FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

NEWEPSO - New developments and testing of EPSO and DEEPSO

João Pedro Antunes Vigo

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Professor Doutor Vladimiro Henrique Barrosa Pinto de Miranda

Second Supervisor: Doutor Leonel de Magalhães Carvalho

July 25, 2016

Resumo

Ao longo das últimas décadas foram sendo desenvolvidos enúmeros modelos matemáticos de resolução de problemas, de maior ou menor complexidade. A *computação evolucionária* surge assim com grande destaque dada a capacidade demonstrada face a estes desafios. Baseado no modelo geral de partículas (PSO, *Particle Swarm Optimization)*, o algoritmo do meta-heurística EPSO, *Evolutionary Particle Swarm Optimization* surge em 2002 no INESC Porto [1]. Todos os métodos populacionais apresentam grandes vantagens sendo evidente a total independência no que toca ao tamanho do próprio sistema a ser analisado, destacando assim a transversalidade na resolução de diversos problema e claro, de sistemas de energia [2].

O EPSO apresenta-se como um modelo com características diferentes dos anteriores, a capacidade de auto-adaptação, as regras de comunicação e outros aspetos são alguns dos pontos a ter em consideração neste trabalho[3][4].

Nesta tese, o principal objetivo é de facto apresentar variantes ao EPSO atualmente conhecido e se possível, melhorar a performance e a eficiêcia deste algoritmo. A constante busca por melhorar os modelos existentes, formando algoritmos cada vez mais eficazes e computacionalmente mais leves formam a principal motivação deste trabalho. Paralelamente a isto, para avaliarmos o comportamento das variantes propostas, torna-se necessário analisar a forma como o programa reage em funções de otimização e em problemas reais de sistemas elétricos de energia.

Palavras-chave: Computação, eficiência, EPSO, algoritmos evolucionários, variante.

Abstract

Several mathematic models linked to the resolution of lower or higher complexity problems have been developed over the course of the last decades. *Evolutionary computation* arises therefore with special emphasis due to its proven ability to address these challenges. Based on the General Particles Model (*PSO – Particle Swarm Optimization*), the meta-heuristic algorithm EPSO (*Evolutionary Particle Swarm Optimization*) appeared in 2002 at INESC Porto [1]. Each population method presents its own particular advantage over classical mathematical programming methods, especially concerning the ability to proceed in a parallel search and discovering the optimum in ill-shaped problems, but all face a "curse of computation effort". This explains why the research for greater efficiency in conceiving and implementing meta-heuristics is an on-going effort worldwide [2].

EPSO benefits from properties existing in several methods, namely evolutionary Programming and PSO. Its very particular characteristics such as its auto-adaptation ability, its communication rules, among others which will be considered in this paper, led the EPSO to stand out from other models [3][4].

This thesis aims to present a current EPSO variant and, if possible, improve both the performance and efficiency of this algorithm by designing new variants. The constant search for improving the existing models, therefore creating algorithms that are not only more efficient but also computationally lighter, stand as the main purpose of this work. In order to evaluate the reaction of the proposed variants, it became necessary to analyze the algorithm's reaction to optimization functions with no restrictions, as well as to electrical energy system problems where constraints to the domain are effective.

The results obtained were extremely successful and resulted in clear net speeding up of the EPSO algorithm.

Palavras-chave: Computation, efficiency, EPSO, evolutionary algorithms, variant.

Acknowledgments

Many consider the easiest part and the most uninteresting part of dissertations. For me, this section is if not the most difficult, not far from that. A page does not serve to weave all thanks and all the feelings I felt during this period of my life.

First of all, a deep thanks and recognition to Dr. Prof. Vladimiro. More than a great teacher, is a person so exciting to know about where all of his experiences in life turn into knowledge, to science. Thank you for believing in me, it is with deep pride that I can say that Dr. Prof. Vladimiro Miranda was my mentor. Thank you for your guidance, the constant challenges made and the inspiration you are to me. I hope not to have disappointed you. A special thanks also to Dr. Leonel Carvalho, the help that you gave me was priceless. Whenever needed, there was a ready door to help me.

To whole INESC Porto, a big thank you. It is without any doubt an environment that breathes science, where it history is build and future is prepared.

A big thanks to all my friends, some of you I have to nominate. Thanks Bruna Tavares for the friendship, the work and the help that you always gave me in this dissertation. Thanks to my friends, José Rodrigues, Válter Rocha, Tânia Tavares, Gil Almeida and João Ribeiro with who, I lived and shared this huge experience. To my old friends, Mosqueteiros, who helped me to take the brain out of work when it was necessary, a huge thanks!

To my family and my dog Spy, thank you does not serve. Were, are and will be my refuge, everything I am I owe to you!

To my college mate, students association, friendship and partner of my life, my girlfriend Ana Mota, a eternal thanks to you! Thanks for the confidences, the calm you gave to me, for being the one who could tell the progress and return of all this work, for the force that always gave me. I will be forever grateful, you know me like no one else and that I will never forget.

Finally, I want to dedicate this work to my dear grandmother, Domingas Martins, who gave her soul to God during this time. This is for you grandma!

João Vigo

"Insanity: doing the same thing over and over again and expecting different results."

Albert Einstein

Contents

List of Figures

List of Tables

List of Acronyms and Symbols

DER	Distributed Energy Resources
ED	Differential Evolution
EE	Evolutionary Strategies
EP	Evolutionary Programming
EPS	Electric Power System
EPSO	Evolutionary Particle Swarm Optimization
GA	Genetic Algorithms
PSO	Particle Swarm Optimization
WWW	*World Wide Web*
δ	Angle phase
U	Voltage

Chapter 1

Introduction

The present chapter presents a succinct contextualization of the topic addressed in this thesis. The methodology, motivations and the way in which evolutionary programming has been developing throughout the last decades are some of the topics will be covered. Both its purpose as well as its structure will be detailed further the following sections.

1.1 Evolutionary Computation and optimization

In an evolutionary computation perspective, optimization may be defined as a parallel search for the solution of a problem.

Understandably there are several types of problems. When dealing with complex and non-linear characteristics, traditional problem-solving models face serious convergence challenges. Models such as the gradient model are example of processes which present serious barriers to convergence within the context of optimization [5] and may get caught in local optimum areas.

Several approaches of optimization models with the ability to overcome the aforementioned problems have arisen over the course of the last decades and up until today. Also known as meta-heuristic methods, these are inspired in natural phenomena from the observation of the behaviour of insect swarms, bird flocks and fish shoals. Each individual behaviour is directly influenced by the simultaneous collective behaviour of the group. A new strand of "computational intelligence" has recently emerged.

Several current strategies have been adopted in such optimization processes, regardless of their type. Evolutionary algorithm models proved to be an efficient tool, not only with much potential but also with a high development capacity, so that its appearance in the resolution of actual problems with great complexity, can be easily understood. Taking this into consideration, evolutionary computation consists in a stochastic optimization technique, where the biology of the natural evolution of species' is used as a context to optimization. These models are iterative, since they allow the solving of several optimization problems, not guaranteeing, however, the convergence for the best solutions solutions except in a probabilistic manner, because most of them depend of parameters that are randomly sampled during the process. The evaluation of the

quality of these algorithms is therefore done taking the average quality of solutions obtained in a number repeated runs.

The successive iterations associated to these models allow the demonstration of their high adjustment capacity as well as their adaptation to a series of single or multi-target problems, including a set of solutions which is successively evaluated and analyzed. Of the several types of evolutionary algorithms currently known, generic algorithms are the most commonly used, as will be explained further in this paper.

1.2 Objective of the Thesis

Over the course of the last years, the progress in the computational area invites us to be part of that same evolution, and breaking the currently existing barriers is a tempting goal. Therefore, this thesis intention remains the presentation of new variants to *EPSO - Evolutionary Particle Swarm Optimization* - evolutionary model.

The search for a more robust, faster and a computationally lighter method is definitely a barrier which is difficult to break. Nonetheless, there are new ideas that can be put into practice, tested and compared to the original model, resulting in the drawing of conclusions which will be detailed later in this thesis.

Considering the constant increase in complexity of the aforementioned problems, the search for equally more complex algorithms implies higher computational efforts, arising therefore the challenge of optimizing the already existing models. In order to achieve this objective, several variants of the original EPSO algorithm [6] will be implemented in the C++ language, registering the program's reaction to the changes.

1.3 Structure of the Thesis

This section intends to briefly contextualize the addressed topic, detailing its essential goals. Besides this introductory part, this text compasses 5 other chapters.

Chapter 2 describes not only the current state-of-art, but also provides a more detailed explanation of the EPSO model-associated concepts used in this thesis and historical aspects behind its formulation.

In Chapter 3, the first EPSO variant is presented. Named "VAREPSO", this remains as the first proposal of the current work. Starting with a short theoretic presentation, it goes through several optimization tests and evaluation of its performance compared with EPSO original.

In Chapter 4, the second variant version of EPSO, "SUBEPSO", is explained with its theory and results are presented with associated computation efforts.

In Chapter 5, a real world problem application is presented. Using EPSO variants, an electrical energy system is solved. Using EPSO strategies, performances from original and variant versions are compared.

Finally, Chapter 6 conducts the main conclusions of this work where some suggestions for future work are also described.

Chapter 2

State of the art

This chapter compasses a review of the state of art about all the investigation work on the theory of evolutionary algorithms that been developed until this moment. This analysis requires a specific level of knowledge, and is framed within concepts such as power systems, evolutionary algorithms and programming.

There are several papers and publications on this subject. This thesis intends to guide us through the main aspects of evolutionary algorithms' systems.

Firstly, a historical framework on evolutionary programming will be put forward, including the timeline and the development that have taken place overtime. Later, an analysis of the different approaches used in order to reach the Evolutionary Algorithm named EPSO will be presented.

This set of algorithms include evolutionary methods that borrow the movement rule from PSO and which use a recombination operation that evolved under selection pressure. This optimization algorithm can be attributed to multiple problems resolution, such as power systems. EPSO has already demonstrated great results as far as accuracy, efficiency and robustness [7]. A short description of the aforementioned problem will be provided in this chapter.

2.1 Historical perspective of Evolutionary Computation

Although the first notions on evolutionary programming date to the 1950s, the term *Evolutionary Computation* has only been officially used to describe the process later in the 1990s, when the first works on algorithms for calculating a given input-output function come about.

Over the course of time, the emergence of visionary minds stimulated the development of multiple classical methods on computational evolution, which have marked the 1960s and have created three general starting points out of which algorithms are currently developed [2].

As aforementioned, three distinct types of algorithms have been developed, namely Evolutionary Programming (EP), Evolutionary Strategies (EE), and Genetic Algorithms (GA). All of those stated previously share the base ideology of adapting the natural evolution of biology as the key for solving several types of problems or paradigms.

The first (EP), as described in [8], was developed at the *Technical University of Berlin*, Germany by three students namely P. Biener, I. Rechenber and H. P. Schwefel. At the same time, across the Atlantic, Lawrence Fogel in San Diego, California took the first steps on programmatic evolution (EP). In that same country at the *University of Michigan Ann Arbor*, John Holland led the development of a Genetic algorithm (GA).

All of the previously mentioned branches applied the concept of collective learning in a process that comprised a population of particles/individuals, and where all evolutionary methods faced challenges in attempting to find the best global solution, therefore basing themselves in the idea of the existence of a *fitness function*. From this function would result a quality measure, which enabled the sorting of the several particles of the population according to a quality index which would later be used at the selection process.

The 25 years following the 1960s have been marked by the separation of each of the nuances in computational evolution. Each of them developed independently so that the international community of researchers faced the require of scheduling periodical meetings and programming conferences. The international workshop *Parallel Problem Solving from Nature*, Dortmund 1991, which came about as the first of its kind, was followed by several others, reaching a point where the *Evolutionary Computation* term became a general definition compassing each and every evolutionary computation area.

2.2 Introduction to Evolutionary Algorithms and to the EPSO

Inspired in biology and the observation of nature, many algorithms have been proposed, the most important family being the Evolutionary Computation algorithms. The *Particle Swarm Optimization, PSO* is another example, as it was inspired by the collective movement of bird flocks or bee swarms [4] and will be approached in greater detail in the following sections.

The relationship found between biology and mathematics/programing is obviously a source of both curiosity and a skepticism which is experienced by the general public. With the intention of combining theories for the formation of a theoretically more powerful model, and specifically in the case being considered, several attempts were made of hybridizing evolutionary process with PSO.

In this context, the formulation of EPSO is one of the most successful. EPSO can be considered a process which, when in comparison with other meta-heuristic models [1][2][9], has proven its superiority straight from the moment of its creation by presenting overall better results and by showing that it could become a method with serious abilities to solve problems with complex optimization issues. In past studies and benchmarking, EPSO has in general revealed to lead to better results than alternative methods [10][11][12][13].

2.3 Particle Swarm Optimization (PSO), the swarm's intelligence

As aforementioned, swarm intelligence comprises a computational technique inspired in the collective behaviour of a group of individuals who act in a coordinated and organized fashion, fostering the information exchange among each of them. As such, the PSO model (Particle Swarm Optimization), created in 1995 by James Kennedy (Social Psychologist) and Russel Eber- hart (Electric Engineer), comes about precisely through behaviour observation of flocks of birds when trying to feed [14].

This is, an optimization model based on a process in which, in a given population, a set of particles is created - or solutions to the problem - which are put in the research space defined by a problem or function. These points are evaluated by an objective function value that translates the best or worst fitness, therefore differentiating each particle in the position it occupies at any given moment.

Each individual is defined by a movement rule that encompasses not only a position vector, but also a speed vector and a memory vector, which saves its past positions. This leads to a process that combines historical positions, the current position and the point where the best *fitness* was registered.

From one iteration to the other, new individuals are generated based on the movement rule, which will be further explained later in this thesis. In a new iteration, each and every particle of the population suffers a specific movement, as if it were a swarm, searching the zone of the dimension space with the best *fitness*.

The set of particles, also known as a swarm, can be seen as a tool with great potential for solving diverse problems, since one single particle has far less exploratory power. On that account, the interaction in a coordinated set of individuals - which communicate among themselves therefore building a data exchange network – is absolutely fundamental.

2.4 Evolutionary Particle Swarm Optimization (EPSO), a new paradigm

Inspired out of the original PSO model, EPSO did since its beginning present a set of solutions for each iteration, also known as particles. From an iteration to the other, each particle X_i moves according to the "movement rule". This rule consists of a formula including several components which define the next position of a given particle. Both the convergence properties and the functioning of algorithms are, therefore, ensured by the movement equation and the cooperation that results from a joint action of 1) the auto-learning process related to the best progression mode towards the optimum, and 2) the cooperation factor. EPSO thus assures the exchange of information among particles when these move in the research space.

ESPO borrow the *particle movement process* from PSO, where each particle is seen as a potential solution for a given optimization problem. In this model each particle groups a set of vectors that define its **position**, more specifically the position vector X_i, the **speed** vector V_i, and the vector for the **best position occupied by the particle up until that exact moment,** b_i. A fourth term

is then added to EPSO, symbolizing the **cooperation**, in other words, the best position occupied by the total set of particles of the swarm, which is memorized in the vector b_G. Put in a concise way, the three main terms featuring the PSO, in combination with the cooperation factor, result in a **movement rule** that determines the new position of each particle of the solution swarm:

$$\mathbf{X}_i^{new} = \mathbf{X}_i + \mathbf{V}_i^{new} \tag{2.1}$$

Where V_i can be defined as the speed of the particle X_i and is calculated as follows:

$$\mathbf{V}_i^{new} = W_{in_i}.\mathbf{V}_i + Rnd().W_{m_i}(\mathbf{b}_i - \mathbf{X}_i) + Rnd().W_{e_i}(\mathbf{b}_G - \mathbf{X}_i)\mathbf{P} \tag{2.2}$$

As previously stated, equation 2.2 compasses the several aforementioned factors, featuring the inertia as its first term, which represents the fact that the particle keeps its movement on the same direction as presented before. Memory stands out as the second term, defined by the presence of the vector with the best *fitness* position that was reached up until that moment. Thirdly, cooperation factor stimulates the swarm's information exchange, attracting particles to the best point reached by the whole *swarm*.

Special note to P term, associated to the passing of information within the *swarm*, with a diagonal matrix affecting all individual dimensions and which contains binary variables with value 1 and probability p, and 0 with probability *1-p* [4][10].

The equation 2.2 presents a set of **W** parameters which allow the occurrence of **mutation** processes within these strategic parameters. In other words, these consist of diagonal matrixes whose weights are set at the beginning of the process and where *in* index stand for the inertia, index *m* for memory weight and index *c* for the weight of the cooperation in a determined posterior position. Rnd() stands for several random numbers belonging to an uniform distribution of [0,1].

At this stage, the evolutionary perspective comes into the equation. Then it's able to consider the adaptation capacity of a population (swarm) and of individuals (particles), as well the creation of descendants from one generation to the other, through the particle movement, from iteration to iteration. The following figure illustrates the movement of a given particle:

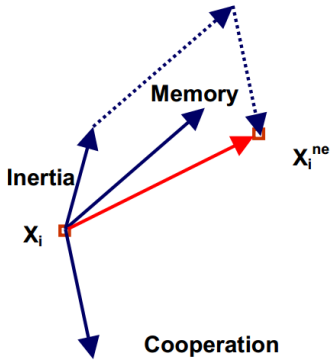

Figure 2.1: Illustration of a particle's movement, influenced by the three terms: inertia, cooperation and memory.

2.4.1 EPSO's ability for auto-adaptation

One of the highlights, if not the most decisive aspect of the EPSO model, is its capacity to auto-adapt in the resolution of any problem. It ables to automatically adjust its parameters and behaviour, as a reaction to the way in which the problem solving is being developed.

Such auto-adaptive models require the algorithm to, by itself, develop the ability to establish and modify its own behaviour according to the problem, avoiding therefore a direct dependency on an exterior rule. Therefore, one is able to conceive an algorithm with both learning ability as well as with intelligent behaviour.

From an historical perspective, auto-adaption strategies in evolutionary algorithms are known to have emerged in 1973, when *Schwefel* proposed mutation mechanisms for achieving better progress. In fact, through that mutation, one is able to reach values close to optimum, fostering a faster progress towards finding the optimum solution.

On the other hand, the algorithm's ability to auto-adapt allows a decrease of the dependence on external parameters. This fact increases the performance control on the algorithm itself. Despite the impossibility of rendering an algorithm totally independent, these offer a high problem solving capacity which often the non-adaptive models are not able to solve.

2.4.2 EPSO, description of the algorithm

The EPSO (Evolutionary Particle Swarm Optimization) can be defined as a **hybrid** process [15]. It merges 1) the optimization abilities of particle swarms through information exchange in the course of their movement, with 2) previously mentioned techniques linked to PSO. Concisely, it is described as an auto-adaptive evolutionary algorithm (sub-chapter 2.2.1) featuring a reorganization of the mutation process taken from the PSO (particle swarm optimization) model [1][2].

Inspired on the biology's philosophy, the model supports itself in a set of stages similar to the ones seen in the evolution theory of life:

1. **Replication** - Each particle X_i is replicated **r** times;
2. **Mutation** - Each particle X_i suffers a mutation of its strategic parameters w;
3. **Reproduction** - Each mutated particle X_i[1] generates a descendant according to the movement equation (eq. 2.1);
4. **Evaluation** - The *fitness* of the new individual is calculated based on the new position that is takes in the dimensional space;
5. **Selection** - Through Stochastic Tournament Selection or other equivalent processes, the best particle survives in order to give birth to a new generation. It ends up being composed by all descendants that have been selected from all particles of the previous generation.

[1] Not only conducts the weights w suffer mutations, so does the vector that saves the swarm's best position until that moment, b_G, consequently enabling an "agitation" even when all particles already converged to the same region of the space and are very close to one another.

The process proceeds in parallel to all individuals assuring thus a new coupling among them through the cooperation term. At same time it conditions the creation of new particles.

2.4.3 Control of the communication among particles and the selection process

Taking the formulation of the PSO model into account, one may conclude that the communication among particles define the efficiency of the algorithm. Therefore, two strategies are known to guarantee the *swarm*'s communication. Firstly, the classic *star scheme* model, which proposes that particles are more dependent on the global optimum in that moment. Secondly the *ring scheme* argues that a particle shows dependency on the best of a given conglomerate of two particles only. With this, it forms a link that results in a communication string and connects all particles.

In the EPSO model, the adopted communication system varies between a *star scheme* and an own version named *cognitive model* (where no communication exists among particles). A set of tests and experiments is put into practice, restricting therefore the communication flow on the problem's global optimum. This allows the search of a wider area for each particle and avoiding an unwanted and premature converge.

As previously explained, each particle of the solution *swarm* suffers a replication process. Each individual is replicated *r* times, and in which the strategic parameters - meaning the weights *(w)* adjacent to each individual - suffer a mutation. After this, it goes through the recombination process by the *movement rule* (described in eq. 2.1). Consequently, *r* descendants of the original particle are located in *r* different places from the dimensional space at stake.

Considering the steps taken up until this moment, it becomes necessary to proceed to the evaluation of the descendants. Firstly, the process of selecting the one with the less *fitness* in case of a minimization function. This way, the **selection or the selection operator** acts directly upon all descendants and one single individual is selected. Thus, be the target of the whole process in the following generation.

The aforementioned process is applied to all particles. As an example, in a problem with n=20 particles, the same amount of steps must be repeated throughout the procedure. Nonetheless, considering a model with exclusively cognitive nature, the communication among particles does not exists. Subsequently it becomes easier to predict that will occur 20 procedures completely independent from each other.

On that account, it becomes relevant to highlight the importance held by the communication term. In fact, communication conducts to avoid this, fostering a certain communication between the processes, and standardizing the *swarm*'s constant movement throughout its successive iterations.

2.5 Differential Evolutionary Particle Swarm Optimization (DEEPSO)

DEEPSO emerges from the fusion of the theoretical bases which are in the root of three evolutionary computation areas. Both the *Evolutionary Particle Swarm optimization (EPSO)* as well as the *Differential Evolution (DE)* add to the *Particle Swarm Optimization (PSO)*. All the previously stated models have similar objectives as far as their mission in concerned. However all of them face a few challenges. DEEPSO comes as successful attempt to find a more robust and general model, that compasses the best points of all the other ones [16].

Besides EPSO and PSO, DE shows a dominant role in DEEPSO modelling. Differential Evolution [17][18] is basically supported by the following: given an individual population, a new solution is generated. This value is created from a given existent particle by adding a fraction of the difference between two (2) points selected from the *swarm*, Xr_1 e Xr_2 [19].

Then, re-combinations assures higher diversity and selection produce a new generation. At last, the elitist selection acts at best solution conducting to its survival and passing on to the following generation. In this fashion, each parent gradually competes for survival with its direct descendant.

DEEPSO stands for the idea of adding a kind of noise to the EPSO-originated search, including an ED operator in the movement definition of a given particle. This same noise, influenced by the local macro-gradient (promoted by PSO), should supposedly result in a development of the global search process towards right direction [20].

DEEPSO presents a similar structure to the EPSO. Yet, in order to add the differential evolution nuance, its movement equation must suffer some alterations, such as:

$$\mathbf{V}^{(k+1)} = A.\mathbf{V}^{(k)} + B.(\mathbf{X}_{r1}^{(k)} - \mathbf{X}_{r2}^{(k)}) + P[C(\mathbf{b}_G^* - \mathbf{X}^{(k)})] \tag{2.3}$$

Where:

$$\mathbf{b}_G^* = \mathbf{b}_G(1 + W_G N(0,1) \tag{2.4}$$

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \mathbf{V}^{(k)} \tag{2.5}$$

$\mathbf{X}^{(k)}{}_{r1}$ and $\mathbf{X}^{(k)}{}_{r2}$ stand for a set of particles selected from the current generation. Several tests proved that, having developed based on macro-gradients [16] and in a context of minimization, the PSO results in a sorting of the selected particles such as the following condition:

$$\mathbf{f}(\mathbf{X}^{(k)}{}_{r1}) < \mathbf{f}(\mathbf{X}^{(k)}{}_{r2}) \tag{2.6}$$

These particles can be extracted from different sources. This conduct us to the definition of several DEEPSO variants. At that moment, one is able to extract particles from the current generation vector $\mathbf{P}_C$ or from $\mathbf{P}_b$which compasses the historical of the best particles. Further, the DEEPSO model proposes that $\mathbf{X}^{(k)}{}_{r2}$ equals to $\mathbf{X}^{(k)}$ so that $\mathbf{X}^{(k)}{}_{r1}$ is sampled.

More than an alternative model, DEEPSO is an EPSO-based evolutionary model. It is, in fact, a hybrid process that merges the optimization capacity of an evolutionary algorithm, with the differential evolution concept. In truth, it has proven to lead to better performances as far as problem optimization in energy systems [21][22]. These kind of issues are characterized by irregular profiles in the dimensional space resulting in the creation of relevant hurdles.

On the other hand, the own formulation of EPSO had already shed light into the advantage of its recombination with the adaptation capacity linked to PSO. DEEPSO suggests, therefore, that this recombination scheme should be broadened to the past set of the best particles.

The fact that both EPSO and DEEPSO show similar structures, supports the argument that the variants proposed to the first may be perfectly applicable to the latter.

2.6 Conclusions

The PSO model conducts not feature competition among particles. Subsequently results in its inability to auto-adapt to the way the problem is being dealt with. In this case, the progress towards the optimum is determined by the movement rule, which is responsible for the creation of the new particles.

On the other hand, in traditional evolutionary algorithms, the presence of mutation/recombination processes dictated the subsequent particle generations. Linked to this reality, convergence in the PSO model is dictated by the values that have been selected as weights, while in more traditional evolutionary models, an auto-adaptation of the strategic parameters occurs and hence contributes with its intelligence to the evolutionary process.

EPSO conducts stand out as the fusion of the best aspects of all worlds, since combines two mechanisms. This algorithm is able to "learn" which values it must define to mutation weights, therefore pushing the progress forward toward the problem's optimum. In other words, it combines two fundamental aspects: firstly, the movement equation - which is a distinctive aspect of PSO and that allows the recombination and creation of new particle generations -, secondly the selection operation with auto-adaptation capacity.

The combination of these two particularities in the improvement of the populations' several particles brings about higher convergence speed to the EPSO model. Moreover, it may also relevant to emphasize that the vector with the current best position also suffers a movement in the dimensional space according to a Gaussian distribution probability. As a result, throughout the progression process as a whole, the *swarm* experiences constant movement processes.

Last, the hybrid properties of EPSO give place to a more robust, more efficient and more reliable algorithm, whose results are considered better than any seen in the past [1]. In electric energy systems, this is used in a broad set of applications, such as, the reduction of energy loss and voltage control [23][2].

Chapter 3

First variant to EPSO Proposal: Change of variable as dimension re-scale - "VAREPSO"

The following chapter intends to explore the testing of a hypothetical EPSO algorithm first variant. It starts with a theoretical introduction, followed by the algorithm model and by the presentation of this variant's impact in the original model.

3.1 The differences of scale between Space Dimensions

It is widely acknowledged that a *swarm* or a single particle pushed by an optimization algorithm will travel through a given dimensional space of where a given objective function is defined. According to its topology, the *swarm* will explore the space and, depending on the values that each of the particles find, it will be attracted and modelled into that same specific space points, such as the *swarm*s' movement inertia for example, as well as the best solution found at that moment which was modelled by the cooperation factor as described in chapter 2.

In many optimization problems, the fact that the *swarm*'s morphology reveals a dimension disproportion in the space it occupies. That would be the case of an imaginary ellipsis representing the space occupied by a given swarm after *n* optimization iterations of a function. The below figure below translates that situation:

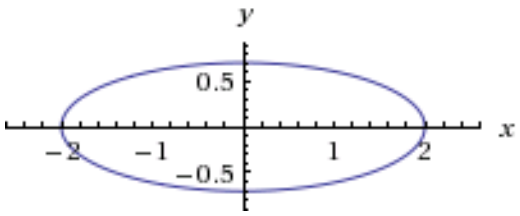

Figure 3.1: Example of an ellipsis.

Through figure 3.1 one can notice that there is a significant discrepancy. We can notice the difference in the space interval comprised between the lower and the higher coordinates in the axis *xx* to axis *yy* dimension. There is a *swarm*'s tendency to search the interior zone of the space it occupies. Consequently emerges the idea of intervene in the dimensional space according to certain criteria.

Bearing the previously mentioned example in mind, two possibilities for a transformation in one of the two dimensions seem feasible. Firstly, the reduction of the space occupied by the cluster in dimension *xx*. Secondly, the increasing of the interval in the *yy* axis. These two proposal would contribute to a higher standardization between dimensions, and, theoretically, contribute to a faster convergence of the optimization process.

The hypothesis that this thesis intends to test compasses the constant search for the transformation of the space, and basically the attempt to make it more similar to a sphere. Its symmetrical feature could be favorable to the progression of the EPSO model.

3.2 The differences of standard deviation between Space Dimensions

An alternative to the use of dimension difference would be to use the standard deviation as a criterion. As widely known, this step translates into a dispersion measure, which calculates the variability of the several values found around that average.

The standard deviation can be calculated using the following equation:

$$\sigma = \sqrt{\frac{1}{N}\sum(x_i - \mu)^2} \tag{3.1}$$

This way, by calculating the standard deviation by dimension, one could have an overview of the space that the particles occupy by dimension. This would accord to a similar criterion as the one previously stated. The most relevant theoretical advantage of this proposal consists in the fact that it is a measure based on the average. For example, in the section 3.1, the existence of an isolated particle – that does contribute to a significant difference in the dimension - is avoided when taking the standard deviation into account.

As such one could consider two hypotheses for analyzing the space that the *swarm* occupies in the dimensional space. It becomes thus necessary to understand which kind of criterion should be utilized while the algorithm runs, in order to understand the right timing for proceeding to transform the space.

3.3 Algorithm formulation

The next logical step would be to structure the alterations that are to be implemented into the original EPSO-algorithm (here stated in chapter 2.4 of). The EPSO model with the aforementioned alterations is presented as follows:

1. **New iteration**;

2. Registration - according to the proposed criterion - of the intervals, through the subtraction of the highest position by the lowest, by dimension;

3. Comparison of the registered intervals for all dimensions;

4. If noticed that a certain dimension presents double the interval of the other *(*$dimen_1$ */* $dimen_2$ *> 2)*:

   In $dimen_1$, for each of the particles, the axis coordinate to be set to half;

   Or

   In $dimen_2$, for each of the particles, the axis coordinate to be set to double;

5. When all dimensions are within an interval that respects point 4;

6. **New iteration**;

As mentioned in section 3.2 of this chapter, by using the standard deviation, as opposed to comparing the dimension difference, one is able to compare the standard deviation among dimensions.

3.4 Testing functions

Regardless of the optimization algorithm being used, the goal is always to find the best solution (optimum solution) or a group of solutions for a given problem. With or without restrictions, these problems stand out by their great complexity and by the broad amount of optimum places that oftentimes make the model's convergence process more complicated.

Therefore, it is fundamental to test the program with problems/functions, with distinctive aspects related to optimum places; discontinuity; the function's morphologies in its own space, as well as the amount of dimensions being considered. Having the solutions to the functions, one is able to proceed to an evaluation of the way the program develops, converging or not, with greater or lesser computational effort. This translates the efficiency presented by the model at stake.

When the program reveals good performances with these testing functions, it is considered able to deal with real problems, such as *optimal power flow*, or power forecasting issues of a wind farm, for example.

In this section, a short presentation will be done to optimization functions used as a reference. These conduct to performances comparisons between the original EPSO-algorithm and the variants to EPSO.

3.4.1 Rosenbrock function

The following function is commonly used in performance tests for optimization problems. Developed in 1960 by Howard H. Rosenbrock, this non-convex mathematical profile features a long parabolic valley, in which is hard to find an optimum solution. The function is defined by:

$$f(x_1 \cdots x_n) = \sum_{i=1}^{n-1} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2) \tag{3.2}$$

It presents a single solution at X_D=$(1,...,1)_D$, for a null function, $f(X)$=0.

The figure 3.2 illustrates the function in question, where one can clearly notice the flat parabola-shaped valley. In these sort of problems, usually hampers global convergence, and stands as one of functions most difficult to optimize.

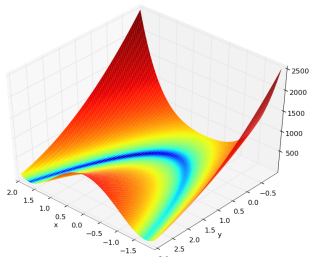

Figure 3.2: Rosenbrock function (3D).

3.4.2 Sphere function

The sphere function, which is easily identifiable through its symmetry, stands out as one of the most well-known functions. It is defined by the following expression:

$$f(x_1 \cdots x_n) = \sum_{i=1}^{n} x_i^2 \tag{3.3}$$

Centred in the axis, this function exhibits one single global optimum at X_D=$(0,...,0)_D$, represented in the following figure:

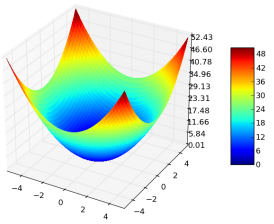

Figure 3.3: Sphere function (3D).

3.4.3 Alpine function

As opposed to the Sphere, the Alpine function does not hold symmetrical features, since it comprises a large amount of minimum places. Its "wavy" style can be seen as a challenge to the optimization algorithms, given the constant risk of being "jammed" in these local solutions. The following figure represents this function's morphology:

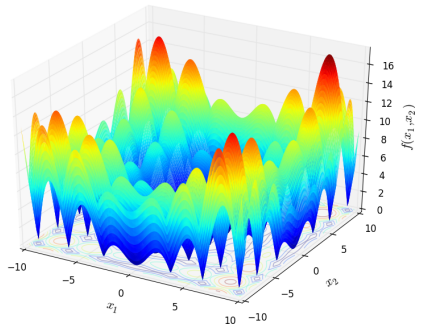

Figure 3.4: Alpine function (3D).

This function is formulated by the following expression:

$$f(x_1 \cdots x_n) = \sum_{i=1}^{n} |x_i sin(x_i) + 0.1x_i| \tag{3.4}$$

The global minimum is registered at X_D=$(0,...,0)_D$.

3.4.4 Griewank function

Defined by the following expression,

$$f(x_1 \cdots x_n) = 1 + \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) \tag{3.5}$$

the Griewank function stands out by its large amount of minimum places, which tend to increase in line with the amount of dimensions being considered, as visible in the following 3D representation of this mathematical model:

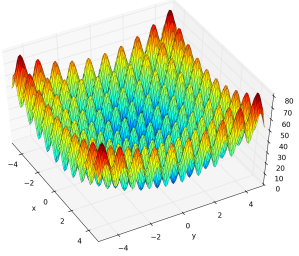

Figure 3.5: Griewank function (3D).

The optimum solution is the origin with value *f(X)=0*.

3.4.5 Ackley function

Last, the Auckley function stands as another example of an optimization model which can be mathematically expressed as follows:

$$f(x_0 \cdots x_n) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i))) + 20 + e \tag{3.6}$$

This function comprises one single global solution at its origin with null function value, represented by the following figure:

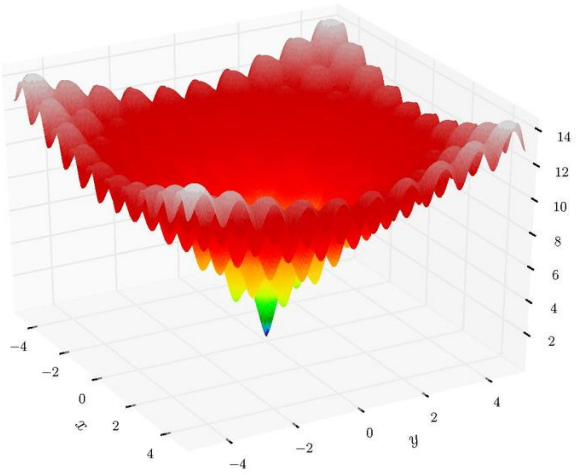

Figure 3.6: Ackley function (3D).

3.5 Testing the modified EPSO

A series of tests has been put in place in order to identify the best approach for a deeper understanding of this subject. Two alternatives were made available: 1) the hypothesis presented in section 3.1, where focus was about interval among particles in the extremities of each dimension; 2) hypothesis in section 3.2, which emphasizes the understanding of the way the individuals are disposed in space, according to dimension.

For the purpose of this thesis, the first criterion was deemed as the most appropriate since, theoretically, it leads to more practical and viable interpretations of the dimension space.

The target of **0.0001** *fitness* value was defined as the stopping criterion. Number of dimensions was established on **30**. With this, a bigger hurdle to overcome for the program. It was defined as nominal for the next tests that each swarm would compass **20 particles**. The results from the original EPSO model and of the proposed variant will be exposed later in this chapter.

The model's performance is measured in terms of computation effort required for the model's convergence. Bearing this in mind, we can compare the results achieved taking: 1) the **iteration amount** into consideration, meaning, the amount of generations which originated in the exploration of the tested dimensional space; 2) the **number of evaluations** registered by the objective function. This can be seen as a more correct measure since it registers the calculation of a given particle's adaptation.

Table 4.1 features the parameters linked to the tested functions:

Table 3.1: Parameters of the used testing functions.

Functions	No. of dimentions	Domain
Rosenbrock (f_1)	30	$[-50;50]^n$
Sphere (f_2)	30	$[-50;50]^n$
Alpine (f_3)	30	$[-10;10]^n$
Griewank (f_4)	30	$[-50;50]^n$
Ackley (f_5)	30	$[-32;32]^n$

3.5.1 Rosenbrock function

At an initial stage, one attempted to learn the way EPSO develops within a problem's optimization in regards to its original model. The following table comprises **10** tests made to the original EPSO, with random onset, an assuring therefore different initial populations within tests.

Table 3.2: Trial of the Rosenbrock function, with the original EPSO version.

Test	No. of Iterations	No. of Evaluations
1	10028	401120
2	9926	397040
3	12277	491080
4	11418	456720
5	9951	398040
6	10968	438720
7	10218	408720
8	10297	411880
9	10633	425320
10	9738	389520
Average	10545.4	421816

Table 3.2 allows the understanding of the great complexity shown by the Rosenbrock function, demonstrated by its broad amount of iterations and evaluations.

Consequently, the next logical step was to implement the proposed variant. The test was immediately focused on transforming the swarm's dimensional space according to the way it was structured at that moment. As stated in section 3.1 of this chapter, two possibilities arise: identified a given dimension that records an interval between particles which is double of the interval of another, one could 1) reduce the first or 2) increase the second. To keep the original characteristics of the target function, we took into consideration the variable change needed in the target function.

Another important point is to understand the conditions in which this variant is to be applied. There are several options to proceed to implement this manipulation: 1) on each iteration, one could analyze and perhaps intervene or 2) after n iterations, so the EPSO model progresses for a number of iterations spreading it self into dimensional space.

In fact, after some trials made, it was **registered that the implementation of this variant in all EPSO iterations made the model slower and inefficient**. This specific variant results in a restriction of the EPSO progression. Hense it limits the space exploration by successive transformations and may appear as possible causes for the above mentioned issue.

Imputing the variant in every single iteration could not be a good strategy. It was visible that, after few iterations, for less levels of *fitness*, the space limitation made the model slower. Consequently, the main idea was setting up the program at its initial stage, in order to guide the way towards a theoretically faster and more efficient convergence.

The priority was now to successively act in the model, where the **20 first iterations** were object of an analysis and, in need be, of an intervention along the aforementioned lines.

The following table exposes the results of 10 tests done to the Rosenbrock function, in which the variant was implemented up until the 20th iteration:

Table 3.3: Trials to Rosenbrock function, with EPSO's first variant, 20 first iterations.

Test	No. of Iterations	No. of Evaluations
1	9998	399920
2	12418	496720
3	8113	324520
4	10744	429760
5	9496	379840
6	8343	333720
7	9280	371200
8	10063	402520
9	10610	424400
10	8712	348480
Average	9777.7	391108

At a first glance, with table 3.3 can notice a lower iteration and evaluations average for tests with the new EPSO-variant (compared with table 3.2).

The specific and complex nature of this function conducts to the great challenge of improving the programs efficiency. Besides, it is clear that the analysis to the dimensional space in the first iterations often fosters not only a quicker convergence when compared to the original model, but also a decrease in the amount of *fitness* evaluations, respectively. Table 3.3 shows convergence cases with lower amounts of generations. This proves that these alterations, besides not affecting the quality of the results, succeed in being more efficient in many other less complex computational tests.

A few charts proving what has been argued are presented below. These illustrate the model's progression at its initial stage, namely the *fitness* value for the first EPSO iterations depending on the amount of evaluations that were conducted. At this stage, **the parity among the initial generated populations was assured**. At the beginning of the program, the 20 particles would remain the same independently of the EPSO's implementation.

Take a look at the following illustration:

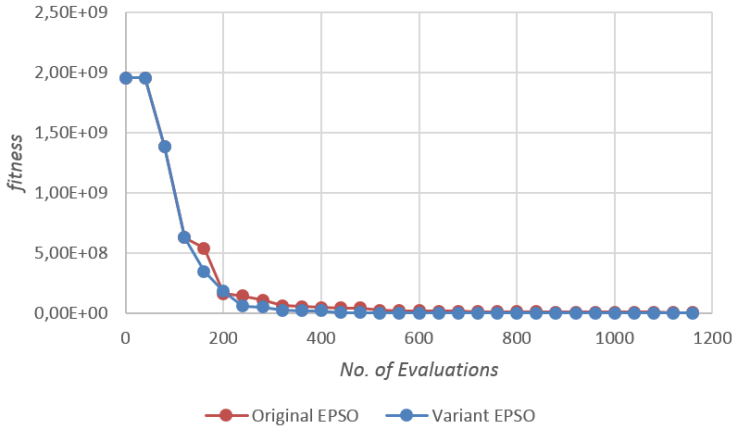

Figure 3.7: Progression of the EPSO original model vs the variant to EPSO, featuring the same initial population (Rosenbrock's function).

The following chart presents an enlargement of figure 3.7, after 180 evaluations were done.

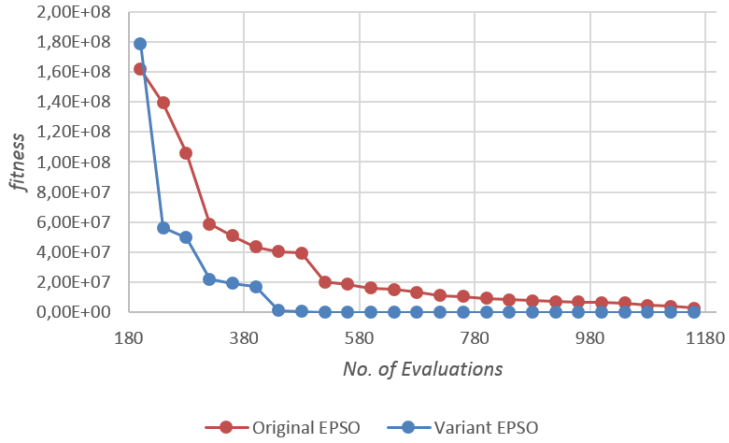

Figure 3.8: Progression of the EPSO original model vs the variant to EPSO, featuring the same initial population (Rosenbrock's function).

The variant's capacity to interpret the dimensional space in a quicker and in a computationally more demanding way - considering it requires a less amount of *fitness* evaluations for reaching zones that are potentially closer of Rosenbrocks' function optimum - is clear in the two previous figures (fig. 3.7 and 3.8).

Another relevant aspect of this case study is considered next. How would be the average behaviour the variant is able to offer to initially different populations? With that purpose, **five (5)** individualized tests to the original were conducted. In parallel, other five (5) trials with the variant to EPSO were also done. Therefore we reached an average of the progression of both algorithms.

Results are represented in the following figure:

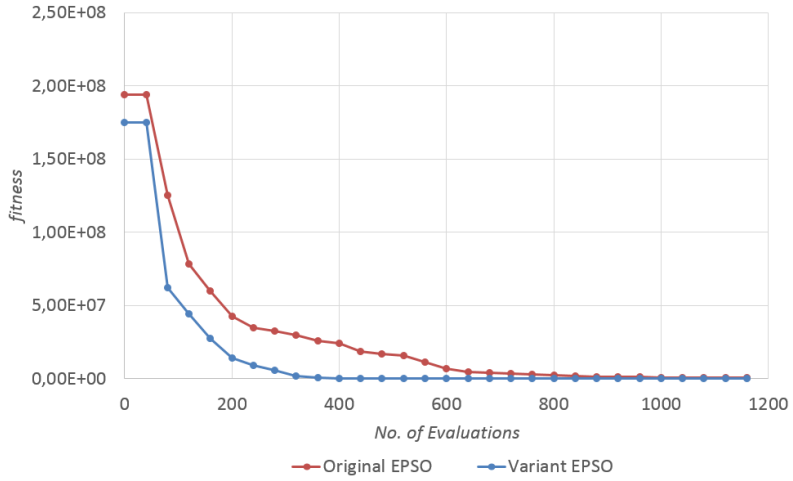

Figure 3.9: Average performance variations between the original EPSO vs the variant to EPSO, considering distinct initial populations (Rosenbrock's function).

Trough figure 3.9 one can verify the argument that has been previously exposed in this thesis. At an initial stage of the progression, the space interpretation allows EPSO guaranteeing some degree of parity between dimensions. Consequently, itself adapts quicker to that same space and finds zones with high convergence potential, more efficiently.

However, the complex morphology presented by this function leads to a shorter improvement margin. At a later stage this thesis will present more evident results ensured by this variant on other functions.

3.5.2 Sphere function

The next function approached in this paper is the Sphere function. This mathematical model differs from all others due to its symmetrical properties. It implies that a dimension transformation and re-interpretation may most likely have little impact on such kind of functions.

In fact, the equivalence between dimensions is the base argument behind this variant's theory. EPSO - at its original version - presents a great ability to solve this function with results in quick convergence.

Next, the results of the implementation of the proposed variant in a sphere will be presented. Used the same test configurations mentioned in chapter 3.5, the following table reflects the results achieved with regards to the original EPSO version, with distinct populations at its initialization:

Table 3.4: Trials on the Sphere function, with the original EPSO version.

Test	No. of Iterations	No. of Evaluations
1	505	20200
2	486	19440
3	562	22480
4	519	20760
5	511	20440
6	567	22680
7	488	19520
8	479	19160
9	493	19720
10	571	22840
Average	518.1	20724

Similar to the procedures followed in Rosenbrock, and in an attempt to replicate previously registered behaviour into a sphere – one opted to apply the variant in the **20 first iterations** of this function. One can consider the following table accordingly:

Table 3.5: Trials on the Sphere Function, with the first EPSO variant, 20 first iterations.

Test	No. of Iterations	No. of Evaluations
1	467	18680
2	559	22360
3	551	22040
4	457	18280
5	416	16640
6	498	19920
7	476	19040
8	433	17320
9	540	21600
10	378	15120
Average	477.5	19100

An initial evaluation of the previous results does not enable a clear perception of the impact that the variant has in the function's optimization. Despite both the average number of iteration and the amount of *fitness* evaluations being slightly lower, the implemented variant leads to a minor reduction.

As already stated at the beginning of this sub-chapter, the symmetry of the sphere function results on these short differences of convergence levels.

In order to face the need of going into further detail, the function's progression at the initial phase of its optimization was re-analyzed. The following chart shows the *fitness* development in relation to the computational effort, considering the same initial population:

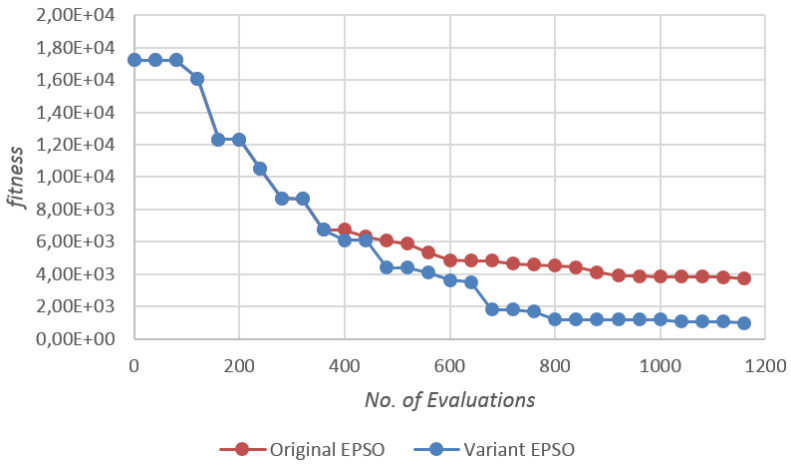

Figure 3.10: Average performance variations between the original EPSO vs the variant to EPSO, featuring the similar initial populations.

Figure 3.10 reveals the impact that this variant's space interpretation has in the development of the EPSO model. Despite not starring the quickest and more efficient convergence, since the figure does not show the rest of the progression up until convergence, this variant allows the optimization of *fitness* levels with lower computational efforts.

As a conclusion, despite the fact that these are not evident results of an increased function's performance, these records can be exciting and advantageous for functions with other types of irregular characteristics.

3.5.3 Alpine function

In contrast with the previous function, this optimization is based on a problem with an irregular morphology which compasses a high amount of minimum locals. As such, variant to EPSO would theoretically show some advantages as far as the program's progression is concerned.

After a set of experiments, one has opted for applying the model not only in its initial iterations, but also **throughout the whole process**. In other words, the proposed model was applied from one generation to the next.

The following table exposes the results of the two versions of EPSO:

Table 3.6: Trials on the Apline function, with the original EPSO version.

Test	No. of Iterations	No. of Evaluations
1	2785	111400
2	1003	40120
3	1783	71320
4	1164	46560
5	1068	42720
6	1198	47920
7	1465	58600
8	1504	60160
9	1322	52880
10	1383	55320
Average	1467.5	58700

Through the application of the EPSO variant, the following was registered:

Table 3.7: Trials to Alpine function, with first EPSO variant, implementing it in all iterations.

Test	No. of Iterations	No. of Evaluations
1	42	1680
2	47	1880
3	87	3480
4	42	1680
5	49	1960
6	43	1720
7	38	1520
8	36	1440
9	49	1960
10	41	1640
Average	47.4	1896

At a first glance one can easily identify a big difference between the effort that the Alpine convergence requires for EPSO at its original version. Having considered the results, one can

verify a reduction of approximately **96%** in the involved computational effort. The following figure features the *fitness* values progression differences according to the computational effort that was directly involved.

The aforementioned relationship is illustrated in the following figure:

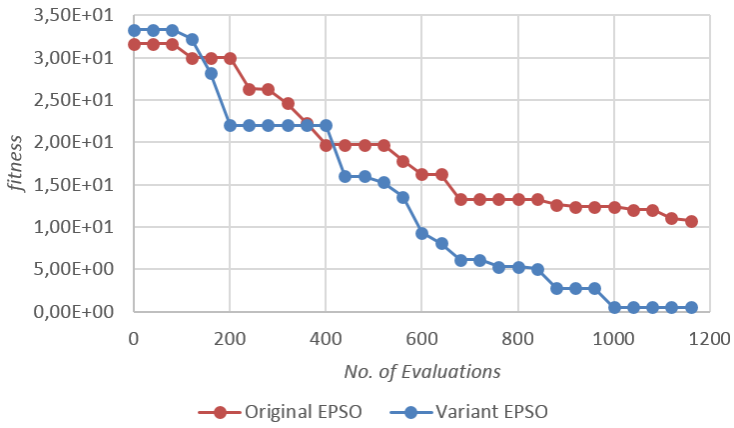

Figure 3.11: Relationship between *fitness* and evaluation amount, for both EPSO versions in the Alpine function.

Figure 3.11 shows the significant difference that this variant offers in regards to convergence in the evolutionary model.

Having followed these procedures, the question arises if this is the case for this specific function only. Several testing targets will thus be defined, such as, for example the significant amount of optimum places – which stands as this function's biggest hurdle -, among others. As such, one will look for functions with similar characteristics.

3.5.4 Griewank function

In line with the previous section, the EPSO variant has been applied in all iterations of this specific problem, utilizing the same configuration reached in the Alpine function.

Resorting to the modified EPSO model, the following values have been registered:

Table 3.8: Trials on the Griewank function, with the original EPSO version.

Test	No. of Iterations	No. of Evaluations
1	160	6400
2	174	6960
3	149	5960
4	1542	61680
5	2611	104440
6	146	5840
7	175	7000
8	180	7200
9	160	6400
10	195	195
Average	549.2	21968

Table 3.9: Trials on the Griewank function, featuring the first EPSO variant, and implemented in all iterations.

Test	No. of Iterations	No. of Evaluations
1	17	680
2	18	720
3	20	800
4	18	720
5	17	680
6	19	720
7	15	600
8	18	720
9	19	760
10	15	600
Average	17.5	700

Considering the two tables above, one can notice that, also in Griewank's case, the modified EPSO is able to reach converge values for computationally lighter and more efficient levels. In this case, the reduction was also around **96%**.

Another interesting aspect of this variant is visible through table 3.8. It is understandable through this table that the amount of evaluations is considerably higher when compared to the rest of the tests. In truth, it was registered that the original EPSO faces a modest obstacle in terms of

convergence, for this particular function. In some trials, we registered that it prolongs themselves for a large number of iterations. This may be due to the fact that it is stuck at a local minimum, where original EPSO shows little ability to converge after that.

By applying the EPSO alteration, one can notice that the space interpretation contributes to program's permanent progression, hence **eliminating the possibility of getting stuck in one of the obstacles which are specific to this function**.

Considering the figure below:

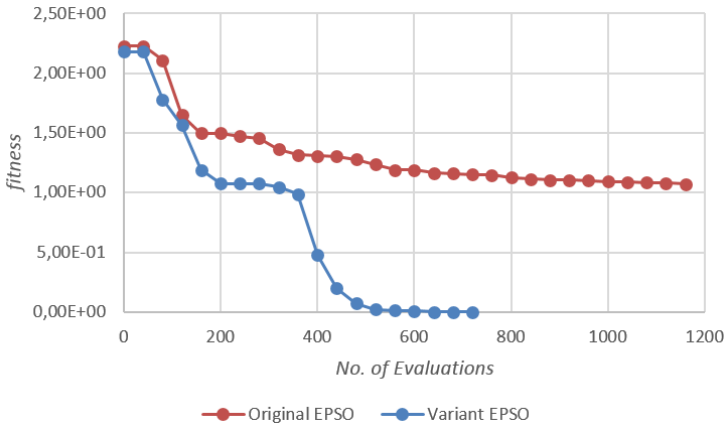

Figure 3.12: Relationship between *fitness* and evaluation amount, for both versions of EPSO, in Griewank function.

Similar to the function addressed earlier in this chapter, Figure 3.12 shows the difference in the behaviour of both EPSO versions. This variant's ability to optimize the original algorithm is clearly evident in the chart above, with a more efficient convergence.

3.5.5 Ackley function

At this stage, it is still deemed necessary to analyze the behaviour of one last optimization function. Known as the Ackley function, this mathematical problem presents a similar morphology, namely a roughness throughout its dimensional space. On the other hand, the origin reveals a deep valley that leads to global minimum. Taking these aspects into consideration the variant to EPSO model was expected to succeed in straightaway. After detecting the zone where most particles were concentrated, the existence of a consequently higher convergence probability.

The table below presents the results of the computational effort of the EPSO's original version:

Table 3.10: Trials on the Ackley function, with the original EPSO version.

Test	No. of Iterations	No. of Evaluations
1	263	10520
2	251	10040
3	269	10760
4	266	10640
5	307	12280
6	229	9160
7	308	12320
8	245	9800
9	295	11800
10	265	10600
Average	269.8	10792

By repeating the process with the currently altered EPSO, the following results were achieved:

Table 3.11: Trials on the Ackley function, implementing the first EPSO variant into all iterations.

Test	No. of Iterations	No. of Evaluations
1	21	840
2	21	840
3	24	960
4	18	720
5	25	1000
6	18	720
7	25	1000
8	26	1040
9	20	800
10	23	920
Average	22.1	884

With analysis made at all values presented in the previous tables, it becomes clear that the

variant proposed throughout the current chapter does have a rather significant impact on the improvement of EPSO's performance.

The reduction value of the computational effort is approximately **92%**. The following figure illustrates these results along the lines of the previous functions. It not only allows us to learn the real difference between both models, but also allows us to understand this variant's abilities in terms of optimization of this function.

Take a look at the following illustration:

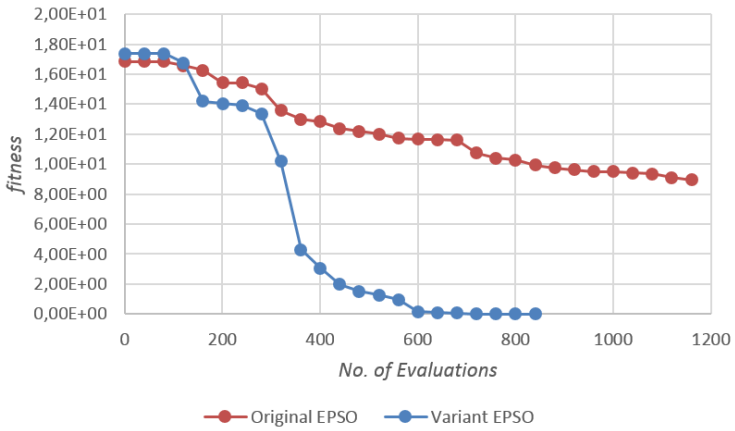

Figure 3.13: Relationship between *fitness* and evaluation amount for both EPSO versions in the Ackley function.

One may conclude, therefore, that all the above mentioned functions have registered an improvement of the EPSO efficiency when compared to the results obtained in the original version of the model.

3.6 Main conclusions

All through chapter 3, this thesis intended to put forward the possibility of analyzing, in an active way and throughout the EPSO's complete development process, the way a given swarm interprets the dimensional space of the objective function. It was clear that the *swarm* located itself in space in a way that its dimensions were completely disparate from each other could lead to an additional computational effort. By decreasing those differences, the set of particles itself reacts in order to understand which zone it must look for, acting upon the possibility that the global minimum of the system and the solution to the problem stands in the central zone of the individuals' conglomerate.

Functions such as Rosenbrock, which present a complexity much higher than the average, in combination with the parabola-shaped depression zone, constitute a hurdle. This stems from the existence of a significant disproportion in differences among each dimension, similar to a two-dimension parabola where the dimension interval differs from the one registered in the latter. Consequently the swarms' convergence is jeopardized, which, at this stage of EPSO, is not able to perceive those differences between dimensions.

Several other hypotheses have been analyzed in this chapter. Interesting would be, as an example, instead of resorting to the reduction of the dimension which presents the higher interval, to opt for applying an interleaved transformation. It consists on a decrease in the bigger dimension and a subsequent increase to the most concentrated dimension, with regards to the particles' conglomeration at its extremities. This attempt intended not only to avoid the consecutive reduction of the dimension space but also to enlarge the narrowest dimension to higher intervals.

The findings have been similar to the ones exposed previously in this chapter, so that one is able to argue that it is considered as more practical to always treat space in the exact same way.

Chapter 4

Second variant to EPSO Proposal: Satellite Swarms - "SUBEPSO"

The current chapter presents the formulation of a new variant of the EPSO algorithm, through the implementation of specific *swarms*, named **son-swarm** or **sub-swarm**. Along with the description of the variant's formulation, the investigation work will be illustrated with practical examples depicting the algorithm's behaviour as a result of these new implementations.

4.1 Sub-swarms implementation

This research project is inspired on the idea of generating new particles' swarms within the initial swarm. Named **mum-swarm**, these new swarms would increase the general ability of deepening certain zones in the problem's domain. Alongside with this, it would allow the identification of possible spaces which would assume bigger importance within the problem and, consequently would lead to a hypothetical intervention on the particle's movement of the swarm.

considering any given optimization problem, the algorithm develops its space exploration strategy and reaches successively better results, presenting therefore a sequence of global optima that are updated. These serve as attractions, to *swarm*'s movement throughout its exploration. However, as the *swarm* moves, the possibility that the *swarm* jams itself into an undesired local minimum becomes bigger.

Taking the aforementioned facts into account, the idea of generating new swarms in order to make a deeper analysis of specific zones seems logical. As such, following a determined occurrence criterion, a random particle is selected from the mum-swarm. With this, a new *swarm* is created and confined to the surrounding area of the **randomly chosen particle.** Theoretically, this approach seems to be a feasible strategy for avoiding possible cases of insistence in zones of minimum locals.

On the other hand, as the algorithm advances, the space position with best *fitness* function, known as global optimum, would also be utilized for a new *swarm* generation. Now, **the best position until then is selected as *swarm*'s position reference**. This conducts to a deeper search

in these specific zones and oftentimes could lead to a quicker and more efficient convergence to global best solution.

As a summary, the generation of these new *swarms* are defined as more precise and pretend occupy a specific space in the domain. Inspired on two hypotheses both could be applied individually or even simultaneously during the EPSO process. The first to be considered is the creation of a son-swarm based on the global optimum registered at a given moment. In other words a new *swarm* will arise in the position of the best current solution, who will be responsible for investigating that specific space zone. The second hypothesis implies that, as an example, one of the 20 particles of the mum-swarm would be selected as a target to a more precise search.

These two hypotheses support this chapter's general concept. Although we accept the fact that such implementation will require an extra computational effort, due to several *fitness* evaluations applied to these sub-swarms. Thus, what remains is to test if this extra effort may be helpful for reaching a more efficient convergence of the main swarm.

4.2 Algorithm Formulation

The implementation of these variants comprise a set of steps. Following the base EPSO process (described in chapter 2.4.2 of this paper), in each new iteration the particle's position is saved to be the origin of a new *swarm* around it. A characterization of this line of thought will be discussed in further detail in the following sections of this chapter.

4.2.1 EPSO iterative model – sub-swarm based on the global optimum

After each iteration of EPSO, it takes place on movement equation the process of selection. At same time, the global best position of system until this moment is a member of movement equation. Hence, this particle position is used as the center of generation of a new *sub-swarm*. This new group of particles remains restricted to spread around this position with a specific percentage limit. The next scheme presents the formulation of this thought:

1. **New iteration;**
2. Registry of the current global optimum's position;
3. Generation of the new sub-swarm whose exploring limits are defined as up to 10% of the position of the optimum global;
4. 50 (for example) iterations of the new son-swarm;
5. Analysis of the result achieved, and verification whether its global optimum has better *fitness* than the value held by the mum-swarm;
6. In case an improvement is registered, updating of the mum-swarm new global optimum of the attained by the satellite *swarm*;
7. **New iteration;**

4.2.2 EPSO iterative model – Sub-swarm based on a random particle

Similarly, this second takes a similar process, at each iteration a randomly particle is selected. Its position vector is subtracted and a new swarm is generated around the particles position at that given iteration. considering the following model:

1. **New iteration;**
2. Random selection of a particle;
3. Registry of the selected particle's position;
4. Generation of the new sub-swarm whose exploration limits are defined as up to 10% of the particle's position;
5. 50 (for example) iterations of the new swarm;
6. Analysis of the obtained result and verification whether its new global optimum holds better *fitness* that the value held by the mum-swarm;
7. In case of improvement, the mum-swarm's new global optimum is updated according to the new solution attained by the satellite *swarm*;
8. **New iteration;**

Having structured both models, the hypothesis of testing variants simultaneously arose. From an iteration to the other, this would result in the creation of two sub-swarms. Theoretically, it allows the program not only to analyse in more detail the space of the actual best particle, but also to explore random areas, that could present solutions with more consistent *fitness* results than the ones found up until that moment. The following sections of this chapter will report the results achieved.

4.3 Testing the modified EPSO

This sub-chapter comprises a series of tests with the proposed modifications. Once again, recurring to diverse optimization functions (presented in chapter 3.4). The focus was not only on the algorithm computational effort, but also on the **amount of iterations** and **number of *fitness* evaluations** that lead to the program's convergence. In other words, it urged to understand the benefits (or lack of them) brought about by the presence of a more localized swarm on the original algorithm's efficiency.

The initial configurations were set in a similar fashion to the ones defined in chapter 3. The stopping criterion was defined as the *fitness* value of **0.0001**, with **30 dimensions** being tested. The amount of individuals comprised by each swarm was limited to **20 particles**.

The below table exposes the parameters according to which the test functions where processed:

Table 4.1: Parameters for the used test functions.

Functions	No. of dimentions	Domain
Rosenbrock (f_1)	30	$[-50;50]^n$
Esfera (f_2)	30	$[-50;50]^n$
Alpine (f_3)	30	$[-10;10]^n$
Griewank (f_4)	30	$[-50;50]^n$
Ackley (f_5)	30	$[-32;32]^n$

4.3.1 Sub-swarm based on a global optimum particle

As previously stated, there were three hypotheses linked to this variant's formulation, namely: 1) to base the swarm on the global optimum or 2) to base it on a random particle, or even 3) to base it on both aspects simultaneously. However, at this stage, the main effort relied on understanding the impact that the first exerted on the EPSO's performance. As such, **the first trials were upon a sub-swarm created in the position of the best current solution (global optimum)**.

4.3.1.1 Rosenbrock function

Once more, the objective was attempting to improve the whole optimization process of the Rosenbrock function. Starting from testing the original EPSO, the achieved results are similar to the ones provided in the previous chapter. It is important to remember that these values came with random and distinct initial populations. Thus resulting in an average of computational effort required by the process.

Table 4.2: Trials on the Rosenbrock function, with the original EPSO-version.

Test	No. of Iterations	No. of Evaluations
1	10028	401120
2	9926	397040
3	12277	491080
4	11418	456720
5	9951	398040
6	10968	438720
7	10218	408720
8	10297	411880
9	10633	425320
10	9738	389520
Average	10545.4	421816

First, it was necessary to understand the moment when the variant was deemed to applied. In other words, when was it necessary to create a new *swarm*. In all new generations of the mum-swarm? Should we create the son-swarm only after n iterations? Several possibilities seem feasible.

In order to take this decision, the **initial population was fixed**. With this, the particles that form the first generation are the same at these following tests. On the other hand it was decided that the sub-swarm would be composed of **20 particles**. As such, after running the original EPSO method, Rosenbrock optimization displays the following performance:

Table 4.3: Results of Rosenbrock's optimization, with the original EPSO version.

No. of Iterations	No. of Evaluations
11275	451000

Having established a comparison term for EPSO's behaviour, the subsequent test implements the variant of the son-swarm's creation in the position of the global optimum. The results of the implementation of the first variant proposed in this thesis (chapter 3, allowed the understanding that an action at the initial stage of the progression brings positive results to its convergence. Taking that into account, this second variant encompasses the implementation of son-swarms in the **20 first iterations**.

A son-swarm is generated for each of the mom-swarm's iteration. This progresses for a maximum of five (5) generations. If, along this process, the sub-swarm manages to find a space position with a lower *fitness* than mum-swarm's global optimum, we opt for deepening that search by broadening the progression of the son-swarm to extra five (5) iterations. If no better result is obtained, we opt for rejecting and eliminating it, hence allowing another generation of the original *swarm*. The results were as follows:

According to the definition of the action patterns at this first trial, the results of the variant to EPSO application, were:

Table 4.4: Results of the Rosenbrock's optimization, by applying the second variant to EPSO (5 iterations + 5 if needed).

No. of Iterations	No. of Evaluations
9716	388640

Table 4.4 registers an inferior computational effort. Further, another trial with a slightly different settings was executed. Hence, in the following test, one tries to deepen the search process of the son-swarm's dimensional space. In other words, in case a better solution than the global optimum is attained, instead of progressing another five (5) generations, it proceeds to search **ten (10) generations** more. Apart from that, the process remains completely similar to the previous one. The results were as follows:

Table 4.5: Results of the Rosenbrock's optimization, by applying the second variant to EPSO (5 iterations + **10** if needed).

No. of Iterations	No. of Evaluations
10067	403120

Despite the prematurity of the results it illustrates, the table above exposes a convergence of the Rosenbrock function with a sequence of values slightly inferior to the ones registered in table 4.3, but worse than the test immediately before (4.4).

The next logical step would be to learn the impact of a more significant depth increase in the search of the satellite *swarm*. This would allow to observe its particles' behaviour as a consequence of their longer progression. Therefore, this third test comprises an increase from ten (10) to **fifteen (15) generations**. It would be expectable for this trial's result to be computationally worse, since the iteration amount of the sub-swarm was increased and it consequently implied extra computational work. The results were as follows:

Table 4.6: Results of the Rosenbrock's optimization, by applying the second variant to EPSO (5 iterations + **15** if needed).

No. of Iterations	No. of Evaluations
10072	403520

Taking the above data into consideration, it is clear that **the increase in the generation amount becomes irrelevant during EPSO**. In fact, it requires a slightly higher effort and thus diverging from this thesis' intention.

The next step is to test the hypothesis that a better performance was registered (table 4.4). Subsequently apply a series of trials with distinctive and random initial populations to the original EPSO, which led to the values illustrated in table 4.2. The table below translates the results of the implementation of variant to EPSO (5 iterations + 5):

Table 4.7: Trials on the Rosenbrock function, by applying the second variant to EPSO (5 iterations + 5 if needed).

Test	No. of Iterations	No. of Evaluations
1	9008	360640
2	10666	426960
3	9104	364520
4	10645	426160
5	9052	362400
6	8264	330880
7	9291	371960
8	10435	417720
9	9707	388600
10	10354	414520
Average	9652.6	386436

Looking at the average results registered in the table above, the comparison with the original EPSO's behaviour (table 4.2) can be done. We can denote a modest improvement in computational performance, mainly due to the decrease in the amount of evaluations and iterations required for reaching convergence. On the other hand, as was mentioned in chapter 3, this optimization problem holds special features with regard to its global optimum zone. For the aforementioned reasons, it becomes extremely important to interpret the way in which dimensions interconnect and correlate with each other in order to strive for more efficient results.

The following graph depicts two curves that illustrate the progression of EPSO's *fitness* value at its initial development stage. EPSO original version and in the variant tested in the previous table are here illustrated. For random initial populations, it presents computation efforts average from several tests made.

Take attention to the following figure:

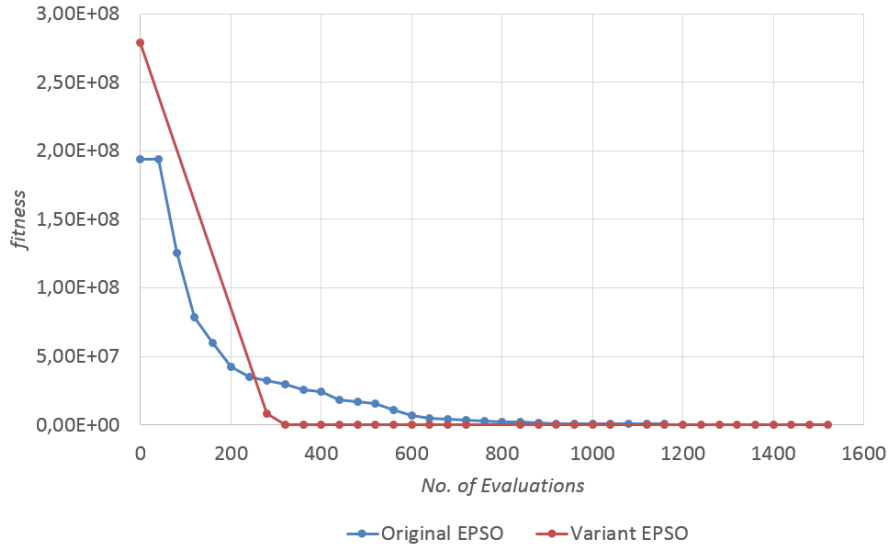

Figure 4.1: Progression of the EPSO original model vs variant to EPSO, in Rosenbrock.

It is important to highlight that the marks along the curve stand for a new iteration throughout development of the mother swarm.

Logically, when a sub-swarm is generated, the evaluations amount increases proportionally, since the latter experiences a particular development. The previous illustrations clearly put forward the progression difference between two characteristic curves. The implementation of satellite swarms on the initial stage of the EPSO development through allow the searching in more detail for a certain zone and stands as catalyst to its convergence.

4.3.1.2 Sphere function

Focus now on the other well-known function. The intention is to understand if the currently proposed variant would have a direct impact on this problem. Taking the findings of the previous chapter into account, the symmetrical features of this type of function stood as a cause for the lack of improvement in computational performance in the moment of its application into the modified EPSO.

The configuration of the variant to EPSO for this problem was similar to the one established for previous tests. With the implementation of son-swarms for **each of the first 20 iterations of the mum-swarm**, this comprises **20 particles** and **develops toward maximum 5 iterations**. In case of the appearance of a particle with inferior fitness than the global best, then another **5 iterations are executed in order to deepen that search**. Therefore, results can be considered as follows:

Table 4.8: Trials on the Sphere function, applying the original EPSO version.

Test	No. of Iterations	No. of Evaluations
1	505	20200
2	486	19440
3	562	22480
4	519	20760
5	511	20440
6	567	22680
7	488	19520
8	479	19160
9	493	19720
10	571	22840
Average	518.1	20724

Applied the modified EPSO:

Table 4.9: Trials on the Sphere function, applying the modified EPSO.

Test	No. of Iterations	No. of Evaluations
1	573	23240
2	792	32000
3	397	16200
4	427	17440
5	543	22040
6	694	28080
7	462	18840
8	471	19200
9	510	20720
10	361	14760
Average	523	21252

Analysing the results achieved for Sphere function, the variant to EPSO appears to be less interventionist and less efficient than its original version. This optimization problem in particular presents a very individual morphology, which EPSO, at its original version, succeeds in being able to solve.

4.3.1.3 Alpine, Griewank and Ackley functions

In this section, the target will lay on the remaining functions that have been tested. As we registered in chapter 3, it was verified that by intervening in all iterations/generations of the main swarm, one is able to obtain rather satisfactory results. Therefore, in order to test these problems, **one starts from creating new sub-swarms for each of the first 20 generations of the mum-swarm, to creating son-swarms in all iterations of the first**. As previously performed, it is necessary to formulate an variant to EPSO with the following settings:

- Sub-swarm located in the position of the current global best;
- This sub-swarm is confined to a research area up to 10% of the coordinates of the reference particle;
- Amount of particles of the mum-swarm: **20**;
- Amount of particles of the son-swarm: **20**;
- Progression of the son-swarm: **5 iterações + 5** (in case of improvement of the global optimum).

Having defined the action principles of the modified EPSO, the performance differences between the original EPSO and the variant proposed here will be presented next.

4.3.1.4 Alpine function

Table 4.10: Optimization tests, with original EPSO.

Test	No. Iterations	No. Evaluations
1	2785	111400
2	1003	40120
3	1783	71320
4	1164	46560
5	1068	42720
6	1198	47920
7	1465	58600
8	1504	60160
9	1322	52880
10	1383	55320
Average	1467.5	58700

Table 4.11: Optimization tests with variant to EPSO.

Test	No. Iterations	No. Evaluations
1	11	680
2	11	680
3	11	680
4	10	640
5	10	640
6	11	680
7	10	640
8	10	640
9	10	640
10	11	680
Average	10.5	660

4.3.1.5 Griewank function

Table 4.12: Optimization tests with original EPSO.

Test	No. Iterations	No. Evaluations
1	160	6400
2	174	6960
3	149	5960
4	1542	61680
5	2611	104440
6	146	5840
7	175	7000
8	180	7200
9	160	6400
10	195	195
Average	549.2	21968

Table 4.13: Optimization tests with variant to EPSO.

Test	No. Iterations	No. Evaluations
1	7	520
2	8	560
3	8	560
4	7	520
5	8	560
6	8	560
7	8	560
8	7	520
9	8	560
10	8	560
Average	7.7	548

4.3.1.6 Ackley function

Table 4.14: Optimization tests with original EPSO.

Test	No. Iterations	No. Evaluations
1	263	10520
2	251	10040
3	269	10760
4	266	10640
5	307	12280
6	229	9160
7	308	12320
8	245	9800
9	295	11800
10	265	10600
Average	269.8	10792

Table 4.15: Optimization tests with variant to EPSO.

Test	No. Iterations	No. Evaluations
1	12	720
2	13	760
3	13	760
4	14	800
5	12	720
6	12	720
7	13	760
8	11	680
9	13	760
10	13	760
Average	12.6	744

Observing the previous tables, it is clearly noticeable the EPSO's variant capacity to optimize the convergence process for these optimization problems. In all three previous tests, reductions on the computational effort reached over **95%**. Again, the current variant holds a very significant impact in progression, by understanding the best strategy for dealing with a certain problem.

The following graphs demonstrate graphically the *fitness* value medium progression curves, according to its required amount of evaluations.

It is important to highlight that the marks along the curve, stand for a generation of particles.

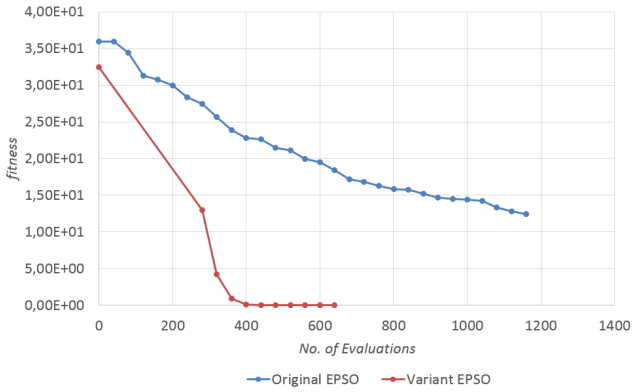

Figure 4.2: Progression of the EPSO original model vs variant to EPSO, in Alpine.

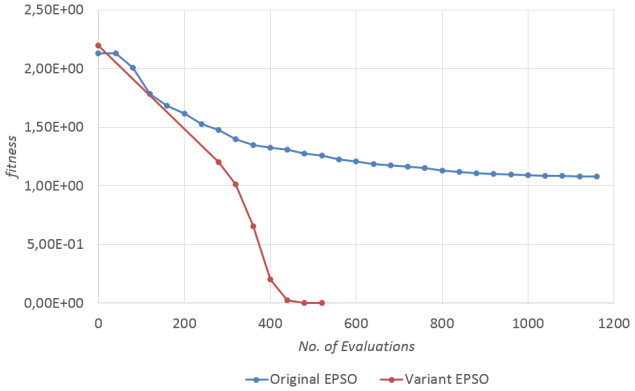

Figure 4.3: Progression of the EPSO original model vs variant to EPSO, in Griewank.

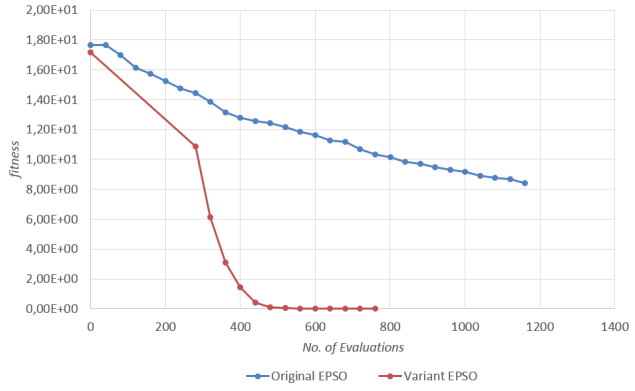

Figure 4.4: Progression of the EPSO original model vs variant to EPSO, in Ackley.

The three figures above reveal the new EPSO algorithm's great ability in terms of convergence, being specially clear the moment when it reached the optimum solution, while the original shows a high *fitness* value.

4.3.2 Summary

Throughout this section 4.3, a series of tests executed recurring to a few optimization functions. The analysed variant to EPSO comprised a new set of particles in global optimum. These were able to analyse the dimensional space in a more detailed way revealing the great efficiency of this approach. That it is evident that the Rosenbrock function needs to be complemented by something that enables the understanding of the way that dimensions correlate, therefore forming an algorithm that is able to understand in which dimension to act as well as the way others react to it.

The subsequent sections put forward a few complimentary tests. As such, the tests performed to the formation of a sub-swarm based in a random particle, as well as other alterations in the general configuration of satellite swarm, will be documented next.

4.4 Tests to the modified EPSO – Other configurations

4.4.1 Sub-swarm based on a random particle

At this stage, the focus lays on another variant proposal explored in point 4.2.2. The son-swarm is now not based on the position of the best current solution, but in the space zone occupied by a particle selected randomly from the total group of individuals. The remaining aspects keep the same parameters that were defined in the variant previously tested.

The below scheme comprises a system composed of the following characteristics:

- Sub-swarm located in the position of the current global best;
- This sub-swarm is confined to a search area up to 10% of the coordinates of the reference particle;
- Amount of particles of the mum-swarm: **20**;
- Amount of particles of the son-swarm: **20**;
- Progression of the son-swarm: textbf5 iterations + 5 (in case of improvement of the global optimum).

4.4.1.1 Rosenbrock function

considering that the previous variant revealed a great capacity in terms of the Alpine, Griewank and Ackley functions, one tried to understand its impact solely on the greatest complexity problem, meaning, the Rosenbrock model.

Having defined the main action aspects, the next step was to execute the trials. Similar to the previous procedure, a sub-swarm based on the position of one of its previously and randomly selected particles appears in **each of the 20 first iterations of the original swarm**.

The results of this variant's implementation are illustrated below:

Table 4.16: Rosenbrock's optimization tests, with original EPSO.

Test	No. Iterations	No. Evaluations
1	10028	401120
2	9926	397040
3	12277	491080
4	11418	456720
5	9951	398040
6	10968	438720
7	10218	408720
8	10297	411880
9	10633	425320
10	9738	389520
Average	10545.4	421816

Table 4.17: Rosenbrock's optimization tests, with variant to EPSO (based on a random particle).

Test	No. Iterations	No. Evaluations
1	10717	428880
2	10010	400600
3	9111	364800
4	10927	437400
5	11231	449440
6	9592	383920
7	10328	409880
8	8843	353720
9	10062	402680
10	8919	356960
Average	9974	398828

By comparing both tests' average values, one is able to denote their similarity, and therefore to conclude that the implementation of this variant to EPSO does not lead to a significant improvement of the involved computational effort. On the other hand, it would be interesting to understand and contrast the equivalence between both variants – the sub-swarm based on the best solution and the other based in a random position – as well as their implications on the *fitness* progression.

The following chart depicts curves related to the *fitness* value and its immediate computation, translated by the amount of performed evaluations. Both register and can be framed within the initial stage (first iterations) of the development of the EPSO model, so that each of the curves identifies the stated variants:

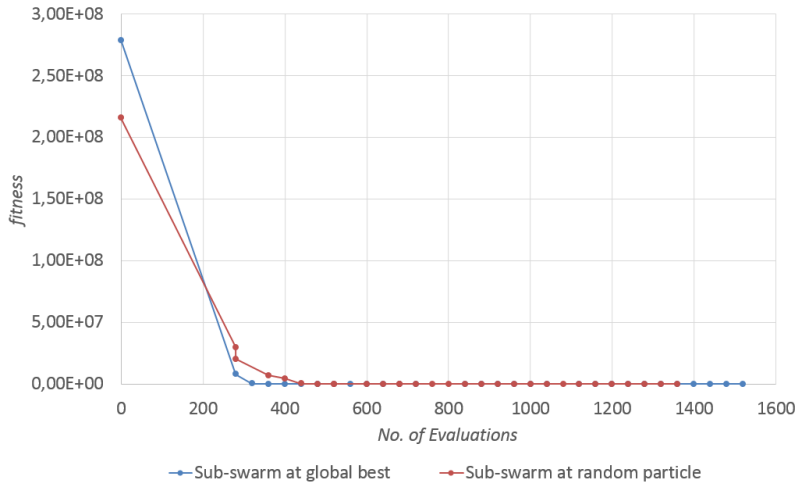

Figure 4.5: Comparison of the average progression, of the Rosenbrocks' *fitness* function, in both sub-swarms' variants of EPSO.

A modest difference between performances is clearly visible. The randomness attributed to the detailed search of the dimensional space is linked to the possibility of selecting particles that do not seem to be advantageous for the development of the mathematical model. On the other side, it is is in the current best solution zone that there is a higher probability of finding the solution. Nonetheless, the randomness that the first offers is often considered a good strategy in functions with great roughness, since it is able to investigate a different zone from the one the model is being attracted to, and where the problem's optimum solution may consequently be located.

As a finishing note, it is relevant to highlight the possibility of combining in the same EPSO model both the ability to generate two sub-swarms simultaneously. A few tests were executed for Rosenbrock and an undesired increase of the involved effort was verified. Hence, this EPSO's version not to be addressed in more detail in this thesis.

4.4.2 Increase of the amount of particles of the son-swarm

Another setting that was tested resided on **the amount of particles by which the son-swarm was composed**. By having a broader number of individuals, the search precision increased, so that it became interesting to understand if that increase would lead to a lower efficiency of the program or the opposite. Therefore, the following tests used the normal configurations of the son-swarm, based on the global optimum, only **differing by using 60 instead of 20 particles to compose the swarm**.

The scheme was the next:

- Sub-swarm located in the position of the current global best;
- This sub-swarm is confined to a search area up to 10% of the coordinates of the reference particle;
- Amount of particles of the mum-swarm: **20**;
- Amount of particles of the son-swarm: **60**;
- Progression of the son-swarm: **5 iterações + 5** (in case of improvement of the global optimum).

This configuration was implemented throughout the twenty (20) first program iterations of the Rosenbrock function.

Results were as follows:

Table 4.18: Rosenbrock's optimization tests, with variant to EPSO **(20 particles)**.

Test	No. Iterations	No. Evaluations
1	9008	360640
2	10666	426960
3	9104	364520
4	10645	426160
5	9052	362400
6	8264	330880
7	9291	371960
8	10435	417720
9	9707	388600
10	10354	414520
Average	9652.6	386436

Table 4.19: Rosenbrock's optimization tests, with variant to EPSO **(60 particles)**.

Test	No. Iterations	No. Evaluations
1	10739	430520
2	10799	432920
3	12251	491000
4	9971	399800
5	9135	366360
6	10329	414240
7	11387	456440
8	10913	437480
9	9666	387720
10	9624	386040
Average	10481.4	420252

Comparing the tables above, results do not seem to be very conclusive, due to the similarity of the registered convergence levels.

Let's turn attention to the following figure:

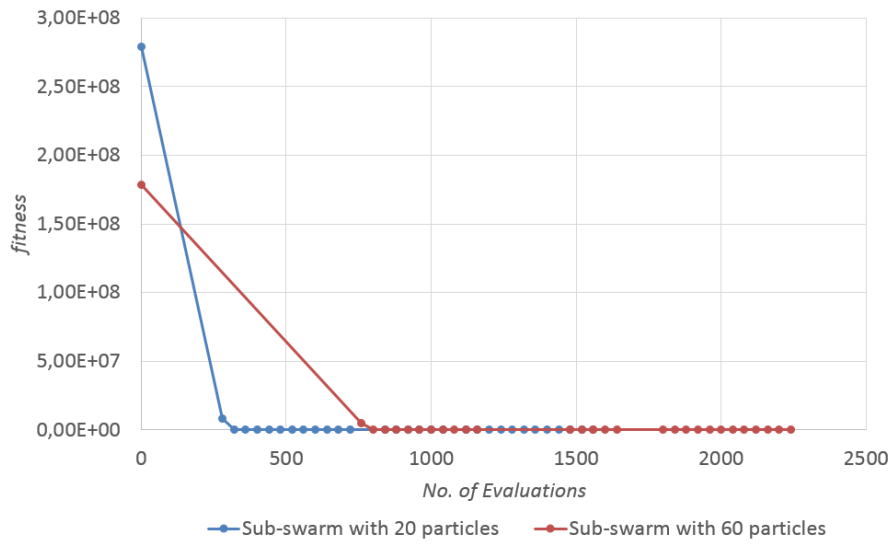

Figure 4.6: Comparison of the average progression of the Rosenbrock's function *fitness*, distinct amount of particles of the son-swarm.

Illustration 4.6 translates an important detail. One can notice that, in the second mark of each of the curves, the *fitness* level reached is practically the same. However, with regard to the set of 60 particles, this is linked to a much higher computational effort. One can subsequently conclude that the increase in particles forming the sub-swarm is not beneficial in this variant to EPSO.

4.4.3 Increase of the exploration limit of the son-swarm

Another point of analysis resided in the attempt of broadening the limits established for the sub-swarm's exploration. **Keeping the particle amount at 20 units, the imit of 10% were increased to 30%**. Theoretically, the broadening of a certain domain could lead to a slight improvement in the search of the optimum solution, due to the enlargement of space that swarm could analyze. Therefore, a set of trials using the configuration mentioned in the previous section (chapter 4.4.2) were performed on the Rosenbrock function, with the only difference that 30% were used instead of the 10%.

The results were as follows:

Table 4.20: Rosenbrock's optimization tests, with variant to EPSO **(limit at 10%)**.

Test	No. Iterations	No. Evaluations
1	9008	360640
2	10666	426960
3	9104	364520
4	10645	426160
5	9052	362400
6	8264	330880
7	9291	371960
8	10435	417720
9	9707	388600
10	10354	414520
Average	9652.6	386436

Table 4.21: Rosenbrock's optimization tests, with variant to EPSO **(limit at 30%)**.

Test	No. Iterations	No. Evaluations
1	9060	362760
2	9965	398960
3	8250	330360
4	10431	417640
5	10100	404400
6	10858	434640
7	9746	390200
8	8993	360080
9	9908	396560
10	10367	415040
Average	9767.8	391064

In the two tables above, it is illustrated that this new configuration does not bring about great changes with regard to the behaviour of this variant to EPSO. The feeling prevailed, that the particle amount in the sub-swarm did not produce great improvements in terms of reduction of the computational effort.

The following figure shows that, besides holding the same evaluations amount, *fitness* reveals similar progression levels:

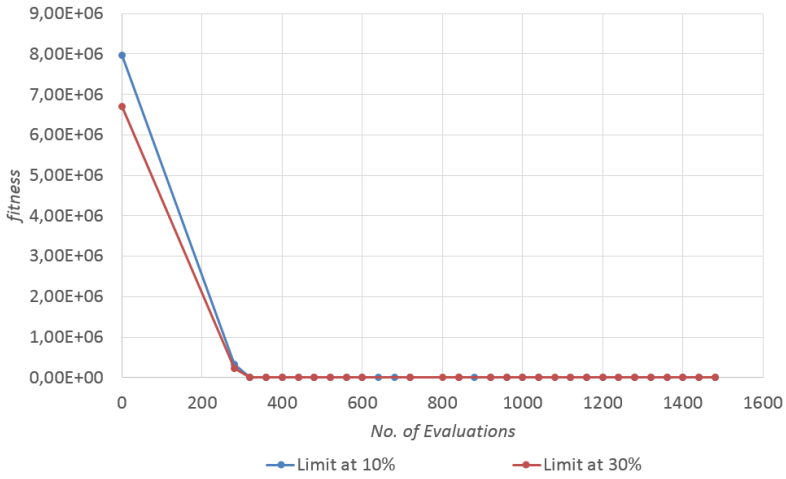

Figure 4.7: Comparison of the average progression of the Rosenbrock's function *fitness* for distinct limits on the son-swarm.

4.5 Main conclusions

It was this chapter's main objective to formulate another variant of the original EPSO model. Permanently striving for a reduction of the computational effort, the idea was to promote the creation of more precise particle groups within the original set.

Having defined the **mum-swarm**, the model of this variant to EPSO is inspired on the group comprising the particles which explore the domain of a given problem. Therefore, and as previously mentioned in this chapter, several successive sub-swarms appear. This could conduct to a better study of a more precise zone in the same space. Known as **son-swarm** or **sub-swarm**, these are confined to a smaller exploration area and were created in one of two positions, namely 1) based the position of the best current solution and the swarm is parametrized according to the position that present the best result of *fitness*, or 2) located in the position of a randomly selected particle.

A set of optimization tests were performed which minimized the functions by using this variant to EPSO. Starting from implementing a sub-swarm in the best particle, to its random selection, results were generally rather exciting, since, for the Alpine, Griewank and Ackley functions, the improvements in efficiency reached approximately **95%**. On the other hand, the Sphere function – due to its symmetry -, and the Rosenbrock function – due to its particular characteristics -, come about as a challenge that must be overcome with other approaches to be implemented into this variant.

Chapter 5

Electric power system application

EPSO has been subject to several tests and trials as far as energy systems are concerned, such as, for example, voltage/reactive control [24][23].

All work developed in the previous sections is put into practice in a real environment throughout this chapter. With that intent, an energy system with a real world identical topology is used. It was attempted to solve this problem by recurring to EPSO. With the implementations of the variants previously explained, the goal was understanding if the results obtained in optimization functions may be transposed to a typical EPS scenario.

5.1 EPS Presentation

5.1.1 Network's topology

The used model is based on a state estimation system. The system stands as an adaptation of the energy distribution in an average voltage network supported by the Benchmark Systems with European configuration [25]. Such a system comprises a tree-phase network with radial or mesh characteristics, adapting itself mainly to a city or to a rural consumption area.

The following study will be exclusively directed to the three-phase system, thus assuming its network symmetry and balance. Taking this into consideration, the line parameters considered are equivalent for all phases.

The figure below illustrates the studied energy system. There are several sources of energy throughout the network, which implies the construction of a topology with distributed production. One may identify these networks' production units at the nodes 1, 6 e 9

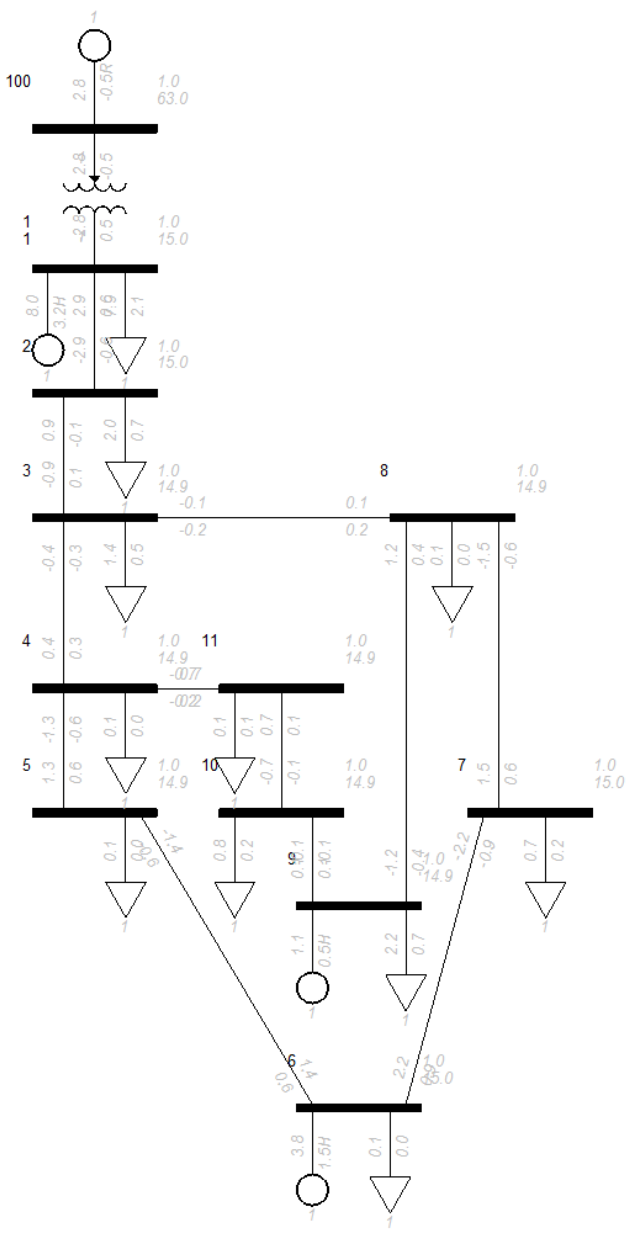

Figure 5.1: Topology of the studied EPS.

The description of the main parameters which identify the network in figure 5.1 are presented in the following section of this thesis.

5.1.2 Network's parameters

This section encompasses the main parameters which identify the network on figure 5.1. Assuming there is no need to detail all parameters, this paper restricts itself to main values such as lines, power factors, charging systems as well as power generated by the sources.

The bus 100 is interconnected to bus 1 through a 63/15 kV transformer with short-circuit impedance of 8% and nominal power of 25MVA.

The base power used, S_b, stood at 25 MVA and the voltage, U_b, stood at 15 kV for average voltage (nodes 1 to 11). The impedance which characterizes the lines, whether in the international system or in p.u. is illustrated below:

Table 5.1: Line Parameters.

Line	From	To	R (Ω)	X (Ω)	R (p.u.)	X (p.u.)
1	1	2	0.31640	0.35000	0.035155556	0.038888889
2	2	3	0.49720	0.55000	0.055244444	0.061111111
3	3	4	0.06780	0.07500	0.007533333	0.008333333
4	4	5	0.06780	0.07500	0.007533333	0.008333333
5	5	6	0.16950	0.18750	0.018833333	0.020833333
6	6	7	0.02260	0.02500	0.002511111	0.002777778
7	7	8	0.19210	0.21250	0.021344444	0.023611111
8	8	9	0.03390	0.03750	0.003766667	0.004166667
9	9	10	0.09040	0.10000	0.010044444	0.011111111
10	10	11	0.03390	0.03750	0.003766667	0.004166667
11	11	4	0.05650	0.06250	0.006277778	0.006944444
12	3	8	0.14690	0.16250	0.016322222	0.018055556

The maximum apparent charging powers for each node, as well as its respective voltage installation factor are depicted in the following table:

Table 5.2: Apparent power, S (kVA).

Bus	Residential	Commercial/Industrial
1	8700	4500
2		2500
3	185	1650
4	245	
5	250	
6	265	
7		900
8	305	
9		2750
10	290	800
11	170	

Table 5.3: Power factor.

Bus	Residential	Commercial/Industrial
1	0.98	0.95
2		0.95
3	0.98	0.95
4	0.98	
5	0.98	
6	0.98	
7		0.95
8	0.98	
9		0.95
10	0.98	0.95
11	0.98	

It is noteworthy that, along the network, the electric charges are divided into two distinct categories, namely "residential" charge and "commercial/industrial" charge.

Integrated distributed resources such as cogeneration sources, mini-hydric and solar photovoltaic are installed in buses 1, 6 and 9. The following table exposes their characteristic voltage values:

Table 5.4: Parameters of the DER units.

Bus	Technology	P (kW)	Q min. (kvar)	Q max (kvar)
1	cogeneration	8000	0	3200,0
6	mini-hydric	5000	0	2000,0
9	solar photovoltaic	3000	0	1200,0

Last, it is important to bear in mind that the charging value is not constant throughout the day. According to the time of the day, typical daily charge rates are registered. Therefore, table 5.5 features the charge rate simulated in this network topology, for a given day:

Table 5.5: Charge rate per hour 17.

Hour	Residential charge	Industrial charge	cogeneration	mini-hydric	solar photovoltaic
0	30,0%	10,0%	0,0%	30,0%	0,0%
1	27,5%	12,0%	0,0%	30,0%	0,0%
2	25,0%	12,0%	0,0%	30,0%	0,0%
3	22,5%	15,0%	0,0%	30,0%	0,0%
4	20,0%	15,0%	0,0%	30,0%	0,0%
5	20,0%	15,0%	0,0%	30,0%	0,0%
6	22,5%	18,0%	0,0%	30,0%	2,9%
7	30,0%	25,0%	0,0%	30,0%	14,9%
8	40,0%	50,0%	50,0%	75,0%	35,2%
9	43,0%	80,0%	50,0%	75,0%	54,8%
10	46,0%	100,0%	100,0%	75,0%	70,2%
11	50,0%	100,0%	100,0%	75,0%	81,4%
12	50,0%	90,0%	100,0%	75,0%	87,5%
13	55,0%	50,0%	75,0%	75,0%	88,6%
14	60,0%	50,0%	75,0%	75,0%	84,5%
15	60,0%	94,0%	100,0%	75,0%	75,6%
16	55,0%	90,0%	100,0%	75,0%	60,1%
17	50,0%	85,0%	100,0%	75,0%	38,2%
18	65,0%	70,0%	75,0%	75,0%	16,6%
19	85,0%	40,0%	50,0%	75,0%	3,5%
20	100,0%	30,0%	0,0%	75,0%	0,1%
21	90,0%	10,0%	0,0%	30,0%	0,0%
22	75,0%	10,0%	0,0%	30,0%	0,0%
23	55,0%	10,0%	0,0%	30,0%	0,0%

A measures' vector, which illustrated the expected values, was associated to this system. These values assumed that each bus holds injection voltage sensors and that nodes 2, 5 and 8 have voltage magnitude sensors.

This investigation work assumed the implementation of PMU (*phasor measurement unit*) sensors along the network. Its location was randomly decided to be on buses 2, 5, and 8 so that they did not only present injection current measures (module and phase) but also the corresponding voltage. Therefore, the injection power and voltage systems were replaced by PMU units, allowing the identification of these buses' injected voltage and current modules and phase.

Several tests to the diverse EPSO models are developed in the subsequent sections.

5.2 EPSO Application

In order to successfully apply EPSO to this problem, it was necessary to work on the objective function to which it was meant to be applied.With the assistance of external programs, a set of results of state estimation was used as reference for EPSO performances comparisons.

5.2.1 Objective function

The criteria used for purposes of optimization of this problem were the minimum squares, meaning the minimization of the sum of the residual squares, or the deviation from the expected value. Therefore, the minimization process was defined by the following equation 5.1:

$$min \sum (erro^2{}_i) \tag{5.1}$$

5.2.2 System restrictions

The balance of the energy system stood also as main point of focus. The restrictions imposed in the model meant to keeping the voltage levels and the angles' phase within acceptable limits. Therefore, the voltage and angles' operation limits were set in tables 5.2 e 5.3, respectively:

$$0.9 < U_i < 1.1 \tag{5.2}$$

$$-0.03 < \delta_i < 0.0 \tag{5.3}$$

5.2.3 Tests and results

The general EPSO model, whether in its original version or in its variant, was applied throughout the following sections. The main parameters defined are comprised in the below list:

- Number of particles: 20;
- Dimensions: 23 (11 angles e 12 voltages);
- Stopping criterion: 2000 iterations.

The stopping criterion is mentioned in the list above. **This chapter's main goal was to effectively understand the variants impact on the model's progression**. Thus, after **2000 iterations**, an analysis of the resulting *fitness* precision is executed.

Theoretically, the goal would be to achieve better results with the EPSO variants than the ones obtained through its original version (using the exact same number of iterations, 2000).

Evidently, a reliable global convergence of the system implies a higher iteration amount than the 2000. However the **main objective is to effectively discern the progressions, instead of**

precision, of the attained result. On the other hand, the time required by each iteration was significantly, so tests were short enough to allow a correct analysis and comparison of the results.

The following sections present the results of the tests applied to the aforementioned problem. It was attempted to evaluate and critique the numbers of the tested EPSO variant, always taking the values associated to the original EPSO into account.

5.2.3.1 Original EPSO version

The original version of EPSO was used for in solving of objective function of the system on figure 5.1. As previously stated, results were considered after 2000 iterations. According to a certain *fitness*, the value of the no. of iterations would be taken into consideration.

Hence, results were as follows:

Table 5.6: Results associated to the algorithm, with the original EPSO version.

No. of Iterations	No. of Evaluations	*Fitness*
2000	80000	0.057820762361

Table 5.7: Results of the states' estimation (2000 iterations), with the original EPSO version.

	Bus	Expected value	Value obtained
Pinj	100	0.112283	0.224988
	1	0.003941	0.009803
	2	-0.080607	0.103471
	3	-0.057295	-2.776706
	4	-0.004299	3.202282
	5	-0.004921	-0.430137
	6	0.145132	2.321324
	7	-0.029184	-2.746036
	8	-0.005973	1.126818
	9	-0.042984	-0.337011
	10	-0.031889	3.571634
	11	-0.003412	-4.09061

Table 5.8: (Continued) Results of the states' estimation (2000 iterations), with the original EPSO version.

	Bus	Expected value	Value obtained
Qinj	100	-0.023694	-0.071918
	1	0.045727	0.132034
	2	-0.026608	0.132516
	3	-0.018338	-2.62192
	4	-0.000970	2.741587
	5	-0.001026	-0.316112
	6	0.058854	2.629146
	7	-0.009395	-3.042322
	8	-0.001181	1.088041
	9	-0.010921	-0.244132
	10	-0.009837	3.957863
	11	-0.000684	-4.182143
teta	2	-0.012758	-0.029751
	5	-0.014999	-0.029981
	8	-0.015267	-0.03
V	2	0.997208	1.062785
	5	0.996683	1.0821
	8	0.995417	1.090428
Pij (p.u.)	3-4	0.016753	1.672014
	6-7	-0.088600	-2.333352
Qij (p.u.)	3-4	0.015972	1.596459
	6-7	-0.034914	-2.604319
I inj(real) (p.u.)	2	-0.080587	0.09234
	5	-0.004959	-0.388849
	8	-0.005914	1.004656
I inj(im) (p.u.)	2	-0.027634	0.123502
	5	-0.001109	-0.303749
	8	-0.001283	1.029239

Having obtained the values shown in tables 5.6 and 5.7, the following step was to define the comparison reference for the subsequent trial. A variant to EPSO will be applied for understanding its impact in the behaviour of its algorithm.

5.2.3.2 Variant to EPSO - SUBEPSO

This chapter encompasses the application of the second EPSO variant developed in chapter 4. Considering that the dimension of a particle was defined by voltages and angles, the first variant did not apply directly to this problem. This variant implies the analysis between dimensions and the respective variable change. Due to this there was a need to adapt that analysis between dimensions in order to assure the independence between both types.

The sub-swarm variant based on a global optimum was applied. As the iterations occur, more precise sub-swarms appear in space. The *swarm* spread limit was defined at 5% of the currently best position.

It is important to highlight that, in this phase, the similar initial situations to ones in previous tests were assured. Therefore, the initial particles' population are kept the identical in the following tests.

Further, the criteria set for the Rosenbrock optimization were also established. As we saw on above chapters, acting in initial phase of model's progression could be beneficial to its convergence. **Thus, this variant was applied to the first 20 iterations**.

The obtained results were as follows:

Table 5.9: Results associated to the algorithm, with EPSO variant (sub-swarm), in the 20 first iterations.

No. of Iterations	No. of Evaluations	*Fitness*
2000	83880	0.054612435172

Table 5.10: Results of the states' estimations (2000 iterations), with the original EPSO version.

	Bus	Expected value	Value obtained
Pinj	100	0.112283	0.170772
	1	0.003941	0.016489
	2	-0.080607	0.123328
	3	-0.057295	-1.913053
	4	-0.004299	1.052419
	5	-0.004921	0.780986
	6	0.145132	2.552114
	7	-0.029184	-3.049186
	8	-0.005973	0.798231
	9	-0.042984	-0.483296
	10	-0.031889	2.380087
	11	-0.003412	-2.317736

Table 5.11: (Continued) Results of states' estimation (2000 iterations), with the original EPSO version.

	Bus	Expected value	Value obtained
Qinj	100	-0.023694	0.02142
	1	0.045727	-0.093733
	2	-0.026608	0.221083
	3	-0.018338	-2.033188
	4	-0.000970	0.983376
	5	-0.001026	0.872867
	6	0.058854	2.77888
	7	-0.009395	-3.297445
	8	-0.001181	0.972222
	9	-0.010921	-0.534619
	10	-0.009837	2.681317
	11	-0.000684	-2.44726
teta	2	-0.012758	-0.028793
	5	-0.014999	-0.029779
	8	-0.015267	-0.03
V	2	0.997208	1.071505
	5	0.996683	1.089903
	8	0.995417	1.089293
Pij (p.u.)	3-4	0.016753	1.126833
	6-7	-0.088600	-2.781308
Qij (p.u.)	3-4	0.015972	1.273907
	6-7	-0.034914	-2.998226
I inj(real) (p.u.)	2	-0.080587	0.108645
	5	-0.004959	0.687699
	8	-0.005914	0.705695
I inj(im) (p.u.)	2	-0.027634	0.204028
	5	-0.001109	0.815375
	8	-0.001283	0.914104

A first analysis of the above results shows a clear reduction of the *fitness* value achieved through the implemented variant. By comparing tables 5.6 and 5.9 one is able to reach that exact conclusion. In fact, EPSO's variant allows the attainment of an inferior value of *fitness*. Thus, an improvement of **5.5%** of the *fitness* value is registered.

Moreover, there was the need to understand the variation of the involved computational effort in the previously stated evolution. As such, through the original EPSO version, it was endeavoured to obtain the necessary iterations amount for reaching the same fitness *fitness* as in table 5.9.

The following result is achieved by running the original EPSO:

Table 5.12: Results associated to the original EPSO algorithm, for a *fitness* value in the range of 0.05461.

No. of Iterations	No. of Evaluations	*Fitness*
2617	104760	0.0541617170252

The below figure portrays the program console while it runs:

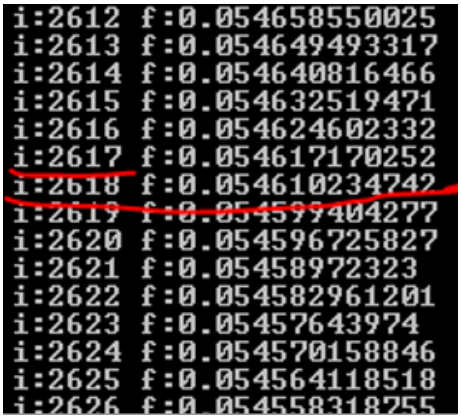

Figure 5.2: Program's console, with original EPSO.

Table 5.12 allows the a concluding remark. The original EPSO version reaches a similar range of *fitness* values as the ones achieved by this variant, only about **600 iterations later**.

The involved computational effort denotes a very significant increase, more specifically an **increase of approximately 2000 iterations**. This means that the tested EPSO variant manages a **reduction of 20%** of the evaluations done.

This variant's capacity in terms of the problem's convergence is very clear, through its application to the estimation of states in a typical electric power system.

It was also important to understand whether the criterion used in the application of this variant had a significant impact on the result. As we did in Rosenbrock function, **the sub-swarm strategy is now apply in all the iterations of the main swarm**. Subsequently, in each new generation of voltage and angles values, a new swarm was generated, within a "radium" up to 5% around the best position.

Running the tests, the following results were obtained:

Table 5.13: Results associated to the algorithm, with EPSO's variant (sub-swarms), in all iterations.

No. of Iterations	No. of Evaluations	*Fitness*
2000	480080	0.054612435172

Before analysing the results depicted by the table above, it is important to recall that the behaviour of the EPSO variant, applied to the 20 first iterations (illustrated in table 5.9). The *fitness* result attained by both methods is exactly the same. This leads to the conclusion that the **sub-swarms promoted after the 20th iteration do not have any effect**.

However, in terms of the evaluations amount, one is able to note a value much higher than the one registered in table 5.9. From processing the variant in all iterations to doing it solely on the 20 first, a reduction of **82.5%** of the associated *fitness* evaluations is achieved. It's justified by the fact that new sub-swarms being created in all iterations. Each of these develop for at least 5 iterations, which consequently increases the evaluation amount executed by the program.

Considering the very significant amount (480080), it is obvious that using a more precise search strategy has a beneficial impact on the initial development stage of the algorithm, instead of during its progression as a whole.

5.2.3.3 SUBEPSO - Tests with different initializations

The acting matrix for the solution of this problem is finally defined. The next logical step was to proceed to the application of a broader set of trials with distinct and aleatory initializations. Therefore, all trials were executed in an independent and aleatory manner.

The goal was, in fact, to understand if the aforementioned impact was transportable to other tests. Firsty, we assured the general similarity of both circumstances and network configurations. Secondly, a series of tests to the original and for the variant of EPSO were put into practice. Due to the the slowness of these tests, the **stopping criterion was defined as 500 iterations**.

Further, the evaluation amount and the function's resulting *fitness* value were recorded. It was expectable that, as previously seen, SUBEPSO would be able to accelerate progression with the same amount of iterations. Therefore we expected to reach lower *fitness* values than the original version.

As far as results are concerned, the following *fitness* values were registered:

Table 5.14: Results with the original EPSO, 500 iterations.

	Test 1	Test 2	Test 3	Test 4	Test 5	**Average**
fitness	0.0804855	0.0621395	0.0664783	0.16282	0.1035848	**0.095102**
Num. of evaluations	20000	20000	20000	20000	20000	**20000**

Table 5.14 depicts the magnitude range of the values obtained through the 500 iterations of the aforementioned problem.

Subsequently, the creation of sub-swarms at the global optimum of each iteration is implemented, more specifically for the first 20 iterations. Each develops over 5 iterations, to which another 5 are performed in the case of detecting a better overall optimum for.

Therefore, results were as follows:

Table 5.15: Results with EPSO variant (SUBEPSO), 500 iterations.

	Test 1	Test 2	Test 3	Test 4	Test 5	**Average**
fitness	0.0675741	0.0426695	0.0645503	0.0996076	0.0430762	**0.063496**
Num. of evaluations	24960	24800	24840	25200	25800	**25120**

Table 5.15 clearly illustrates that the average of the evaluations amount, in five trials, is approximately 25 000 units – an increase of 25% when compared to the value registered in table 5.14. Nonetheless, variant to EPSO shows a **decrease of 33%** of the *fitness* value for the same iterations (500).

The application of variant to EPSO fostered alterations in two distinct aspects. First, the natural increase of the associated evaluation amount, through the creation of these swarms and their development. Second, the reduction of the *fitness* value for the problem's optimization.

It is evident that this variant has a direct impact on the acceleration of the progression of this optimization model. The question remained if this increase in the evaluation amount is compensated by the optimization levels achieved. As demonstrated in the table 5.12 above, the original EPSO reached these *fitness* values for a much higher iteration amount, requiring an extra computational effort linked to the additional evaluations.

This leads us to the next conclusion. The proposed variant shows results which foster the algorithm's acceleration. Although 500 iterations are little relevant in terms of the resolution of a problem with acceptable values, the presented tests stimulate a belief. Even at an initial stage of the model's progression, the implementation of this variant can stand out when compared to the results previously presented with original EPSO.

The values of the states' estimation achieved along the realization of the trials was also a concern. An average of the estimations for the original EPSO and its variant was calculated. These values allow a different view than the *fitness*, since one is able to compare the resulting values with the ones expected. Furthermore, they enable the calculation of a general average of the error associated to each of them.

Hence, results were as follows:

Table 5.16: Average results of the states estimation (500 iterations), for both EPSO versions.

	Bus	Expected value	Estimation (Original EPSO)	Estimation (Variant to EPSO)
Pinj	100	0.08910	0.0891046	0.070266
	1	0.01751	0.0175084	0.061099
	2	-0.17825	-0.1782528	-0.252172
	3	0.82182	0.8218242	-0.728709
	4	-1.87733	-1.8773304	2.428942
	6	1.49151	1.4915102	0.487478
	7	-0.029184	-0.6673564	0.426413
	8	-0.005973	0.6070634	-0.152635
	9	-0.042984	-0.9594	-0.325402
	10	-0.031889	0.6717898	0.319699
	11	-0.003412	0.1196432	-0.561263
Qinj	100	-0.023694	0.1597206	0.117805
	1	0.045727	-0.041767	0.029625
	2	-0.026608	-0.1816716	-0.300902
	3	-0.018338	1.1196568	-0.736406
	4	-0.000970	-2.5967068	2.733043
	5	-0.001026	0.5275438	-1.563843
	6	0.058854	1.5628548	0.503679
	7	-0.009395	-0.7666096	0.398718
	8	-0.001181	0.458133	-0.312926
	9	-0.010921	-0.9729418	-0.343537
	10	-0.009837	0.8004516	0.705075
	11	-0.000684	0.4836442	-0.907468

Table 5.17: (Continued) Results of states estimation (500 iterations), for both EPSO versions.

	Bus	Expected value	Estimation (Original EPSO)	Estimation (Variant to EPSO)
teta	2	-0.012758	-0.007568	-0.007526
	5	-0.014999	-0.0053066	-0.005803
	8	-0.015267	-0.0072226	-0.007006
V	2	0.997208	0.9643994	0.959109
	5	0.996683	0.9808246	0.976313
	8	0.995417	0.9572476	0.959974
Pij (p.u.)	3-4	0.016753	-0.747992	0.796111
	6-7	-0.088600	-1.0163954	0.066898
Qij (p.u.)	3-4	0.015972	-0.9759548	0.886734
	6-7	-0.034914	-1.112069	0.031459
I inj (real) (p.u.)	2	-0.080587	-0.1962468	-0.266069
	5	-0.004959	0.3368602	-1.624804
	8	-0.005914	0.6176518	-0.170746
I inj (im) (p.u.)	2	-0.027634	-0.2025372	-0.320361
	5	-0.001109	0.5127934	-1.743194
	8	-0.001283	0.468867	-0.352486

The values registered in these tables conducts to the calculation of the total average deviation error associated. In the table below, are presented to both optimization methods:

Table 5.18: Average total deviation, for the two versions of EPSO, after 500 iterations.

	Total average deviation
Original EPSO	0.584063513
Variant EPSO	0.547556223

The table 5.18 depicts a summary of the two last paragraphs. After 500 iterations, variant to EPSO extracts an average deviation inferior to its original version. In fact, it is able to optimize the error by **6,3%**.

The variant's impact at the initial phase of this problem's is visible. Actually, it promotes a faster fitness progression, fostering an increase of the computational efficiency.

During the running of the previous tests, the *fitness* value of the 30 first iterations of each trial was extracted. These values allow the construction of a graphic view of the difference between the average progression of both EPSO versions.

As such, the below chart illustrates the curves related both to the iterations' amount and the respective *fitness* value.

Take in consideration the following chart:

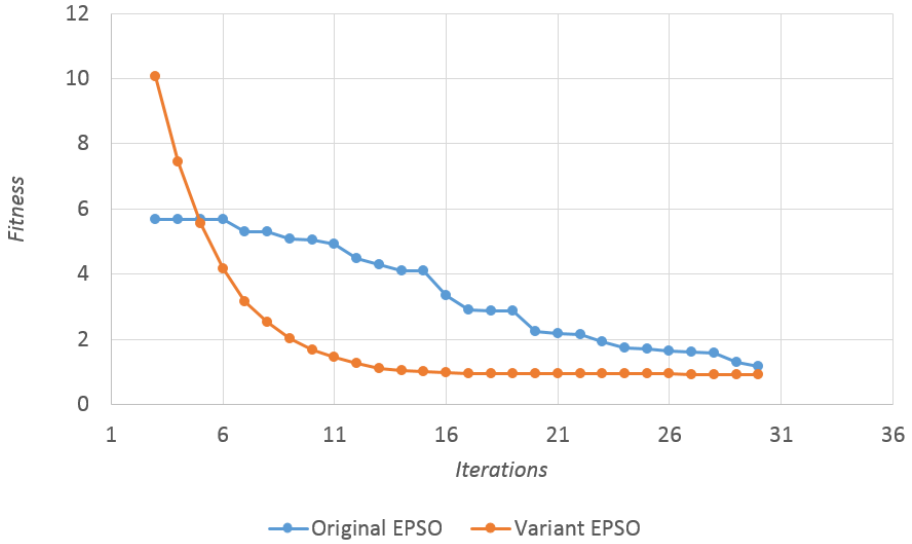

Figure 5.3: Contrast of the *fitness* progressions on both EPSO versions, for the 30 first iterations.

The work developed in this chapter is represented in figure 5.3[1] The curve associated to this EPSO variant depicts an abrupt decrease of the *fitness* value, while the curve associated to the original model presents a "stair" characteristic, due the slower and more gradual decrease of the objective function.

Other important aspect is visible especially on its 16th iteration, since the progression of the variant's curve reaches a phase where it takes place in a much slower manner. This suggests that the implementation might not hold a big impact and must therefore be uncoupled from EPSO. The strategy of applying sub-swarms at the 20 first iterations is considered beneficial, not only in terms of acceleration of the model, but also as far as the subsequent efficiency.

[1]The two first iterations are not presented since, at a randomly created initial population, the value of the objective function is significantly superior to the following iterations. Its inclusion would consequently lead to a significant distortion in terms of scale, thus rendering the resulting graph imperceptible.

5.3 Main conclusions

An important conclusion might be drawn from this chapter. SUBEPSO shows potential to get some improvements from the original EPSO model. Its application to problems in energy systems done in this chapter is just an example of that.

A series of trials and comparisons of the results among original EPSO and the variant were put forward. Whether holding identical conditions between the two (chapters 5.2.3.1 e 5.2.3.2) or in aleatory and distinct conditions (chapter 5.2.3.3), one is able perceive the good performance achieved by the variant to EPSO. In fact, results always depicted the supremacy of the computational efficiency it reached, by simultaneously needing less iterations to achieve a certain *fitness* level. Despite the extra computation required for the creation and progression of satellite swarms, it promotes the acceleration of *fitness* progression where the iterations are spared.

Also noteworthy was the criterion used at the variant's implementation. In fact, the approach used for the 20 first iterations can be considered as the best strategy mainly due to the possibility of all iterations. SUBEPSO revealed a beneficial impact at the initial stage of the development process, since it is able to indicate the zone of the dimensional space to which the program must progress. Nonetheless, for smaller values of *fitness*, when the program does not progress in a significant way, its presence does not reveal great impact, so that it may be considered unnecessary.

There are numerous challenges that can be put in question with EPSO. However, SUBEPSO revealed, until now, great potential at optimization processes. It had contributed to computational efforts reductions, without damaging the quality and reliability of its results.

Chapter 6

Conclusions and future work

Throughout the several months that comprised the preparation of this study through multiple readings, ideas development, execution of tests and trials, analysis and critics of the respective results, the main goal of this thesis was to attempt to develop and, if possible, to improve the EPSO model. This powerful optimization tool, inspired biology and the observation of nature, opens a door to a reality that was considered a challenge until not so long ago, namely the evolutionary computation.

Having accomplished a general framing of this investigation work in chapter 2, the moment of reflexion and analysis arrives, when one attempts to understanding whether the goals proposed in this thesis were successfully achieved or not, and which future challenges may arise from this work.

6.1 Goals achieved

Two general variants of the original EPSO model were presented. Supported by a previous theoretical study, each of them was subject to testing, in an attempt to understand if the results were effectively better than those attained with the original model. It is important to highlight that not only the iterations' amount comprised by the process, but also the amount of *fitness* evaluations required, translate to the user's computational effort inherent until their convergence, according to a certain stopping criterion.

Chapter 3 compasses the first variant proposal, named VAREPSO. Inspired on the idea of a change of variable targeting the reorganization of a swarm's disposition. From iteration to iteration, it was subject to the analysis of the way each of dimensions was disposed in space. According to a criterion, each of the particles was reorganized for each of the dimensions that revealed comparatively superior magnitudes. For the purpose of this thesis, it was defined that action would take place in case a dimension presented double the magnitude of another.

This idea arose during the observation of the way a swarm interpreted an optimization problem. In fact, it was clear that at certain moments, the swarm spread through space, promoting situations with a great disparity among dimensions was registered. Theoretically, one attempted to prove

that a reduction of these differences could lead to an improvement of the process. By accelerating its convergence it could results in a reduction of the involved computation.

A series of tests were executed in optimization functions, which revealed this variant's potential especially with regard to the Alpine, Griewank and Ackley functions. It registered improvements of over 90% of the computational effort, while the Sphere and the Rosenbrock functions, for the previously stated reasons, did not register a significant impact. Although this variant is supported by a well-established theoretical basis. There's a need of readaptation the objective functions, especially in terms of the modifications that take place in the particles' position, which leads to slower processes and jeopardizes practicality of this variant.

Chapter 4 addressed the second EPSO variant. SUBEPSO arises from an idea which can be summarized through a simple question: *"Why not having an EPSO within EPSO itself?"*. Despite the apparent incoherence of this question, it does make sense. Considering a general swarm known as mum-swarm, one could establish small particle groups parametrized in the best way. This ideology allowed that certain space zones were carefully analysed by these groups, named son-swarms. Whether in the position of the best current particle or in the position of a randomly selected individual, the goal was to deepen the search in the most favourable zone at that moment, or to proceed and search in other areas of the dimensional space that could prevent possible "jams" in optimum places, respectively.

Similar to what was done in the first variant, this was put into practice trough the execution of several optimization tests. With original EPSO version results as a reference, these trials allowed the collection of data. Several configurations were successively being tested, in order to reach a good parametrization of variant's acting model. As a summary, one was able to identify the great capacity SUBEPSO revealed in its version based on the global optima. In truth, it leads to improvements in the range of 96% in terms of computation for some of the optimization functions.

The versatility associated to its application, since it was only necessary to parametrize the way this variant was implemented, reveals a great potential as far as the goal of this dissertation is concerned. This variant presents as a possible answer to original EPSO model optimization in terms of the results' speed, effort and robustness.

Finally, the last section of this thesis (chapter 5) comprises the attempt to apply part of what was presented and developed into a real energy system. Steps such as defining the problem, implementing the EPSO variant, executing it and comparing the performance with the original version are addressed in this chapter.

The problem analysis consisted of a state estimation based on the European Benchmark Systems configuration. The optimization criterion was defined through the minimization of the sum of the square deviations to an expected value. The results attained from the application of the SUBEPSO variant were rather exciting. Resulting *fitness* values were inferior to the ones associated to the original EPSO, since the latter required more iterations for reaching the same range of magnitude as the first. These results illustrate SUBEPSO's potential for optimization functions, due to its clear capacity of accelerating EPSOS's progression and requires less evaluations and iterations. Subsequently it contributed for a relevant reduction of the involved computation effort.

As a summary, two variants arise in the attempt to launch a new paradigm in the scope of evolutionary computation, and especially in the scope of EPSO. The constant strive for increasing the models' efficiency, whilst keeping their robustness, was the main scope of this thesis.

6.2 Future investigation work

As stated in the motivational aspects of this thesis, the constant search for new developments, especially with regard to improving the efficiency of the existing models, stands as the ultimate goal which may never be fully achieved.

It is urgent to develop the ideas exposed in this paper in an even deeper way. The analysis, development, implementation and testing of alternative variants are natural challenges that emerges and this thesis attempted to overcome. The investigation work exposed here may serve as an inspiration and as starting point for future developments, not only as far as the general EPSO model is concerned, but also in all areas of evolutionary computation.

As such, it may be interesting to redefine the action approaches of the first variant (VAREPSO), whilst keeping the theoretical basis detailed here, turn it into a model with greater action capacity of objective functions, and especially to turn it into a more practical algorithm.

Moreover, despite the considerable development achieved by the SUBEPSO variant, another aspect that may be subject to future investigations lies on the increase of its intelligence and autonomy. It may allow the algorithm to understand the way it must face a certain problem, and which parametrization it must opt for, becoming completely independent from its user.

There is other project to be done in future. A compilation of both variants could be a good strategy. In fact, applying dimensions re-scaling on sub-swarms may conduct to a faster process of convergence, where dimensions interpretations during the spreading process are done to son-swarm.

References

[1] Vladimiro Miranda and Nuno Fonseca. "EPSO-best-of-two-worlds meta-heuristic applied to power system problems". In *Proceedings of WCCI/CEC – World Conference on Computational Intelligence, Conference on Evolutionary Computation*, volume 2, pages 1080–1085. Honolulu (Hawaii), USA, May 2002. doi:10.1109/CEC.2002.1004393.

[2] Vladimiro Miranda and Nuno Fonseca. "EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems". In *Proc. of the Asia Pacific IEEE/PES Transmission and Distribution Conference and Exhibition*, volume 2, pages 745–750. Yokohama, Japan, Oct 2002. doi:10.1109/TDC.2002.1177567.

[3] Vladimiro Miranda. "Computação Evolucionária Fenotípica". *Notes from lessons, version*, 2, 2005.

[4] Vladimiro Miranda. "Evolutionary algorithms with particle swarm movements". In *Intelligent Systems Application to Power Systems 2005. Proceedings of the 13th International Conference on*, pages 6–21. Arlington, VA, IEEE, Nov 2005. doi:10.1109/ISAP.2005.1599236.

[5] Cristina Andreia Araújo Cerqueira. "Estudo de variantes de Optimização por Enxames Evolucionários de Partículas (EPSO) e o seu comportamento num problema real de previsão de potência de um parque eólico". Technical report, INESC Porto, Oct 2005.

[6] EPSO code c++, 2009. URL: http://epso.inescporto.pt/epso-code-c.

[7] Vladimiro Miranda and Nuno Fonseca. "New evolutionary particle swarm algorithm (EPSO) applied to voltage/VAR control". In *Proceedings of PSCC'02 – Power System Computation Conference*. Sevilla, Spain, Jun 24-28 2002.

[8] Hans-Georg Beyer. "Evolution strategies". *Scholarpedia*, 2(8):1965, 2007.

[9] H Mori and Y Komatsu. "A hybrid method of optimal data mining and artificial neural network for voltage stability assessment". IEEE St. Petersburg, Power Tech, Russia, Jun 2005. doi:10.1109/PTC.2005.4524377.

[10] Vladimiro Miranda, Hrvoje Keko, and Alvaro Jaramillo Duque. "Stochastic star communication topology in evolutionary particle swarms (EPSO)". *International journal of computational intelligence research*, 4(2):105–116, Jan 2008. doi:10.1109/TDC.2002.1177567.

[11] Mehdi Eghbal, E.E.El-Araby, Naoto Yorino and Yoshifumi Zoka. "Application of Meta-heuristic Methods to Reactive Power Planning: A Comparative Study for GA, PSO and EPSO". *Proceedings - IEEE International Conference on Systems, Man and Cybernetics 2007*, pp. 3755-3760. Montreal (Quebec), Canada, Oct 7-10 2007.

[12] D. Midence and A. Vargas. "Estudio Comparativo de Algoritmos de Computación Evolutiva en la Optimización de la Confiabilidad en Redes de Distribución de Potencia". *XIII ERIAC - Décimo tercer Encuentro Regional iberoamericano de CIGRÉ*, comité C4. Puerto Iguazú, Argentina, May 2009.

[13] Daniel L. Souza, Otavio N. Teixeira, Dionne Monteiro, Roberto C. Limao de Oliveira. "A CUDA-Based Cooperative Evolutionary Multi-Swarm Optimization Applied to Engineering Problems". *CSBC 2012*. Curitiba, Brasil, Jul 2012.

[14] James Kennedy. "Particle swarm optimization". In *Encyclopedia of machine learning*, pages 760–766. Springer US, 2011. doi:10.1007/978-0-387-30164-8_630.

[15] Morten Løvbjerg, Thomas Kiel Rasmussen, and Thiemo Krink. "Hybrid Particle Swarm Optimiser with Breeding and Subpopulations". In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 469–476. Morgan Kaufmann, 2001.

[16] Vladimiro Miranda and Naing Win-Oo. "New experiments with EPSO — Evolutionary particle swarm optimization". In *Proc. IEEE Swarm Intell. Symp*, pages 162–169. Indianápolis, Indiana, USA, May 2006.

[17] Rainer Storn and Kenneth Price. "Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces". volume 3. ICSI Berkeley, 1995.

[18] Rainer Storn and Kenneth Price. "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces". *Journal of global optimization*, 11(4):341–359, Dec 1997. doi:10.1023/A:1008202821328.

[19] Vladimiro Miranda and Renan Alves. "Differential Evolutionary Particle Swarm Optimization (DEEPSO): A Successful Hybrid". In *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC), 2013 BRICS Congress on*, pages 368–374. Ipojuca, IEEE, Sept 2013. doi:10.1109/BRICS-CCI-CBIC.2013.68.

[20] Vladimiro Miranda and Renan Alves. "PAR/PST location and sizing in power grids with wind power uncertainty". In *Probabilistic Methods Applied to Power Systems (PMAPS), 2014 International Conference on*, pages 1–6. Durham, IEEE, July 2014. doi:10.1109/PMAPS.2014.6960679.

[21] Vladimiro Miranda, Leonel De Magalhães Carvalho, Mauro Augusto Da Rosa, Armando M Leite Da Silva, and Chanan Singh. "Improving power system reliability calculation efficiency with EPSO variants". *Power Systems, IEEE Transactions on*, 24(4), Oct 2009. doi:10.1109/TPWRS.2009.2030397.

[22] Hrvoje Keko, A Jaramillo Duque, and Vladimiro Miranda. "A multiple scenario security constrained reactive power planning tool using EPSO". In *Intelligent Systems Applications to Power Systems 2007. ISAP 2007. International Conference on*, pages 1–6. IEEE, Nov 2007. doi:10.1109/ISAP.2007.4441589.

[23] Manuel A Matos, M Teresa Ponce de Leão, J Tomé Saraiva, J Nuno Fidalgo, Vladimiro Miranda, J Peças Lopes, J Rui Ferreira, Jorge MC Pereira, L Miguel Proença, and J Luís Pinto. "Metaheuristics applied to power systems". In *Metaheuristics: computer decision-making*, pages 449–464. 2004. doi:10.1007/978-1-4757-4137-7_21.

[24] Jorge Pereira, J Tomé Saraiva, and MT Ponce de Leao. "Identification of operation strategies of distribution networks using a simulated annealing approach". In *Electric Power Engineering, 1999. PowerTech Budapest 99. International Conference on*, page 42. Budapest, Hungary, USA, Sept, 1999. doi:10.1109/PTC.1999.826473.

[25] Kai Strunz, N Hatziargyriou, and C Andrieu. "Benchmark systems for network integration of renewable and distributed energy resources". *Cigre Task Force C*, 6:04–02, 2009.

Appendix A

Annex A - Optimization Functions - *Fitness* progression with VAREPSO

In these following sections, will be presented the average of *fitness* values (five tests) of the initial part of progressions using the optimization functions.

Table A.1: Average *fitness* progression for Rosenbrock.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	VAREPSO Average Fitness
0	0	194172718,8	175235160,1
1	40	194172718,8	175235160,1
2	80	125539476,4	62381402,47
3	120	78365804,98	44449139,38
4	160	59783095,35	27427519,22
5	200	42762980,4	14340051,38
6	240	34816560,4	9088186,964
7	280	32708154,25	5863137,478
8	320	29884871,31	1879691,199
9	360	25943177,19	831068,2099
10	400	24178101,11	254751,3201
11	440	18596443,67	37686,17977
12	480	16797845,86	6190,346801
13	520	15741977,11	1579,36143
14	560	11139653,32	144,9671026
15	600	6976714,288	69,6317534
16	640	4818929,213	32,9558982
17	680	4320670,884	29,8847834
18	720	3279333,209	29,5427166
19	760	2694541,44	29,0362482
20	800	2546905,833	28,988712
21	840	2001177,702	28,9757474
22	880	1382358,494	28,9617194
23	920	1097731,11	28,9419284

Table A.2: Average *fitness* progression for Spheric.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	VAREPSO Average Fitness
0	0	17223,27646	17223,27646
1	40	17223,27646	17223,27646
2	80	17223,27646	17223,27646
3	120	16100,00171	16100,00171
4	160	12310,65359	12310,65359
5	200	12310,65359	12310,65359
6	240	10518,79736	10518,79736
7	280	8691,710498	8691,710498
8	320	8639,879179	8639,879179
9	360	6747,377597	6747,313562
10	400	6747,377597	6106,220893
11	440	6320,901054	6106,220893
12	480	6060,184128	4390,520685
13	520	5879,330911	4390,520685
14	560	5315,170931	4090,297823
15	600	4866,479767	3609,547496
16	640	4834,9592	3504,88353
17	680	4834,9592	1796,905427
18	720	4655,648964	1796,905427
19	760	4581,521126	1682,826226
20	800	4506,962654	1200,260072
21	840	4436,382028	1200,260072
22	880	4111,368266	1200,260072
23	920	3899,462983	1200,260072
24	960	3860,598106	1200,260072
25	1000	3844,797482	1200,260072
26	1040	3835,045357	1078,454911
27	1080	3828,517937	1078,454911
28	1120	3824,120274	1078,454911
29	1160	3715,404036	979,974919

Table A.3: Average *fitness* progression for Alpine.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	VAREPSO Average Fitness
0	0	35,9538862	32,8970778
1	40	35,9538862	32,8970778
2	80	34,4221108	32,442835
3	120	31,3321966	30,7421684
4	160	30,7503036	29,0792442
5	200	29,9556016	23,693867
6	240	28,3727462	22,569063
7	280	27,4871988	21,0586482
8	320	25,6444882	16,8099998
9	360	23,9107704	13,3329394
10	400	22,845699	11,0543674
11	440	22,6000922	7,1996172
12	480	21,5039254	6,3382584
13	520	21,0990208	5,9749364
14	560	19,9881314	4,858454
15	600	19,5507956	3,7792054
16	640	18,4186326	2,306636
17	680	17,209605	1,781897
18	720	16,8688824	1,6768786
19	760	16,288402	1,4389352
20	800	15,8724836	1,3008588
21	840	15,7278066	1,2069858
22	880	15,2454104	0,721425
23	920	14,6827308	0,6782828
24	960	14,4760778	0,657872
25	1000	14,4142694	0,2001692
26	1040	14,2229088	0,1775634
27	1080	13,3263392	0,1720378
28	1120	12,7652472	0,170965
29	1160	12,4476104	0,1677704

Table A.4: Average *fitness* progression for Griewank.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	VAREPSO Average Fitness
0	0	2,1272156	2,1941718
1	40	2,1272156	2,1941718
2	80	2,0056922	2,010798
3	120	1,785832	1,7177346
4	160	1,6854152	1,3155208
5	200	1,6150012	1,2029956
6	240	1,525274	1,177816
7	280	1,4788814	1,1200284
8	320	1,3953214	1,069447
9	360	1,3483064	1,0442044
10	400	1,3270264	0,9032702
11	440	1,3069126	0,6690308
12	480	1,2774172	0,4496994
13	520	1,257716	0,3792134
14	560	1,227376	0,1641602
15	600	1,2058306	0,043274
16	640	1,1872488	0,01223675
17	680	1,1748338	0,003065
18	720	1,1610396	0,00046225
19	760	1,1513324	0,000148333
20	800	1,1312092	0,00011
21	840	1,1170004	
22	880	1,1069902	
23	920	1,1010632	
24	960	1,0960282	
25	1000	1,0918178	
26	1040	1,0878764	
27	1080	1,083724	
28	1120	1,0815064	
29	1160	1,0769866	

Table A.5: Average *fitness* progression for Ackley.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	VAREPSO Average Fitness
0	0	17,6581796	17,6367304
1	40	17,6581796	17,6367304
2	80	16,9796888	17,0942466
3	120	16,1508206	15,934814
4	160	15,7476316	14,197003
5	200	15,2542944	13,0156614
6	240	14,7683638	12,0668164
7	280	14,4361982	10,1018782
8	320	13,8733618	8,1685652
9	360	13,1516598	4,1585082
10	400	12,7917364	3,65422
11	440	12,588249	3,097701
12	480	12,4302636	1,9377384
13	520	12,1814044	1,3048564
14	560	11,8506328	0,6693416
15	600	11,612533	0,4096524
16	640	11,2926176	0,2282024
17	680	11,1646564	0,1118778
18	720	10,6760234	0,0488508
19	760	10,354053	0,0225648
20	800	10,147467	0,009278333
21	840	9,8636312	0,001961333
22	880	9,7297268	0,000653
23	920	9,4800032	0,000653
24	960	9,3187768	0,000546
25	1000	9,1532018	
26	1040	8,8902258	
27	1080	8,7909076	
28	1120	8,6948526	
29	1160	8,417546	

Appendix B

Annex B - Optimization Functions - *Fitness* progression with SUBEPSO

In these following sections, will be presented the average of *fitness* values (five tests) of the initial part of progressions using the optimization functions. SUBEPSO based on optimal best particle.

Table B.1: Average *fitness* progression for Rosenbrock.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	SUBEPSO No. of Evaluations	SUBEPSO Average Fitness
0	0	194172718,8	0	279036332,5
1	40	194172718,8	280	7974098,516
2	80	125539476,4	320	331583,2755
3	120	78365804,98	360	8028,752884
4	160	59783095,35	400	466,0383082
5	200	42762980,4	440	72,6450506
6	240	34816560,4	480	35,4426788
7	280	32708154,25	520	30,1486646
8	320	29884871,31	560	29,1914092
9	360	25943177,19	600	29,004002
10	400	24178101,11	640	28,9734672
11	440	18596443,67	680	28,911961
12	480	16797845,86	720	28,869315
13	520	15741977,11	880	28,8424998
14	560	11139653,32	920	28,815715
15	600	6976714,288	840	28,7939284
16	640	4818929,213	960	28,7708542
17	680	4320670,884	1000	28,752504
18	720	3279333,209	1040	28,7354482
19	760	2694541,44	1120	28,7080584
20	800	2546905,833	1160	28,692316
21	840	2001177,702	1200	28,6770596
22	880	1382358,494	1240	28,6677622
23	920	1097731,11	1280	28,6575658

Table B.2: Average *fitness* progression for Alpine.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	SUBEPSO No. of Evaluations	SUBEPSO Average Fitness
0	0	35,9538862	0	32,430227
1	40	35,9538862	280	12,9475694
2	80	34,4221108	320	4,2492772
3	120	31,3321966	360	0,9384506
4	160	30,7503036	400	0,1371006
5	200	29,9556016	440	0,0312368
6	240	28,3727462	480	0,0098194
7	280	27,4871988	520	0,0037884
8	320	25,6444882	560	0,0014376
9	360	23,9107704	600	0,0004204
10	400	22,845699	640	0,000188
11	440	22,6000922		
12	480	21,5039254		
13	520	21,0990208		
14	560	19,9881314		
15	600	19,5507956		
16	640	18,4186326		
17	680	17,209605		
18	720	16,8688824		
19	760	16,288402		
20	800	15,8724836		
21	840	15,7278066		
22	880	15,2454104		
23	920	14,6827308		
24	960	14,4760778		
25	1000	14,4142694		
26	1040	14,2229088		
27	1080	13,3263392		
28	1120	12,7652472		
29	1160	12,4476104		

Table B.3: Average *fitness* progression for Griewank.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	SUBEPSO No. of Evaluations	SUBEPSO Average Fitness
0	0	2,1272156	0	2,1973864
1	40	2,1272156	280	1,2044196
2	80	2,0056922	320	1,0133082
3	120	1,785832	360	0,655945
4	160	1,6854152	400	0,2037992
5	200	1,6150012	440	0,0218694
6	240	1,525274	480	0,0020616
7	280	1,4788814	520	0,00026575
8	320	1,3953214		
9	360	1,3483064		
10	400	1,3270264		
11	440	1,3069126		
12	480	1,2774172		
13	520	1,257716		
14	560	1,227376		
15	600	1,2058306		
16	640	1,1872488		
17	680	1,1748338		
18	720	1,1610396		
19	760	1,1513324		
20	800	1,1312092		
21	840	1,1170004		
22	880	1,1069902		
23	920	1,1010632		
24	960	1,0960282		
25	1000	1,0918178		
26	1040	1,0878764		
27	1080	1,083724		
28	1120	1,0815064		
29	1160	1,0769866		

Table B.4: Average *fitness* progression for Ackley.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	SUBEPSO No. of Evaluations	SUBEPSO Average Fitness
0	0	17,6581796	0	17,1705096
1	40	17,6581796	280	10,8738984
2	80	16,9796888	320	6,1335644
3	120	16,1508206	360	3,091681
4	160	15,7476316	400	1,4301574
5	200	15,2542944	440	0,4080326
6	240	14,7683638	480	0,1114216
7	280	14,4361982	520	0,0385132
8	320	13,8733618	560	0,0135644
9	360	13,1516598	600	0,004049
10	400	12,7917364	640	0,0015468
11	440	12,588249	680	0,0005362
12	480	12,4302636	720	0,000268667
13	520	12,1814044	760	0,000136
14	560	11,8506328		
15	600	11,612533		
16	640	11,2926176		
17	680	11,1646564		
18	720	10,6760234		
19	760	10,354053		
20	800	10,147467		
21	840	9,8636312		
22	880	9,7297268		
23	920	9,4800032		
24	960	9,3187768		
25	1000	9,1532018		
26	1040	8,8902258		
27	1080	8,7909076		
28	1120	8,6948526		
29	1160	8,417546		

Appendix C

Annex C - Energy Power System - *Fitness* progression

This annex presents the difference of performances between original EPSO version and SUBEPSO used for state estimation from capitle 5. These values result from the average of five different tests.

Table C.1: Difference of performances for state estimation.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	SUBEPSO No. of Evaluations	SUBEPSO Average Fitness
0	0	6118,263017	0	7281,837013
1	40	6118,263017	280	1375,45352
2	80	5,6716154	560	10,0745906
3	120	5,6716154	920	7,4287428
4	160	5,6716154	1280	5,5575896
5	200	5,6716154	1640	4,1502542
6	240	5,3094206	2000	3,1657626
7	280	5,2985234	2360	2,5076982
8	320	5,0729308	2720	2,0219564
9	360	5,0554678	3080	1,6839968
10	400	4,9234016	3440	1,4475794
11	440	4,4858202	3800	1,2541516
12	480	4,2777934	4160	1,1167102
13	520	4,1112002	4520	1,0353182
14	560	4,107674	4880	0,9944216
15	600	3,3450592	5240	0,9698476
16	640	2,8982374	5600	0,956164
17	680	2,876539	5840	0,9522314
18	720	2,865951	6080	0,9521002
19	760	2,2464922	6320	0,9521002
20	800	2,182039	6360	0,9521002

Table C.2: (Continued) Difference of performances for state estimation.

iteration	No. of Evaluations	*Original EPSO* Average Fitness	SUBEPSO No. of Evaluations	SUBEPSO Average Fitness
21	840	2,1538674	6400	0,9521002
22	880	1,912671	6440	0,9516208
23	920	1,7396226	6480	0,9516208
24	960	1,6861254	6520	0,9516208
25	1000	1,6529554	6560	0,9516208
26	1040	1,6203836	6600	0,9132242
27	1080	1,5813656	6640	0,9023654
28	1120	1,2848436	6680	0,90136
29	1160	1,1795014	6720	0,8980406

Appendix D

Annex D - Article for submission

Following all the progress and results obtained through this dissertation. It was decided to write a paper, where the main achievements and ideas of EPSO developments would be presented.

As consequence of the results obtained through the developed strategy in this thesis, it was decided to write a paper, in which the main achievements and ideas of such strategy are presented. The article will be submitted for future publication in IEEE Transactions.

VAREPSO and SUBEPSO - New developments and testing of EPSO and DEEPSO

João Vigo* Vladimiro Miranda† Leonel Carvalho‡

***Abstract*—This paper reports new developments made at the general model of EPSO (Evolutionary Particle Swarm Optimization). Besides the original process behind EPSO, the main goal is actually presenting variants to EPSO and, if possible, improve its performance and efficiency, at same time that results are compared and analysed with same values from original version of EPSO. The constant search for improving the existing models, therefore creating algorithms that are not only more efficient but also computationally lighter, stand as the main purpose of this paper. It will be presented two (2) new variants, its ideologies and theories behind, the results and its performances.**

***Index Terms*—Computation, Efficiency, EPSO, Evolutionary Algorithms, Variant.**

I. Introduction

THIS paper pretends to introduce to the community two new variants of the meta-heuristic process EPSO - Evolutionary Particle Swarm Optimization. Born in 2002 at INESC Porto [1], this algorithm came out from the PSO - Particle Swarm Optimization, that comprises a computational technique inspired in the collective behaviour of a group of individuals who act in a coordinated and organized fashion, fostering the information exchange between each of them.

EPSO stands out as the fusion of the best aspects of all worlds, since it is able to combine two mechanisms and to allow the algorithm to "learn" which values it must define to mutation weights. Hence it pushes the progress forward toward the problem's optimum. In other words, it combines two fundamental aspects: on the one hand the *movement equation* - which is a distinctive aspect from PSO -, and on the other the selection operation with auto-adaptation capacity.

This fusion is what defines EPSO as a hybrid optimization model. Several tests showed a more robust, more efficient and more reliable algorithm, whose results may be considered better than any seen in the past. In electric energy systems, this is also used in a broad set of applications.

The search not only for a more robust but also for a quicker and a computationally lighter method is definitely a barrier which is difficult to break. Nonetheless, there are several theories arising from brain stimulation exercises, from which a set of ideas can be put into practice, tested and compared to the original model.

*João Vigo, Master in Electrical and Computers Engineering Student at FEUP (912 699 111 — email: joaovigo18@gmail.com)

†V. Miranda is with INESC Porto and also with FEUP, Faculty of Engineering of the University of Porto, Portugal (email: vmiranda@inescporto.pt

‡L. Carvalho is with INESC Porto and also with FEUP, Faculty of Engineering of the University of Porto, Portugal (email: lcarvalho@inescporto.pt)

Considering the constant increase in complexity of the aforementioned problems, the search for equally more complex algorithms implies higher computational efforts. With this arises the challenge of optimizing the already existing models. In order to achieve this goal, several variants of the original EPSO algorithm will be implemented in the C++ language [2], registering the program's reaction to the implemented changes. As it was said before, this paper will present some results from these proposed variants of EPSO.

II. State of the Art

Inspired in biology and the observation of nature, many algorithms have been proposed, the most important family being the Evolutionary Computation algorithms. The *Particle Swarm Optimization, PSO* is another example, as it was inspired by the collective movement of bird flocks or bee swarms and will be approached in greater detail in the following sections. [3][4].

Born of the original PSO model, ESPO borrows the *particle movement process*. In this model each particle groups a set of vectors that define its **position**, more specifically the position vector X_i, the **speed** vector V_i, and the vector for the **best position occupied by the particle up until that exact moment,** b_i. A fourth term is then added to EPSO, symbolizing the **cooperation**, in other words, the best position occupied by the total set of particles of the swarm, which is memorized in the vector b_G. The **movement rule** that determines the new position of each particle of the solution swarm:

$$\mathbf{X}_i^{new} = \mathbf{X}_i + \mathbf{V}_i^{new} \tag{1}$$

Where V_i can be defined as the speed of the particle X_i and is calculated as follows:

$$\mathbf{V}_i^{new} = W_{in_i}.\mathbf{V}_i + Rnd().W_{m_i}(\mathbf{b}_i - \mathbf{X}_i) + Rnd().W_{e_i}(\mathbf{b}_G - \mathbf{X}_i)\mathbf{P} \tag{2}$$

As previously stated, the equation 2 compasses the several aforementioned factors, featuring the inertia as its first term, which represents the fact that the particle keeps its movement on the same direction as presented before. Memory stands out as the second term, defined by the presence of the vector with the best *fitness* position that was reached up until that moment, and which attracts the particle to that best. Last but not least, the cooperation factor stimulates the swarm's information exchange, so that the particle is also attracted to the best point reached by the whole cluster.

III. Variants to EPSO - VAREPSO and SUBEPSO

The first proposal of a variant to original version of EPSO, called "VAREPSO", is inspired on thoughts from change of variables. From iteration to iteration, the swarm is analysed and the dimensions are target of evaluations about their relation between each other. According with some criterion, each particle suffers a re-scaling on dimension that shows a large amplitude between extremes particles.

This idea emerges from the fact that, at some moments, the swarm spreads through dimensional space and big disparities among dimensions appear. With this, we do a reduction of the differences of distance between extreme particles of dimension. This increases the equality among dimensions and could conduct to a faster and computational lighter convergence.

Secondly, the "SUBEPSO" is a variant that come from a simple rhetoric question: "Why not the creation of an EPSO inside another?". At first sight, this could be seems strange, however, it makes a little sense. The main idea was creating a smaller swarm inside the big one, which we call mum-swarm. The smaller one, named son-swarm, would be centralized around one of two options: 1) based on the global best position that until that iteration, has registered the best *fitness* yet or 2) based on a position from a random selected particle.

The objective of these satellite swarms were to further research in area in which it showed more favorable at that time or then look in other areas of dimensional space to prevent possible "jams" in local optima, respectively. Therefore, in each generation of particles at main swarm, it would be created a smaller cluster that, developed with few iterations could analyse and evaluate if that specific dimensional space is favorable or not to mum-swarm progression.

IV. Results of Variants to EPSO

Both variants were target of several tests. Using optimization functions and having the reference of original EPSO results, we could evaluate the performances of each other. For optimization with these variants to EPSO, were used functions like Rosenbrock, Sphere, Alpine, Griewank and Ackley. The following table shows the performances of each variant:

Table I
Average of computational efforts with optimization of several functions.

	Function	No. of Iterations	No. of Evaluations
Original EPSO	Rosenbrock	10545.4	421816
	Sphere	518.1	20724
	Alpine	1467.5	58700
	Griewank	549.2	21968
	Ackley	269.8	10792
VAREPSO	Rosenbrock	9777.7	391108
	Sphere	477.5	19100
	Alpine	47.4	1896
	Griewank	17.5	700
	Ackley	22.1	884
SUBEPSO	Rosenbrock	9652.6	386436
	Sphere	523	21252
	Alpine	10.5	660
	Griewank	7.7	548
	Ackley	12.6	744

With some of these results, it has shown the potential that these variants have. To prove that, SUBEPSO was applied to a energy system problem, based on states estimation, where least squares criterion was used on minimization of error deviation. The following figure illustrates the performance:

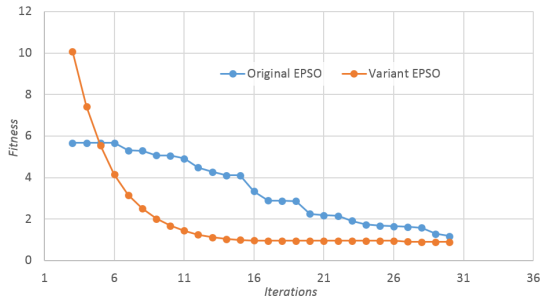

Figure 1. *Fitness* progression for states estimation, with original EPSO and SUBEPSO.

The previous illustration shows the effect that this variant has the acceleration in beginning of the optimization process, contributing to increase of the computational efficiency of the original algorithm EPSO.

V. Conclusions

Through the presented results, we can withdraw several important conclusions. First, an interpretation of how the swarm spreads into the dimensional space, doing successive analyses about the difference between dimensions could be beneficial to the convergence of EPSO. Actually, VAREPSO showed some potential solving problem like, Alpine, Griewank and Ackley with computation efforts improvements in order of 90%.

In other hand, a specific swarm created inside swarm-mum proved that is a good variant EPSO strategy, showing results of improvements in order of 96% at same functions, and otherwise that in a real environment like a energy system problem, SUBEPSO improved the original version of EPSO, by accelerating the progression of *fitness*.

Finally, two variants arise in the attempt to launch a new paradigm in the scope of evolutionary computation, and especially in the scope of EPSO. The constant strive for increasing the models' efficiency, whilst keeping their robustness, was the main scope of this thesis.

References

[1] Vladimiro Miranda and Nuno Fonseca. "EPSO-best-of-two-worlds meta-heuristic applied to power system problems". *In Proceedings of WCCI/CEC – World Conference on Computational Intelligence, Conference on Evolutionary Computation*, volume 2, pages 1080–1085. Honolulu (Hawaii), USA, May 2002. DOI:10.1109/CEC.2002.1004393.

[2] EPSO code c++, 2009. URL: http://epso.inescporto.pt/epso-code-c.

[3] Vladimiro Miranda and Nuno Fonseca. "EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems". *In Proc. of the Asia Pacific IEEE/PES Transmission and Distribution Conference and Exhibition*, volume 2, pages 745–750. Yokohama, Japan, Oct 2002. DOI:10.1109/TDC.2002.1177567.

[4] Vladimiro Miranda. "Evolutionary algorithms with particle swarm movements". *In Intelligent Systems Application to Power Systems 2005. Proceedings of the 13th International Conference on*, pages 6–21. Arlington, VA, IEEE, Nov 2005. doi:10.1109/ISAP.2005. 1599236.