Department of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

Real-valued Optimization by Subspace Projection and Multi-armed Bandit Techniques

Chun-Jen Peng

Advisor: Tain-Li Yu, Ph.D.

106 6
June, 2017

106
6

Acknowledgments

I'm glad to thank ...

...

(region of interest) (benchmark)
Photoshoot

Abstract

This thesis presents a benchmark for region of interest (ROI) detection. ROI detection has many useful applications and many algorithms have been proposed to automatically detect ROIs. Unfortunately, due to the lack of benchmarks, these methods were often tested on small data sets that are not available to others, making fair comparisons of these methods difficult. Examples from many fields have shown that repeatable experiments using published benchmarks are crucial to the fast advancement of the fields. To fill the gap, this thesis presents our design for a collaborative game, called Photoshoot, to collect human ROI annotations for constructing an ROI benchmark. With this game, we have gathered a large number of annotations and fused them into aggregated ROI models. We use these models to evaluate five ROI detection algorithms quantitatively. Furthermore, by using the benchmark as training data, learning-based ROI detection algorithms become viable.

Contents

Acknowledgments **i**

iii

v

Abstract **vii**

1 Introduction **1**

1.1 Thesis Objectives 1

1.2 Roadmap 2

2 Real-valued Optimization Algorithms **3**

2.1 Covariance Matrix Adaptation Evolution Strategy 3

2.2 Standard Particle Swarm Optimization 3

2.3 Ant Colony Optimization for Continuous Domain 6

3 Clustering Techniques **9**

3.1 K-Means clustering 9

3.2 Determine number of clusters 9

3.2.1 Silhouette coefficient 9

3.2.2 Gap statistics 9

3.2.3 Dip test 9

3.3 Heirarchical Clustering 9

4 Linear Projection **11**

4.1 Affine Transformation 11

4.1.1 Translation 11

4.1.2 Rotation 11

4.1.3 Scaling 11

4.1.4 Shear 11

4.1.5 Affine transformation matrix 11
4.2 Projection 11
4.2.1 Basic Projection 11
4.2.2 Homogeneous Coordinate 11
4.2.3 Perspective Projection 12
4.3 Optimization for Projection Matrix 12
4.3.1 Loss function 12
4.3.2 Optimization Algorithms 12

5 Multi-armed Bandit Algorithms 13
5.1 The Multi-armed Bandit Problem 13
5.2 Some common MAB Algorithm 13
5.2.1 UCB 13
5.2.2 POKER 13
5.3 The New Bandit Technique 14

6 The New Bandit Technique 15
6.1 Framework of the New Bandit Algorithm 15
6.2 Initialization and Unimodal Identification 15
6.3 Define Region of Interest 15
6.4 Remain Evaluations Allocation 16
6.5 Recluster 16

7 Experiments 17
7.1 Test Problems 17
7.1.1 CEC2005 25 benchmark problems 17
7.2 Experiment Settings 17

8 Conclusion 19

Bibliography 21

List of Figures

2.1 (a) SPSO 2011. (b) SPSO 2006. 5

2.2 (a) Discrete probability distribution $p(c_{ij}|s^p)$. (b) Continuous probability density function $p(x|s^p)$ 7

2.3 (a) Discrete probability distribution $p(c_{ij}|s^p)$. (b) Continuous probability density function $p(x|s^p)$ 7

List of Tables

7.1 Summary of the parameters used by ACO_R 17

Chapter 1

Introduction

Attention plays an important role in human vision. For example, when we look at an image, our eye movements comprise a succession of *fixations* (repetitive positioning of eyes to parts of the image) and *saccades* (rapid eye jump). Those parts of the image that cause eye fixations and capture primary attention are called *regions of interest* (ROIs). Studies in visual attention and eye movement have shown that humans generally only attend to a few ROIs. Detecting these visually attentive regions in images is challenging but useful in many multimedia applications, such as automatic thumbnail cropping, object recognition, content-based image retrieval, adaptive image compression and automatic browsing in small-screen devices.

Describe the common dilemma between exploration and exploitation for real-valued optimization algorithms. Describe our basic assumption that problems worth solving are hierarchical decomposable [**?**].

1.1 Thesis Objectives

We propose a technique that helps identify *regions of interest* (ROI) to explore and *allocate resources* according to remaining evaluations.

First, describe why is it important to identify ROIs. Describe how subspace projection creates a *well-defined boundary* that some algorithms need. Describe how subspace projection helps solve *inseparable problems* while enhance the ability to find optimium.

Second, describe how the proposed resources allocation benefits optimization. Describe different strategies one should take given different evaluations left. Describe how Multi-armed Bandit (MAB) algorithms are suitable for the scenario, instead of decision theory, reinforcement learning and Markov Decision Process. MAB learns models from outcomes while the actions do not change the state of the world.

1.2 Roadmap

This thesis is composed of seven chapters.

Chapter 2 presents three optimization algorithms that are adopted for comparisons. These three algorithms each have different characteristics. The Covariance Matrix Adaptation Evolutionary Strategy. The Standard Particle Swarm Optimization. The Ant Colony Optimization for Continous Domain.

Chapter 3 presents some clustering techniques that guides the construction of ROIs and later becomes the initial points for algorithms in each arm.

Chapter 4 first describes four basic affine transformation: translation, rotation, scaling and shearing. Then the projective transformation and homogeneous coordinate are presented.

Chapter 5 briefly describes some common multi-armed bandit algorithms, including ... Then we present our new bandit techniques. Tranditional bandit algorithms focus on minimizing regret, while our new bandit focus on the probability of getting a rank 1 result.

Chapter 6 gives details of our new algorithms. First, the framework and pseudo code are given. Then a detailed process of initialization is given in ...

Chapter 8 summarizes this thesis. The conclusion and contributions are also given. Some further improvements and future works are also discussed at the end.

Chapter 2

Real-valued Optimization Algorithms

Overview of real-valued optimization

2.1 Covariance Matrix Adaptation Evolution Strategy

Describe history of *Evolutionary Strategies* (ES). The simplest algorithm is (1+1)-ES. Here we describe the (1+1)-ES with one-fifth success rule with independent restarts. The pseudo code of (1+1)-ES is given in Algorithm 1.

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is an extended version fo CSA-ES with de-randomized adaptation of covariance matrix. Describe the underlying covariance matrix model.

Describe how to update *mean*.

Describe how to update *covariance matrix*.

Describe *step-size* control.

2.2 Standard Particle Swarm Optimization

Particle Swarm Optimization was first proposed by ... The swarm intelligence family... However, there are multiple varients of PSO over the years.

Standard PSO provides a well defined version of PSO that follows the basic principles. It is intend to be a milestone for future comparison, instead of the best algorithm on the market.

So far, there have been three successive versions of standard PSO: SPSO 2006,

Algorithm 1: (1+1)-ES with 1/5 success-rule

```
$\boldsymbol{X}_n$: solution of the $n^{th}$ iteration, $\sigma_n$: step size of the $n^{th}$ iteration,
$N(\mathbf{0}, \boldsymbol{I})$: multivariant normal distribution with mean vector $\mathbf{0}$
and identical covariance matrix $\boldsymbol{I}$.

input  : $f$: evaluation function
output: $X_{n+1}$: best solution

Initialize $\boldsymbol{X}_0, \sigma_0$
while termination criterion not met do
    $\widetilde{\boldsymbol{X}}_n = \boldsymbol{X}_n + \sigma_n N(\mathbf{0}, \boldsymbol{I})$
    if $f(\widetilde{\boldsymbol{X}}_n) \leq f(\boldsymbol{X}_n)$ then
        $\boldsymbol{X}_{n+1} = \widetilde{\boldsymbol{X}}_n$
        $\sigma_{n+1} = 1.5\sigma_n$
    else
        $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n$
        $\sigma_{n+1} = 1.5^{-1/4}\sigma_n$
return $\boldsymbol{X}_{n+1}$
```

2007 and 2011 The underlying principles of these three versions are generally the same as all PSO varients. The exact formula and implementation are slightly different due to latest theoretical progress.

Describe swarm size definition and basic elements for each particle. Initialization of the swarm. The swarm size, denoted as S, differs in SPSO 2006 and SPSO 2011. In both SPSO 2006 and SPSO 2007, the initial number of particles for dimension D is defined as:

$$S = 10 + \lfloor 2\sqrt{D} \rfloor,$$

However, in SPSO 2011, the swarm size can be defined by user, yet suggested as 40 [2] since the original swarm size is far from optimal.

Each particle in the swarm possesses the following elements: current position, current velocity, personal pervious best position, and previous best position in the neighbourhood.

Describe random topology and when to update topology. The information links... The adaptive random topology described in [1] is formally equivalent to ”Stocastic Star”.

Describe velocity update for SPSO 2006 and SPSO 2011 Update Velocity as shown in Figure 2.1

Describe boundary and out-of-bound handling.

The pseudo code defined in [5]. The pseudo code is given in Algorithm 2.

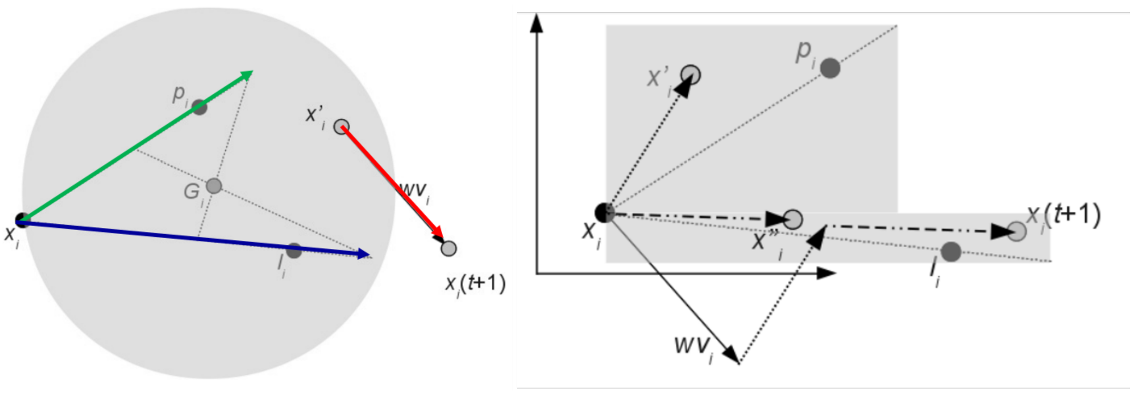

Figure 2.1: (a) SPSO 2011. (b) SPSO 2006.

Algorithm 2: Standard PSO 2011

```
$\boldsymbol{X}_n$: solution of the $n^{th}$ iteration, $\sigma_n$: step size of the $n^{th}$ iteration,
$N(\boldsymbol{0}, \boldsymbol{I})$: multivariant normal distribution with mean vector $\boldsymbol{0}$
and identical covariance matrix $\boldsymbol{I}$.

input  : $f$: evaluation function
output: $X_{n+1}$: best solution

Initialize $\boldsymbol{X}_0, \sigma_0$
while termination criterion not met do
    $\widetilde{\boldsymbol{X}}_n = \boldsymbol{X}_n + \sigma_n N(\boldsymbol{0}, \boldsymbol{I})$
    if $f(\widetilde{\boldsymbol{X}}_n) \leq f(\boldsymbol{X}_n)$ then
        $\boldsymbol{X}_{n+1} = \widetilde{\boldsymbol{X}}_n$
        $\sigma_{n+1} = 1.5\sigma_n$
    else
        $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n$
        $\sigma_{n+1} = 1.5^{-1/4}\sigma_n$
return $\boldsymbol{X}_{n+1}$
```

2.3 Ant Colony Optimization for Continuous Domain

Ant Colony optimization (ACO) is first proposed by Dorigo [3] to solve combinatorial optimization problems, including scheduling, routing, and timetabling. These problems aim to find optimal *combinations* or *permutations* of finit sets of available components. Inspired by the foraging behavior of natural ants, ACO mimics the pheromone deposition of ants along the trail to a food source. The deposited pheromone, which indicates the quantity and quality of the food, creates an indirect communication among ants and enables them to find the shortest paths. The pseudo code of ACO is given in Algorithm 3. Two major procedures: *solution construction* and *phermone update*, are detailed in the following paragraph.

Consider a search space $\boldsymbol{S}$ defined over a finit set of all possible *solution components*, denoted by $\boldsymbol{C}$. Each solution component, denoted by c_{ij}, is a decision variable X_i instantiated with value $v_i^j \in \boldsymbol{D}_i = \{v_i^1, ..., v_i^{|\boldsymbol{D}_i|}\}$. To construct a new solution, an artificial ants starts with an empty partial solution $s^p = \emptyset$. During each construction step, the partial solution s^p is extended with a feasible solution from the set $N(s^p) \in \boldsymbol{C} \setminus s^p$. The probabilistic pheromone model adopted for selecting a feasible solution from $N(s^p)$ can be defined as follows:

$$p(c_{ij}|s^p) = \frac{\tau_{ij}^{\alpha} \cdot \eta(c_{ij})^{\beta}}{\sum_{c_{i\ell} \in N(s^p)} \tau_{i\ell}^{\alpha} \cdot \eta(c_{i\ell})^{\beta}}, \forall c_{ij} \in N(s^p), \tag{2.1}$$

where τ_{ij} is the pheromone value associated with component c_{ij}, and $\eta(\cdot)$ is a weighting function. α and β are positive parameters which determine the relation between phermone and heuristic information.

The pheromone update

Over the years, multiple approaches of extending the ACO on continous domain have been given. One of the most successful version is ACO$_R$, proposed by Socha and Dorigo in 2008 [4]. It extends ACO to the continuous domain without making any major conceptual change to its structure. The fundamental idea underlying ACO$_R$ is the shift from using a discrete probability distribution to using a continuous one, demonstrated in Figure 2.2.

A enhanced Gaussian kernel PDF as shown in Figure 2.3.

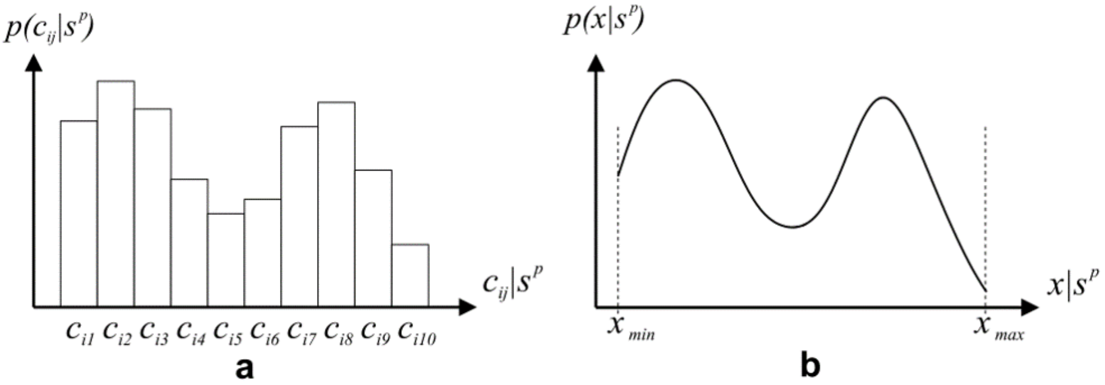

Figure 2.2: (a) Discrete probability distribution $p(c_{ij}|s^p)$. (b) Continuous probability density function $p(x|s^p)$

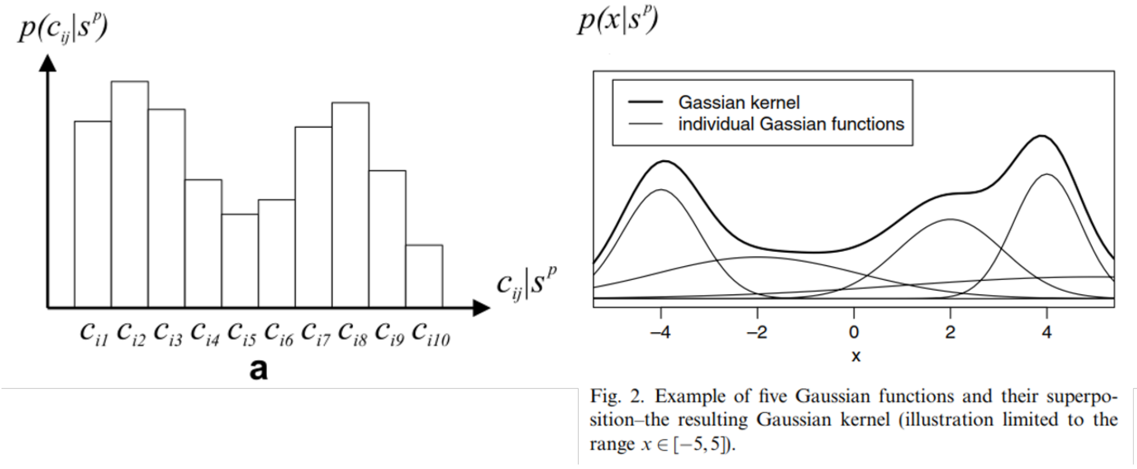

Fig. 2. Example of five Gaussian functions and their superposition–the resulting Gaussian kernel (illustration limited to the range $x \in [-5, 5]$).

Figure 2.3: (a) Discrete probability distribution $p(c_{ij}|s^p)$. (b) Continuous probability density function $p(x|s^p)$

Algorithm 3: Ant Colony Optimization metaheuristic

```
while  termination criterion not met do
    schedule activities
    solution contruction by ants();
    phermone update();
    daemon actions();
```

Chapter 3

Clustering Techniques

Describe our basic assumption of function decomposition. Each subproblem should be composed of an observalbe uni-model. We wish to identify these uni-models through clustering techniques.

3.1 K-Means clustering

Describe how K-means clustering works and why it is popular

Describe the limits for K-Means clustering, e.g. it cannot identify density nor unimodality.

3.2 Determine number of clusters

3.2.1 Silhouette coefficient

Describe how silhouette score decides number of clusters

3.2.2 Gap statistics

Describe how gap statistics estimates number of clusters.

3.2.3 Dip test

Describe how Dip-test checks unimodality Describe skynny-dip clustering

3.3 Heirarchical Clustering

Describe the advantage of considering fitness instead of just density.

Chapter 4

Linear Projection

For each uni-model, we wish to define a region of interest for exploitation.

Projection creates a well-defined hypercube boundary on subspace that some algorithms need, e.g. SPSO.

Subspace projection also gives advantage to solving *inseparabel problems.*

4.1 Affine Transformation

4.1.1 Translation

4.1.2 Rotation

4.1.3 Scaling

4.1.4 Shear

4.1.5 Affine transformation matrix

4.2 Projection

4.2.1 Basic Projection

4.2.2 Homogeneous Coordinate

Augmented Matrix

4.2.3 Perspective Projection

4.3 Optimization for Projection Matrix

4.3.1 Loss function

Describe our loss function:

- Points within boundaries should be within $[0,1]^D$ in subspace
- Points within other boundaries should be out of $[0,1]^D$ in subspace
- Point with best fitness should be at the center
- We hope that the standard deviation on each dimension should be around 0.2
- Minimal error for inverse transformation

4.3.2 Optimization Algorithms

Describe the difficulty of finding a fast yet powerful algorithm for *hyperparameters optimization*.

We first tried CMA-ES.

Later, we utilize the (1+1)-ES, described in Algorithm 1.

Chapter 5

Multi-armed Bandit Algorithms

After identifying the unimodals in Chapter 3 and defining a ROI for exploitation in Chapter 4, we still need to decide how to allocate our resources. Describe the exploration vs. exploitation dilemma. One should take different strategies according to evaluations left. Instead of letting the algorithms handling both eploration and exploitation, we propose a bandit technique to help manage exploitation.

Multi-armed Bandit Algorithms are suitable for this scenario. It learns model from outcomes and the actions do not change the state of the world.

Unlike canonical MAB algorithms that minimize regrets, we wish to maximize the probability of gaining the maximum rank.

5.1 The Multi-armed Bandit Problem

Multi-aremed Bandit (MAB) Problem describes an agent needs to decide in K arms to pull at time t and receives a reward.

Describe *policy* and *regret*.

The goal is to minimize regret.

5.2 Some common MAB Algorithm

5.2.1 UCB

5.2.2 POKER

The Price of Knowledge and Estimated Reward (POKER) strategy considers three ideas: pricing uncertainty, exploiting the lever distribution and taking into account the horizon

5.3 The New Bandit Technique

Chapter 6

The New Bandit Technique

In this chapter, we illustrate the proposed algorithm step-by-step.

Overview of real-valued optimization

6.1 Framework of the New Bandit Algorithm

The goal is to identify the ROI for exploitation and maintain exploration through resource allocation.

Each algorithm needs to be modified to satisfy the following conditions:

1. Update one individual at a time
2. The algorithm can be projected onto a subspace and continue iterating
3. Replace one individual with a given position and fitness

The pseudocode of our new algorithm is given in Algorithm **??**

6.2 Initialization and Unimodal Identification

For a problem with D dimension, initialize with $100D$ points.

Keep only top 50% of points for unimodal identification. Use clustering techniques mentioned in Chapter 3.

Iteratively add $10D$ points until estimated cluster number is identical.

With a given cluster number k, do K-Means clustering.

6.3 Define Region of Interest

Each arm is composed of an algorithm and a projection matrix. Initial matrix is set as a tight hyperbox that contains all points in the given cluster. Optimize matrix

Algorithm 4: Framework of the new Bandit Algorithm

```
$\boldsymbol{X}_n$: solution of the $n^{th}$ iteration, $\sigma_n$: step size of the $n^{th}$ iteration,
$N(\mathbf{0}, \boldsymbol{I})$: multivariant normal distribution with mean vector $\mathbf{0}$
and identical covariance matrix $\boldsymbol{I}$.
input  : $f$: evaluation function
output: $X_{n+1}$: best solution
Initialize $\boldsymbol{X}_0, \sigma_0$
while termination criterion not met do
    $\widetilde{\boldsymbol{X}}_n = \boldsymbol{X}_n + \sigma_n N(\mathbf{0}, \boldsymbol{I})$
    if $f(\widetilde{\boldsymbol{X}}_n) \leq f(\boldsymbol{X}_n)$ then
        $\boldsymbol{X}_{n+1} = \widetilde{\boldsymbol{X}}_n$
        $\sigma_{n+1} = 1.5\sigma_n$
    else
        $\boldsymbol{X}_{n+1} = \boldsymbol{X}_n$
        $\sigma_{n+1} = 1.5^{-1/4}\sigma_n$
return $\boldsymbol{X}_{n+1}$
```

one-by-one. Resize each cluster to match the required population for each algorithm. Start algorithm with given initial positions and fitnesses.

6.4 Remain Evaluations Allocation

Calcualte remain evaluations allocatoin and normalize the value to 0 1. Add newly calculated allocatoin to record. Choose arm the argmax allocation to pull, i.e. update one individual. Max arm allocation -1.

6.5 Recluster

Chapter 7

Experiments

Overview of real-valued optimization

7.1 Test Problems

7.1.1 CEC2005 25 benchmark problems

Describe termination criterion and evaluation method for CEC2005 Describe reason for 25 repeat runs. Describe how initial setting effect real-valued optimization, so that all algorithms should start with identicle initial status.

7.2 Experiment Settings

Describe the parameters setting for CMA-ES, SPSO and ACOR CMA-ES initial mean and std, population settings SPSO2011 parameters (c, w) settings, and population settings For ACOR, we set our parameters according to the original paper **??**. The parameters are shown in Table 7.2.

Describe our bandit parameters setting, including the initial population, maximum number of arms, and (1+1)-ES step size.

Parameter	Symbol	Value
No. of ants used in an iteration	m	2
Speed of convergence	ξ	0.85
Locality of the search process	q	10^{-4}
Archive size	k	50

Table 7.1: Summary of the parameters used by ACO_R

Chapter 8

Conclusion

Contribution: find potential region to search allocate resources

Weakness: Potentially more NFE on unimodals and moving regions. Requires a better clustering techniques to identify underlying unimodals.

Future work: Non-linear transformation

Attention plays an important role in human vision. For example, when we look at an image, our eye movements comprise a succession of *fixations* (repetitive positioning of eyes to parts of the image) and *saccades* (rapid eye jump). Those parts of the image that cause eye fixations and capture primary attention are called *regions of interest* (ROIs). Studies in visual attention and eye movement have shown that humans generally only attend to a few ROIs. Detecting these visually attentive regions in images is challenging but useful in many multimedia applications, such as automatic thumbnail cropping, object recognition, content-based image retrieval, adaptive image compression and automatic browsing in small-screen devices.

Bibliography

[1] M. Clerc. Back to random topology. *Relatório Técnico, mar*, 2007.

[2] M. Clerc. Standard particle swarm optimisation. 2012.

[3] M. Dorigo and G. Di Caro. Ant colony optimization: a new meta-heuristic. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 2, pages 1470–1477. IEEE, 1999.

[4] K. Socha and M. Dorigo. Ant colony optimization for continuous domains. *European journal of operational research*, 185(3):1155–1173, 2008.

[5] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas. Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 2337–2344. IEEE, 2013.