28]. One example per mutation strategy, in a two-dimensional problem, can be found in Fig. 2.2.
2.2 Crossover

The second step of the DE cycle is called crossover. Same as in mutation, $PS$ different offspring ($T_g$), also called trial vectors, are created at generation $g$. In order to perform crossover, there are two main alternatives: Binomial and Exponential. To create the $i$th offspring ($T_{i,g}$) following the Binomial crossover Eq. (2.7) has to be followed.

\[
T_{i,g}[j] = \begin{cases} 
V_{i,g}[j] & \text{if } \text{rand}(0,1) \leq CR \quad \& \quad j = \text{randIndex} \\
X_{i,g}[j] & \text{otherwise} 
\end{cases} \quad (2.7)
\]

where $j$ is the $j$th dimension of the different solutions, $\text{rand}(0,1)$ is a random value that follows an uniform distribution inside the range $[0,1]$ and $CR$ stands for the crossover rate, a value inside the range $[0,1]$. Additionally, $\text{randIndex} \in \{1,2,\ldots,n\}$ is a random integer value that force to select at least one parameter from the offspring. An example is given in Fig. 2.3

The exponential crossover constructs the offspring vector containing a sequence of consecutive components (in a circular manner) taken from the mutant vector. The offspring vector is derived, in Eq. (2.8), as follows

\[
\text{Figure 2.2: Two-dimensional example of 6 different mutation strategies.}
\]
Chapter 2. Differential Evolution

Figure 2.3: Example of binomial crossover

\[ T_{i,g}[j] = \begin{cases} V_{i,g}[j] & \text{if } j \in \{k, \{k + 1\}_n, \{k + 2\}_n, \ldots, \{k + L - 1\}_n\} \\ X_{i,g}[j] & \text{otherwise} \end{cases} \]

(2.8)

where \(k\) and \(L\) are two randomly selected integers in \(\{1, 2, \ldots, (n - 1)\}\), \(n\) stands for the dimension of the problem and \(L\) represents the number of elements that will be taken from the mutant vector. \(\{x\}_n\) is \(x\) if \(x \leq n\) and \(x - n\) if \(x > n\). An example (with \(k=3, L=3\)) is given in Fig. 2.4.

Figure 2.4: Example of exponential crossover \((k=3, L=3)\)

2.3 Selection

The third and last step of the DE cycle is called selection. In this step, the created offsprings will be compared with their corresponding parent to obtain the population for the next generation \((X_{g+1})\). To obtain \(X_{g+1}\), Eq. (2.9) is followed.
where $T_{i,g}$ represents the $i$th offspring and $X_{i,g}$ stands for the $i$th individual from the population. Additionally, $f(T_{i,g})$ and $f(X_{i,g})$ are the fitness values of the $i$th offspring and the $i$th individual from the population, respectively.
Chapter 3

Related Work

In this chapter, a literature review, concerning three different aspects of DE, is given. In Section 3.1, a literature review of adaptation of mutation strategies is presented. Secondly, a review of different alternatives to adapt the main parameters of DE is shown in Section 3.2. Lastly, different ways and methods of hybridization between DE and LS methods are presented in Section 3.3.

3.1 Adaptation of mutation strategies in Differential Evolution

As previously mentioned, a variety of different mutation strategies exist within DE, each one of them with different exploration-exploitation properties. There is no single mutation strategy better than the others for all the optimization problems [29], so it is essential to adapt between different mutation strategies. There are two different ways of adapting between different mutation strategies. The first alternative is to switch between different mutation strategies using the feedback obtained during the search. The second alternative is to mix different mutation strategies into a single one, and then by adapting an extra parameter, the new mutation strategy is generated.

In line with the first alternative, the first adaptive DE, which included some different mutation strategies at the same time, was SaDE, presented in 2005 by Qin [22] and improved in 2009 [30]. The improved version, presented in 2009, aimed for adaptation of four different mutation strategies: DE/rand/1/bin, DE/rand-to-best/2/bin, DE/rand/2/bin and DE/current-to-rand/1. The different mutation strategies were assigned with a probability to different individuals in the population. These probabilities change after each learning period, where the success rate of each mutation strategy was used to decide the new prob-
abilities. Similarly, in ZEPDE [23] proposed in 2016, five different mutation strategies were used in a similar way. The difference between ZEPDE and SaDE is that in ZEPDE, instead of using the success rate as in SaDE, the difference between fitness of the created solutions and the fitness of the worst created solution of the same generation is utilized. In 2011, EPSDE was proposed by Mallipeddi [31]. In EPSDE, three different mutation operators are used: DE/best/2/bin, DE/rand/1/bin and DE/current-to-rand/1. EPSDE randomly selects a mutation strategy for each individual in the population. If a mutation strategy manages to improve an individual, then that individual will use the same mutation strategy. Additionally, that mutation strategy will be saved in a success pool, in which the successful combination of the mutation strategy with the parameters is saved. Otherwise if a mutation strategy does not manage to improve an individual, then the mutation strategy is either randomly reinitialized or replaced by one from the pool of successful mutation strategies. A similar idea is proposed in CoDE [32], in which a pool of three different mutation strategies (DE/rand/1/bin, DE/rand/2/bin and DE/current-to-rand/1) is used. For each individual the three different mutation strategies are used, then the best trial vector of the three is compared with the parent vector when performing the selection. The more recent algorithm NRDE [33] randomly selects between DE/rand/1 and DE/best/1, without using any feedback from the population.

Regarding the second alternative, JADE [24], proposed in 2009, was the first adaptive DE algorithm creating a new mutation operator as a combination of two different strategies. This mutation operator, called DE/current-to-pBest/1/bin, is a combination of two mutation strategies: DE/current-to-rand/1/bin and DE/current-to-best/1/bin. In JADE, the parameter $p$, used to adapt between the different mutation strategies, is fixed during the run time. Likewise, the same strategy with a different $p$ is used in SHADE [25]. Differently from JADE, dn-DADE [34] controls the parameter $p$ using a deterministic rule. Higher values of $p$ are used in early stages of the search, while smaller values are used in final stages.

More recently in 2017, EADE [35] was proposed, which switches with a 0.5 probability between DE/rand/1 and an advanced mutation strategy. This advanced mutation strategy divides the population into three areas: top, middle and bottom, selecting one individual from each area. Then, the differences between these three individuals are calculated to create the offspring. Likewise, a similar idea was proposed in EFADE [36]. It switches between DE/rand/1 and an advanced mutation strategy with a probability. This probability is adapted, so that DE/rand/1 will be used with a higher chance at the beginning of the search and lower chance at the end.
3.2 Parameter adaptation in Differential Evolution

The parameters used in DE (F, CR and PS) are first dependant on the characteristics of the problem (modality, separability ...) and second dependant on the stage of the search since the properties and distribution of the population will change through the time.

According to the literature, specially in [37], there are two ways of finding the right parameter setting: parameter tuning and parameter control. In parameter tuning, the best assignment of parameters is found by hand. This is ineffective, since it requires a high computational time. On the other hand, in parameter control, the parameters of the DE will change based on some deterministic rule or based on the feedback provided by the running of the algorithm. The different methods to change the values of the parameters on DE can be divided into three subcategories:

- **Deterministic Parameter Control (Deterministic):** Methods under this category modify the parameters based on a deterministic rule. This rule specifies how to change the parameters without any feedback from the search process. The majority of these methods use the time as variable to modify the parameters.

- **Adaptive Parameter Control (Adaptive):** In this case, the methods adapting the parameters make use of the feedback obtained from the different individuals of the population. Then, these updated parameters will be used by all individuals of the population, meaning that global parameters are adapted.

- **Self-Adaptive Parameter Control (Self-adaptive):** Differently from adaptive parameter control, here each individual in the population will have its own parameters, CR and F. Then, these parameters will be encoded with the individuals and they will be updated by some rule or will undergo an evolution process.

Some algorithms attempt to combine the different categories. Combination of Adaptive and Deterministic, and self-Adaptive and Deterministic are normally found in the literature [38, 34].

The majority of the algorithms use methods that fall into the Adaptive Parameter Control subcategory. One of the first adaptive DE algorithms is SaDE [22], proposed in 2005. In SaDE, the F and the CR are generated randomly for each of the individuals following a normal distribution. Moreover, the mean of the CR distribution is adapted using the average of the successful CR values.
during a learning period, while the mean used to generate the F values is not updated.

The idea of using successful F and CR values to update the parameters was commonly adopted afterwards. In 2009, JADE was proposed by Zhang and Sanderson [24]. JADE uses a normal distribution to calculate the new F values and a Cauchy distribution to calculate the new CR values. Differently from SaDE, JADE adapts the mean values of both distributions, using the Lehmer and arithmetic average of the successful instances of F and CR, respectively. Similarly to this work, MDE.pBX was proposed in 2011 [39], that changes the Lehmer mean to a pow mean. Likewise, Wang proposed dn DADE [34], in which the standard deviation of the normal distribution used to calculate the CR is also updated. An improved version of JADE was proposed in [40], in which the standard deviation of the distributions are updated too.

SHADE [25] makes use of a memory to record the successful F and CR values, which is used as means of probability distributions for F and CR generation. L-SHADE [38] was proposed as improvement of SHADE by the same authors in 2014. The new element brought into this algorithm is the linear reduction of the PS with the number of generations. L-SHADE obtained an incredible performance and many variant algorithms based on L-SHADE were proposed afterwards [41, 42, 43, 44].

Some algorithms take into consideration the effect of combinations of F and CR values. In 2011, EPSDE was proposed by Mallipeddi [31]. EPSDE randomly chooses a combination of F and CR values from a pool for each individual of the population. Successful combinations are then stored in a successful pool. When one offspring fails to improve its parent, F and CR values are reinitialized from the initial pool or from the successful pool. A different alternative was proposed in ZEPDE [23], in which the parameter space, for F and CR is divided into 4 regions. Each region has a center point that will be used to define the mean values of distributions.

More recently, HMJCDE [45] was proposed as a combination of a modified version of CoDE and a modified version of JADE. In AGDE [46], random values on the range $[0.1, 1]$ are generated as F. CR is generated following an uniform distribution with two ranges: from 0.05 to 0.15 and from 0.9 to 1. The probability to select the range to use for each individual is calculated using the success rate obtained from the first generation until the current generation. In IDEbd [47] two normal distributions are used to create F and CR respectively. The mean values of the normal distributions are decided based on the fitness ranking of individuals inside the population. Better solutions will use a smaller mean value, while a bigger mean value will be used for the weaker solutions to create F and CR.
3.3 Hybridization between local search methods and Differential Evolution

There are also algorithms that use self-adaptive methods to adapt the different parameters. In 2006, jDE [48] was proposed which assigned F and CR value to each individual in the population. Then these values get evolved in the search process in the sense that better combination of F and CR values have higher chances to enter the next generation by producing stronger trial solutions. In DESAP [49] F, CR and the PS are embedded into the individuals of the population. These three parameters will undergo mutation and then a small perturbation is applied to them. Two different versions of DESAP are called DESAP-rel and DESAP-abs. The difference between them lies in how the PS is embedded into the individuals. In DESAP-rel a relative value is used indicating how much the PS will change, while in DESAP-abs the actual PS is used.

3.3 Hybridization between local search methods and Differential Evolution

One of the biggest challenges of EAs is to keep a good exploration-exploitation rate. If an algorithm has too much exploration, then it will not converge fast enough to find a good solution. On the contrary, if the algorithm exploits too much in problems with many local optima, it will suffer from local optima stagnation problem. Hence a good balance is necessary when combining a LS method of exploitation with DE of global exploration.

This is not an easy task. In the following sections, different strategies that allocate resources to LS and different LS algorithms are reviewed.

3.3.1 Strategies to assign resources to local search

There are different strategies of combining a LS method with DE. The difficulty of this hybridization is to keep a good Local/Global ($\frac{L}{G}$) ratio. Having a high degree of LS will make the solutions, affected by LS, converge faster, but it has a higher chance to get stuck into local optima. On the other hand, if we keep a high degree of global search, the population will not converge fast enough. In order to keep a good $\frac{L}{G}$ ratio, the following three different factors have to be considered [50].

The first factor is related to which individuals will be affected by the LS method. There are algorithms that only apply LS methods to the best individual of the population [51, 27]. These methods avoid to expend unnecessary evaluations in applying LS to a bad solution. On the other hand, there are also methods that apply LS to more individuals. The number of individuals affected
by LS can be fixed, for example, LS is applied to the 20% best individuals in the population [52]. Alternatively, the amount of individuals affected by LS can change from one generation to another, for example, in DEcfbLS [26] only individuals with their fitness above the average will undergo LS. In [53], LS is applied to all the individuals in the population with a probability.

The second factor is related to the frequency in which the LS method will be applied. In the literature, there are two different tendencies: applying LS after every generation such as DECLS [51], DECH [52], DETLS and DEILS [27], or applying LS after a predefined number of generations, as in [26] and [54].

The third factor concerns the length of the LS operator. According to [21], three categories can be defined for the different methods:

- **Fixed length:** In this category, the length of the LS is the same on each case and it does not vary on the entire search. As an example, DECLS [51] defined a length equal to the dimension of the problem divided by 5. Other algorithms have a predefined length equal to 1 [53].

- **Dynamic length:** A deterministic rule defines the length of the LS method, as in [55], in which the length of the LS is larger at the beginning and smaller at the end.

- **Adaptive length:** The length of the LS method will change depending on the feedback from the search. For example, while the new solution is better than the current one, continue the execution of the LS method. [27, 56, 21].

### 3.3.2 Local search methods in DE

As previously mentioned, combining a global search method, as DE, with a LS method can result in significant benefit when searching for the global optimum. In the literature, there are several different LS methods that are combined with DE. According to [21, 50], all these methods can be divided into two categories:

- **Local improvement process (LIP) oriented LS (LLS):** In LLS, local improvements are applied to the individuals inside the population of DE. With these small perturbations, the algorithm manages to find better solutions.

- **Crossover-based LS (XLS):** In XLS, the individuals inside the population are used for combination to produce new offspring. For this reason, these
methods has a high degree of adaptation, since the population is evolving as the search progresses. In other words, more exploration will be performed in early stages of the search, while more exploitation will be performed at the end of the process.

Different approaches from the two different categories of LS methods are reviewed below.

**LLS in DE:** Even though LLS is not the most common practice, there exist some methods in the literature. Chaos Search [57] can be performed to create small perturbations to the solution. DECH [52], proposed by Pei-chong, hybridizes Chaos Search with DE, applying small perturbations to 20% of the population. New solutions will replace the old ones if they obtain a better performance. Likewise, chaotic LS was applied to the best individual in [51].

A different approach was proposed by Poikolainen [26], in which the search is performed in each single dimension instead of multiple dimensions at the same time. The mechanism is similar to the Hooke-Jeeves algorithm [58], in which a small perturbation, \( p \), is applied to the first dimension in the negative direction. If the new solution is not better, then a small perturbation \( p/2 \) is applied to the opposite direction. After going through all the dimensions, the final offspring is compared with the starting solution. If it is better, the search will start again from the new point. If not, the perturbation will be halved and the search start again. This procedure continues for 40 iterations.

**XLS in DE:** One of the earliest memetic DE algorithms is DEfirSPX proposed by Noman and Iba [59], in which simplex search [60] is applied to the best individual in the population for a fix number of iterations. Simplex makes use of different individuals from the population, making the LS method belong to XLS. An improved version, DEEachSPX [21], was proposed in 2008, in which a hill climbing process is used to determine the search length, i.e., the search continues while the new individual is better than the parent.

A different alternative have been handled by Ali et al. [27], with the proposal of two different LS algorithms: Trigonometric LS (TLS) and Interpolated LS (ILS). With the help of three random individuals from the population, TLS will create an offspring as the center of a hypergeometric triangle. This center point will move with the usage of the fitness of the three solutions, such that the center will be closer to the solution with better fitness. This solution is based on the Trigonometric Mutation Operator, presented by Fan [61]. The second LS method, ILS, also takes three random individuals from the population. Then, a quadratic curse is fitted into the three points and the minimum of this curse is taken as the new offspring.

An Orthogonal design (OD) has been used, by Dai et al., to create a DE with
Orthogonal LS (OLSDE) [62]. OLSDE takes two random solutions to create different offspring. In order to create the offspring, the different parameters of a solution are divided into different groups, trying to avoid grouping parameters with large relative range into the same group. Then, the worst individual from the population is replaced by the best offspring. A similar method was addressed by Peng, called Taguchi LS (TLS) [63]. In this work three points are needed: two randomly selected from the population and the middle point between them. Then the individuals are divided into 4 parts, and 9 new offspring are created as a combination of the different parts of the different individuals. Then, for every part of each offspring the importance is measured. A final individual is created by taking each part of the offspring with higher score.
Chapter 4

Research Questions and Contributions

In this chapter, first the research questions (RQs) related to the main DE problems are described in Section 4.1. Second, the research contributions (RCs) made to deal with the RQs are outlined in Section 4.2. Finally, an overview of the papers supporting this thesis is given in Section 4.3.

4.1 Research Questions

Given the weakness and drawbacks of DE stated in the previous chapter, the following three RQs are identified.

RQ1: How to adapt the selection between different mutation strategies, so that DE can be adapted to different stages and search spaces considering the individuals of the population?

During the past two decades, a diversity of mutation strategies have been proposed with a variety of properties, making difficult to select the right one for a specific problem. Moreover, the right mutation strategy does not only depend on the specific problem. Firstly, the current state of the search will affect the selection of the mutation strategy. At the start of the search, it is always better to more explore the search space, while exploiting promising regions should be more preferred at the later stages of the search. Secondly, it is obvious that better individuals are more difficult to improve than worse ones and hence they may need to receive different mutation operators. For these two reasons, the adaptation of different mutation strategies with consideration of solution fitness is crucial.
for further advancing the DE performance.

**RQ2:** How online adaptation of the control parameters of DE can be implemented to make the search more efficient for different problems and in different regions of the space?

As previously mentioned, researchers have been focusing on adapting two parameters of DE (F and CR). These approaches utilize successful values of F and CR in order to update the center points of different probability functions. By using all the successful values, these methods ignore the fact that stochastic operators (mutation and crossover) can make bad parameter values also get success by chance. This imprecise judgment of ”success” would prevent the DE parameters from being adapted towards the most effective values.

**RQ3:** How local search can be conducted inside the global search process of DE to enhance its performance?

When combining the global search method as DE and a LS method, the most important question is to which degree of exploitation the LS has to be performed. Too much exploitation can cause the search to get stuck into a local optimum. On the contrary, if the LS method explores too much, the global search by DE will not be able to converge to a high quality solution given limited computing resources. For this reason, a proper balance between exploration and exploitation has to be maintained for a LS method to be used in a memetic DE algorithm.

### 4.2 Research Contribution

To tackle the aforementioned RQs, the following RCs have been made within the thesis project.

**RC1:** The first RC deals with RQ1 on how to adapt between different mutation strategies. A technique that adapts the selection between different mutation strategies is proposed in PAPER A. This technique, called Rank-based Adaptation of Mutation strategies (RAM), will adapt the selection from multiple mutation strategies by updating selection probabilities based on their success rates. So far, other methods previously presented (i.e. SaDE [30]) also use the success rates in order to adapt between the different mutation strategies. The novelty of RAM is that the population is ranked according to the fitness values of the individuals. Then, the population will be divided into different groups. Each group
has its own probability per mutation strategy based on the knowledge gained during the previous learning periods. During a learning period, the success rates of the different mutation strategies on each group are calculated and lately used to adapt the different selection probabilities. As shown in PAPER A, RAM can be used with any number of mutation strategies as well as it can be combined with any DE algorithm already established.

RC2: The second RC deals with RQ2. The method is presented in PAPER B and called Greedy Adaptive Differential Evolution (GADE). It adapts the parameter assignment for F and CR using a local greedy search approach. The key idea is to generate two new candidates in the neighborhood of the current (parameter) assignment, and all the three candidates are then evaluated during a learning period. After the learning period, the best candidate is taken as the current assignment for F and CR, respectively, and this procedure is repeated. GADE has two desirable properties. First, assessment of candidates is made more reliable since each candidate is applied a number of times in the learning period. Second, only a short learning period is needed due to the low number of candidates to evaluate prior to parameter updating. This makes GADE particularly attractive in many real world applications in which the total number of fitness evaluations allowable in algorithm execution is not sufficiently high. Further, GADE has been successfully applied to solve the industrial problem of optimal harmonic filter design for power transmission. (PAPER C).

RC3 The third RC provides a method that will adapt the two control parameters of DE (F and CR) in a joint manner. This method, called Joint Adaptation of Parameters in Differential Evolution (JAPDE), is proposed in PAPER D. Like other adaptive DE algorithms (e.g. [30, 24]), JAPDE also generates F and CR values using Cauchy and Normal distributions, respectively. The novelty in JAPDE is that it considers the pairs of mean values of the two distributions (for F and CR), and each pair has a chance to be selected. The adaptation of the control parameters is thereby implemented by manipulating a matrix of probabilities for a complete set of combinations of the two mean values. Successful pairs will receive high likelihoods of selection after probability updating through many learning periods, while an exploration plane is used to avoid fast convergence. Moreover, JAPDE has been combined with the proposed mutation adaptation method RAM, giving rise to the RAM-JAPDE algorithm, which adapts the selection probabilities for mutation strategies and control parameter pairs at the same time.
RC4: The fourth RC lies in the suggestion of Eager Random Search (ERS) that is hybridized with DE (PAPER E). ERS belongs to the LLS category, in which local improvements are made to the individuals of the population. More precisely, ERS is a family of three stochastic LS schemes. Each scheme uses a different probability distribution in order to create random perturbations to the best solution of the population. Due to the nature of the different probability functions, different behaviours of exploration/exploitation are enabled with the LS. The three LS schemes in ERS are summarized below:

- **Random Local Search (RLS)** makes use of a Uniform distribution in order to perform more exploration of the search space.

- **Cauchy Local Search (CLS)** uses a Cauchy distribution, which will exploit promising regions with high probability, while still having a small probability to explore different regions of the search space.

- **Normal Local Search (NLS)** applies small perturbations using a Normal distribution, exploiting promising regions of the search space.

RC5: The fifth RC is the proposal of a new memetic framework for enhancing DE algorithms by integration with Alopex local search (ALS). It is addressed in PAPER F and referred to as MFDEALS. ALS uses the individuals from the population to produce new ones, so it belongs to the XLS category of LS methods in MAs. At the early stage when the individuals are widely distributed, ALS will explore broader regions. Later, as the search is progressed to focus on some promising regions, ALS will become more exploitative in testing those regions in order to find even better solutions. As the behavior of ALS is controlled by the status (population diversity and search stage) of the global search by DE, we achieve an automatic adaption of the LS characteristics in favour of more cooperative behaviour between exploration and exploitation. Besides, a computing resource allocation strategy is developed in MFDEALS to decide when, where, how often and how intensive to apply ALS inside the global search procedure of DE.

An overview of how the RCs affect the different operations of DE, or how the RCs add a new operation into DE is given in Fig. 4.1. Additionally, the relation between the papers and the RQs is given in table 4.1.
4.3 Overview of the Papers

In this subsection, the 6 included paper and my contributions are presented below:

![Differential Evolution and Research Contributions](image)

Figure 4.1: Overview of the relation between the Research Contributions and DE. G represent the maximum number of generations

<table>
<thead>
<tr>
<th>Papers</th>
<th>RQ1</th>
<th>RQ2</th>
<th>RQ3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAPER A</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PAPER B</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PAPER C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAPER D</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>PAPER E</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>PAPER F</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Table 4.1: Relation between included papers and research questions.

4.3 Overview of the Papers

In this subsection, the 6 included paper and my contributions are presented below:

**Abstract**: Differential evolution has many mutation strategies which are problem dependent. Some Adaptive Differential Evolution techniques have been proposed tackling this problem. But therein all individuals are treated equally without taking into account how good these solutions are. In this paper, a new method called Ranked-based Mutation Adaptation (RAM) is proposed, which takes into consideration the ranking of an individual in the whole population. This method will assign different probabilities of choosing different mutation strategies to different groups in which the population is divided. RAM has been integrated into several well-known adaptive differential evolution algorithms and its performance has been tested on the benchmark suit proposed in CEC2014. The experimental results shows the use of RAM can produce generally better quality solutions than the original adaptive algorithms.

**Status**: Published.

**My Contribution**: I am the main author of the paper contributing with the idea, the algorithm, the implementation and the experiments. Also, I wrote the major part of the paper.


**Abstract**: Differential evolution (DE) presents a class of evolutionary and meta-heuristic techniques that have been applied successfully to solve many real-world problems. However, the performance of DE is significantly influenced by its control parameters such as scaling factor and crossover probability. This paper proposes a new adaptive DE algorithm by greedy adjustment of the control parameters during the running of DE. The basic idea is to perform greedy search for better parameter assignments in successive learning periods in the whole evolutionary process. Within each learning period, the current parameter assignment and its neighboring assignments are tested (used) in a number of times to acquire a reliable assessment of their suitability in the stochastic environment with DE operations. Subsequently the current assignment is updated with the
best candidate identified from the neighborhood and the search then moves on to the next learning period. This greedy parameter adjustment method has been incorporated into basic DE, leading to a new DE algorithm termed as Greedy Adaptive Differential Evolution (GADE). GADE has been tested on 25 benchmark functions in comparison with five other DE variants. The results of evaluation demonstrate that GADE is strongly competitive: it obtained the best rank among the counterparts in terms of the summation of relative errors across the benchmark functions with a high dimensionality.

**Status:** Published.

**My Contribution:** I am the main author of the paper contributing with the idea, the algorithm, the implementation and the experiments. Also, I wrote the major part of the paper.


  **abstract:** Harmonic filtering has been widely applied to reduce harmonic distortion in power distribution systems. This paper investigates a new method of exploiting Differential Evolution (DE) to support the optimal design of harmonic filters. DE is a class of stochastic and population-based optimization algorithms that are expected to have stronger global ability than trajectory-based optimization techniques in locating the best component sizes for filters. However, the performance of DE is largely affected by its two control parameters: scaling factor and crossover rate, which are problem dependent. How to decide appropriate setting for these two parameters presents a practical difficulty in real applications. Greedy Adaptive Differential Evolution (GADE) algorithm is suggested in the paper as a more convenient and effective means to automatically optimize filter designs. GADE is attractive in that it does not require proper setting of the scaling factor and crossover rate prior to the running of the program. Instead it enables dynamic adjustment of the DE parameters during the course of search for performance improvement. The results of tests on several problem examples have demonstrated that the use of GADE leads to the discovery of better filter circuits facilitating less harmonic distortion.
than the basic DE method.

**Status:** Published.

**My Contribution:** I am the main author of the paper contributing with the idea, the algorithm, the implementation and the experiments. Also, I wrote the major part of the paper.


**Abstract:** Differential evolution (DE) is a population-based metaheuristic algorithm that has been proved powerful in solving a wide range of real-parameter optimization tasks. However, the selection of the mutation strategy and control parameters in DE are problem dependent, and inappropriate specification of them will lead to poor performance of the algorithm such as slow convergence and early stagnation with a local optimum. This paper proposes a new method termed as Joint Adaptation of Parameters in DE (JAPDE). The key idea lies in dynamically updating the selection probabilities for a complete set of pairs of parameter generating functions based on feedback information acquired during the search by DE. Further, the Rank-Based Adaptation (RAM) method is also utilized here to facilitate the learning of multiple probability distributions for mutation selection, each of which corresponds to an interval of fitness ranks of individuals. The coupling of RAM with JAPDE results in the new RAM-JAPDE algorithm that enables simultaneous adaptation of the selection probabilities for pairs of control parameters and mutation strategies in DE. The merit of RAM-JAPDE has been evaluated on the benchmark test suit proposed in CEC2014 in comparison to many well-known DE algorithms. The results of experiments demonstrate that the proposed RAM-JAPDE algorithm outperforms or is competitive to the other related DE variants that perform mutation strategy and control parameter adaptation respectively.

**Status:** Submitted.

**My Contribution:** I am the main author of the paper contributing with the idea, the algorithm, the implementation and the experiments. Also, I wrote the major part of the paper.


**abstract:** Differential evolution (DE) presents a class of evolutionary computing techniques that appear effective to handle real parameter optimization tasks in many practical applications. However, the performance of DE is not always perfect to ensure fast convergence to the global optimum. It can easily get stagnation resulting in low precision of acquired results or even failure. This paper proposes a new memetic DE algorithm by incorporating Eager Random Search (ERS) to enhance the performance of a basic DE algorithm. ERS is a local search method that is eager to replace the current solution by a better candidate in the neighborhood. Three concrete local search strategies for ERS are further introduced and discussed, leading to variants of the proposed memetic DE algorithm. In addition, only a small subset of randomly selected variables is used in each step of the local search for randomly deciding the next trial solution. The results of tests on a set of benchmark problems have demonstrated that the hybridization of DE with Eager Random Search can substantially augment DE algorithms to find better or more precise solutions while not requiring extra computing resources.

**Status:** Published.

**My Contribution:** I am the main author of the paper contributing with the idea, the algorithm, the implementation and the experiments. Also, I wrote the major part of the paper.


**abstract:** Differential Evolution (DE) represents a class of population-based optimization techniques that uses differences of vectors to search for optimal solutions in the search space. However, promising solutions/regions are not adequately exploited by a traditional DE algorithm. Memetic computing has been popular in recent years to enhance the exploitation of global algorithms via incorporation of local search. This paper proposes a new memetic framework to enhance DE algorithms using Alopex Local Search (MFDEALS).
The novelty of the proposed MFDEALS framework lies in that the behavior of exploitation (by Alopex local search) can be controlled based on the DE global exploration status (population diversity and search stage). Additionally, an adaptive parameter inside the Alopex local search enables smooth transition of its behavior from exploratory to exploitative during the search process. A study of the important components of MFDEALS shows that there is a synergy between them. MFDEALS has been integrated with both the canonical DE method and the adaptive DE algorithm L-SHADE, leading to the MDEALS and ML-SHADEALS algorithms respectively. Both algorithms were tested on the benchmark functions from the IEEE CEC’2014 Conference. The experiment results show that MDEALS not only improves the original DE algorithm but also outperforms other memetic DE algorithms by obtaining better quality solutions. Further, the comparison between ML-SHADEALS and L-SHADE demonstrates that applying the MFDEALS framework with Alopex local search can significantly enhance the performance of L-SHADE.

**Status:** Published.

**My Contribution:** I am the main author of the paper contributing with the idea, the algorithm, the implementation and the experiments. Also, I wrote the major part of the paper.
Chapter 5

Research Methodology

At first, two industrial partners, within the EMOPAC project, proposed different optimization problems, for which there were no specific automatic solutions other than manual tuning of the parameters.

The research started with the study of different optimization techniques, finding out that DE was the most suitable alternative for real-parameter optimization.

Then, a study of the state-of-art DE algorithms was carried out. I identified the following three RQs:

**RQ1:** How to adapt the selection between different mutation strategies, so that DE can be adapted to different stages and search spaces considering the individuals of the population?

**RQ2:** How an online adaptation of the control parameters of DE can be implemented to make the search more efficient for different problems and in different regions of the space?

**RQ3:** How local search can be conducted inside the global search process of DE to enhance its performance?

I believe that solving the above three RQs can help overcoming the two problems: low convergence speed and local optima stagnation, which are often encountered when DE is used in real applications.

To achieve my research goals, I followed a deductive-like research method. The following 6 steps were followed:

1. Different hypotheses were made:
• First, for different individuals of the population, it can be better to use distinct policy to adapt their mutation strategies.

• Second, by greedy adjustment of the two control parameters in the neighborhood, it is possible to dynamically adapt the parameters of DE to fit different problems and search stages.

• Third, by jointly modifying the two control parameters of DE based on a complete coverage of their combinations, we can more effectively solve a wide range of complex optimization problems.

• Fourth, incorporating LS into DE will enhance the ability of the algorithm to exploit promising regions leading to the discovery of better solutions if a balance of exploration and exploitation is properly maintained.

2. Different methods were proposed for each of the above hypotheses.

3. The methods were tested and validated on different benchmarks predefined by the community.

4. If a proposed solution was desirable and/or outperformed the state-of-art DE algorithms, then the results were published and the solution could be used on the optimization problems defined by the industrial partners.

5. On the contrary, if the solution was not better than its counterparts, then two options were available:

   (a) Tune the current solution to improve its performance

   (b) Propose a complete new solution.

After selecting either option (a) or (b), the process is repeated from step 3.

An overview of the followed steps is shown in Figure 5.1.
Figure 5.1: Overview of the followed research methodology.
Chapter 6

Conclusion and Future Work

In this thesis, different methods are proposed to improve DE for continuous optimization. Firstly, a new method that adapts between different mutation strategies, called Ranked-based Adaptation of Mutation strategies (RAM), has been proposed. RAM does not treat each individual equally and group them considering their fitness values. These groups will have different probabilities for the different mutation strategies. Secondly, two different methods for adaptation of control parameters, called Joint adaptation of parameters in DE (JAPDE) and Greedy Adaptive Differential Evolution (GADE), have been proposed. The former will adapt the mean values of the distributions, used to generate \( F \)'s and \( CR \)'s, in a joint manner by maintaining and updating a probability matrix for different combinations. Additionally, JAPDE has been combined together with RAM to enhance its performance. On the other hand, the latter will independently adapt the mean values of probability distributions for generation of \( F \)'s and \( CR \)'s respectively by studying the performance of two neighbouring candidates. Thirdly, two different LS methods have been proposed: Eager Random search (ERS) and Memetic Framework in Differential Evolution with Alopex Local Search (MFDEALS). The former contains a family of three different LS operators which have different exploration/exploitation behaviours. ERS belongs to the LLS category of memetic algorithms, which performs perturbations to the best individual of the population for local improvement. The latter employs ALS combined with DE, which achieves an automatic transition from more exploration in early stages of the search to more exploitative behaviour in later stages. This memetic framework belongs to the XLS category of memetic algorithms, since ALS is a method that uses individuals from the population to create potentially better solutions.

These methods have been tested on different benchmark problems in comparison with many state-of-the-art algorithms. The important conclusions that
can be drawn from this thesis are summarized in the following:

- RAM has been tested in combination with 4 well-known adaptive DE algorithms which use different mutation strategies at the same time. It was shown that incorporating RAM into the different algorithms could improve their performance. For this reason, it can be concluded that considering fitness of individuals when selecting mutation strategies will bring benefits to advance DE methods. Additionally, RAM has been combined with JAPDE, significantly improving its performance.

- Greedy search can be used for parameter adaptation in DE to find better assignments for F and CR respectively in the neighborhood of the current candidates.

- Since $F$ and $CR$ are related to each other, jointly considering them can obtain even better results than handling $F$ and $CR$ independently. The combined algorithm RAM-JAPDE developed in this thesis outperforms other well-known adaptive DE algorithms.

- Applying an adaptive DE method in real world scenarios, as designing optimal harmonic filters in power systems, is beneficial for not only improving DE performance but also saving time of engineers in tuning the parameters.

- The usage of a LS method combined with DE has been proven to be beneficial to its performance. However, this enhancement of the performance depends on the exploration/exploitation behaviour of the used LS. An exploratory LS could be beneficial in multimodal scenarios yet detrimental in unimodal problems.

- A LS method that changes its exploration/exploitation behaviour during the search process is beneficial for the performance of DE, since it can more explore the search space at early stages so as to avoid the problem of early convergence. Then, at later stages of the search, it is good to more exploit the promising regions of the search space in order to fine tune the best solutions. Additionally, it has been proven important to distribute the resources to DE and LS in a smart way, since expending to many resources on either exploration or exploitation will deteriorate the quality of the final solutions.

This thesis does not investigate all the possible ways to enhance DE. For example, the adaptation of the PS has not been studied, nor the possibility of
running a multi-population DE. These alternatives can be interesting directions of future research to further improve the DE performance [38, 64]. Additionally, the proposed adaptive and memetic DE algorithms will be applied and evaluated in a number of industrial scenarios.
Bibliography


