TECHNICAL REPORT

TR 2007/02 **ISSN 0874-338X**

Niching Techniques: a Study on the Cluster Geometry Optimization Problem

Tiago Leitão, Francisco B. Pereira, Jorge M. C. Marques,

Jorge Tavares, Ernesto Costa

CISUC

Centre for Informatics and Systems of the University of Coimbra

Niching Techniques: a Study on the Cluster Geometry Optimization Problem

Tiago Leitão[1], Francisco B. Pereira[1,2], Jorge M.C. Marques[3], Jorge Tavares[1], Ernesto Costa[1]

[1] Centro de Informática e Sistemas da Universidade de Coimbra
Polo II, Pinhal de Marrocos, 3030 Coimbra, Portugal

[2] Instituto Superior de Engenharia de Coimbra,
Quinta da Nora, 3030 Coimbra, Portugal

[3] Depart. de Química da Universidade de Coimbra, 3004-535 Coimbra, Portugal

E-mail: tleitao@student.dei.uc.pt,
{xico, jast, ernesto}@dei.uc.pt,
qtmarque@ci.uc.pt

Abstract: Niching techniques are commonly used in Evolutionary Computation to maintain population diversity and help to explore more efficiently the solutions space. In complex optimization problems, the definition of a distance measure between individuals is a serious difficulty for this type of techniques. The cluster geometry optimization problem has singularities that make the use of a reliable distance function computationally prohibitive. We analyze the behavior of an alternative niching technique, the Spatially-Dispersed Genetic Algorithm, and its application on the cluster geometry optimization. Results show that the algorithm is able do efficiently discovers several niches when searching for good solutions. We also present a detailed parametric analysis that helps to understand the role played by the different parameters of the method.

Keywords: Niches, cluster geometry optimization, parameter analysis.

1. INTRODUCTION

Atomic and molecular clusters may have from a few to a huge number of aggregated particles (atoms or molecules). The function describing the interactions among those particles is designated as Potential Energy Surface (PES) and plays a central role in Chemistry. Then, exploring the energy landscapes of such a multi-dimensional PES has long been a desiderata but also a very hard task to achieve. In particular, finding the arrangement of the cluster's particles that corresponds to the lowest energy (global minimum) has been a key-step to the research in this scientific domain.

Since the early 1990's, Evolutionary Algorithms (EAs) have been increasingly applied to several problems from the Chemistry/Biochemistry area. Cluster geometry optimization is an example of one of these problems [5], [8], [10]. Nearly all the approaches rely on hybrid algorithms combining EAs with local methods that use first order derivative information to guide search into the nearest local optimum.

In the real world, we can find a variety of problems for which there are many good solutions. Even when we can identify the optimal solution for a given situation, there might be some alternatives also recognized as good solutions for that problem. As a result of the need to look for more than one single point in the search space, niching techniques have been developed during the last two decades [6], [11]. When enhanced with these techniques, EAs are capable to efficiently explore the search space and to look for different good solutions simultaneously.

Cluster geometry optimization is an example of one of these problems. Usually we are interested in discovering, not only the global minimum, but also some good quality local minima, as they correspond to stable geometries with relevant structural properties.

Therefore, the aim of this work is to develop and test an evolutionary architecture that is able to search for good solutions simultaneously. Spatially Dispersed Genetic Algorithm (sdGA) [2] , a niching technique recently proposed, will be used for this purpose. The main advantage of sdGA over standard techniques is that is does not require the definition of a distance measure between individuals to create and maintain niches throughout the optimization. This is an important feature, as measuring the distance between solutions for a given cluster geometry optimization problem is not a trivial task. The research described in this paper comprises a comprehensive analysis concerning the influence of different parameters on the efficiency of the sdGA.

The structure of this paper is the following: In section 2, Morse clusters and the evolutionary algorithm used in this work are described. In section 3 we briefly present two classical niching techniques. Section 4 describes the sdGA method. The experimental results and interpretation are presented in section 5. In section 6 the main conclusions are gathered.

2. MORSE CLUSTERS AND EAS

2.1. Problem Description

Morse clusters are considered a benchmark for testing the performance of new methods for cluster structure optimization. The energy of such a cluster is represented by the N-particle pair-wise additive potential [13] defined as

$$V_{Morse} = \varepsilon \sum_{i}^{N-1} \sum_{j>i}^{N} \left[e^{-2\beta(r_{ij} - r_0)} - 2e^{-\beta(r_{ij} - r_0)} \right] \qquad (1)$$

where the variable r_{ij} is the Cartesian distance between atoms i and j in the cluster structure. The bond dissociation energy ε, the equilibrium bond length r_o and the range exponent of the potential β are parameters defined for each individual pair-wise

Morse interaction. Usually, these are assumed to be constant for all interactions in a cluster formed by only one type of atoms.

2.2. Evolutionary Approaches

EAs have been used since 1993 for cluster geometry optimization. A comprehensive review of these efforts, including an outline of state-of-the-art applications, can be found at [7] . In what concerns the application of EAs to Morse clusters, the most important works are from Johnston and collaborators [10], [14] . The EA used in this research is similar to the one used by these researchers. Its main components have been proposed and evaluated by different teams [1], [8], [10], [16]. Also, in a previous work [17], we studied its properties and determined the specific role played by each one of the genetic operators. Here we adopt the same evolutionary model, enhancing it with tools to create and maintain niches during search. Next we briefly describe its main components.

2.2.1. Chromosome Representation and Evaluation

For aggregates with N atoms, a solution is composed by 3xN real values specifying the Cartesian coordinates of each one of the particles. Zeiri proposed this representation in 1995 [16] and, since then, it has become the most widely used in this context [8], [10] . The coordinate values range between 0 and λ. We set λ to $N^{1/3}$. This interval ensures that the aggregate volume scales correctly with N [14] . To evaluate an individual we just have to calculate its potential energy. Equation 1 is used to assign fitness.

2.2.2. Population Model and Genetic Operators

A standard generational model is adopted. The current set of variation operators is used: crossover and mutation.

Cut and splice crossover, proposed by Deaven and Ho [1] , is directly applied to the cluster structure and several authors claim that it enhances the performance of the algorithm [8], [14]. . The purpose of this operator is to exchange sub-clusters between parents. Therefore, it cuts them in two sections and creates descendants by combining complementary parts.

Sigma mutation is used in this work. Mutation is performed in atoms, i.e., when applied it modifies the value of the three coordinates. The used operator is an evolutionary strategy like mutation and acts in the following way: when undergoing mutation, the new value v_{new} for each one of the three coordinates of an atom (x, y, z) is obtained from the old value v_{old} in the following way:

$$v_{new} = v_{old} + \sigma \times N(0,1) \qquad (2)$$

where $N(0,1)$ represents a random value sampled from a standard Normal distribution and σ is a parameter from the algorithm. The new value must be between 0 and λ.

2.2.3. Local Optimization

Local optimization is performed with the Broyden-Fletcher-Goldfarb-Shanno limited memory quasi-Newton method (L-BFGS) of Liu and Nocedal [12]. L-BFGS is a powerful optimization technique that aims to combine the modest storage and computational requirements of conjugate gradient methods with the superlinear convergence exhibited by full memory quasi Newton methods.

L-BFGS is applied to every generated individual. During local search, the maximum number of iterations that can be performed is specified by a parameter of the algorithm, the Local Search Length (LSL). However, L-BFGS stops as soon as it finds a local optimum, so the effective number of iterations can be smaller than the value specified by LSL.

3. CLASSICAL NICHING TECHNIQUES

3.1. Niching Techniques

The primary aim of using methods in EAs capable of create and maintain population niches throughout the evolution cycle is to look for more than on single solution for a problem. Their contribution resides on helping the populations to explore more efficiently the solutions space. Assuming that the created niches are exploring distinct sectors of the search space, niching techniques can also be seen as a way of promoting diversity and preventing the population from stagnation and premature convergence.

There are several niching techniques successfully applied to a vast number of optimization problems. Sharing and crowding are two of the most widely used techniques.

3.1.1. Sharing

The fitness sharing technique, proposed by Goldberg [6] , consists in the penalization of each individual according to his similarity with the rest of the population. Thus, the individuals have higher probabilities of surviving when they are exploring portions of the search space with less density − the more the neighborhood of an individual is populated, the worst will be his fitness.

When using this technique, the fitness of each individual is given by the formula

$$F(i) = \frac{f_i}{\sum^{n} sh(d_{ij}),} \qquad (3)$$

where f_i is the regular fitness for an individual, n is the population size and $sh(d_{ij})$ is the similarity function defined as

$$sh(d_{ij}) = \begin{cases} 1 - \dfrac{d_{ii}}{\sigma_{sh}} & \text{, if } d_{ij} < \sigma_{sh} \\ 0 & \text{, otherwise,} \end{cases} \tag{4}$$

where d_{ij} is the distance between two individuals (i.e., the similarity function) and σ_{sh} is a threshold above which the algorithm considers that the individuals has no similarity.

3.1.2. Crowding

This technique, proposed by De Jong [11] , follows the same principle of fitness sharing: the similarity among the population individuals. Adopting the same concept of similarity from the previous technique, the algorithm ensures that the new individuals have higher probabilities of replacing old individuals that are more alike them.

Although his main functioning is different from fitness sharing, this technique leads us to the same primary need: defining a reliable function capable of measuring the similarity among individuals. Once the distance between any two individuals can be correctly quantified, the algorithm ensures that the new individuals at each generation have higher probabilities of replacing the individuals of the previous population that are more alike them.

4. SPATIALLY-DISPERSED GENETIC ALGORITHM

The main goal on global cluster geometry optimization, like in almost every optimization problems, is to attain the optimal solution for the problem. However, some cluster structures may have worst PES than the best known solutions and still be considered good solutions. For Morse clusters [15] , there are some alternative good solutions beyond the best known solution for this potential. Usually, these alternative solutions consist in local optima and are also relevant because their different types of geometry and symmetry distinguish themselves from the best known solutions.

Using niching techniques can be an effective way of looking for different good cluster geometries simultaneously. The most common niching techniques, already described, have been successfully applied to different multiobjective problems, helping the populations to efficiently explore the solutions space. Nevertheless, both these traditional techniques have a limitation: a reliable similarity measure between any two individuals of the population is required.

As a result of the need to define the similarity measure, niching techniques become, in some way, problem-dependant. Calibrating this function, which plays a central role on the niching process, can be a serious obstacle when applying the traditional techniques to some problems. A reliable similarity function can be computationally prohibitive or even impossible to define.

For the problem of cluster geometry optimization, defining a reliable function that can measure the individuals similarity in a reasonable amount of time turns out to be a very hard task. The same cluster structure can be encoded in innumerable ways. A single translation or rotation performed in the structure can lead to different genotypes for similar structures, thus increasing the difficulty of defining a good similarity function.

Considering the complexity of such a function for this particular problem, the Spatially-Dispersed Genetic Algorithm (sdGA) proposed by Grant Dick [2] , seems to be a suitable option to preserve niches in the population, once this architecture is totally free from similarity functions. Further than the search for different good solutions for the problem, this work has two additional motivation factors. On the one hand, there are no records of applications of the sdGA to optimization problems with such complexity − to the best of our knowledge, this architecture was only tested in the optimization of a few mathematical functions and some classic EAs problems [3], [4] , such as the Griewangk's function or the One-Max Problem.

On the other hand, known approaches [9] to the cluster geometry problem that rely on niching methods adopt fitness based criteria to preserve diversity (i.e., these methods aim to maintain clusters with different potential energy, but they do not try to look for alternative good solutions simultaneously).

In the next we present a comprehensive description of the main components of sdGA.

4.1. Starting the Algorithm

The sdGA functioning is based on the distribution of the individuals on a Euclidean space in two dimensions. Each individual of the population gets a x and y initial coordinate which corresponds to his start position on the spatial model. This values are assigned from a random value between 0 and *max_dim*, a specified parameter of the algorithm, compelling all the individuals to a max_dim^2 area, as shown on Figure 1 to a population of 20 individuals.

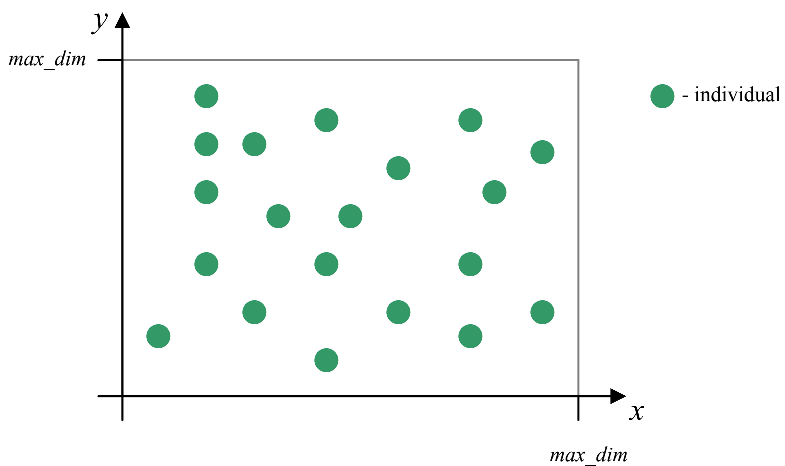

Figure 1 - spatial distribution of the individuals

After the assignment of the initial coordinates, the bounds are removed and, in the course of the optimization, the new individuals can take positions out of this area.

4.2. Selection

Beyond the initial space boundaries, a visibility radius is defined for all the individuals. This visibility radius is used to decide which individuals can mate with each selected individual, like shown in Figure 2 (from where the initial space bounds have already been removed).

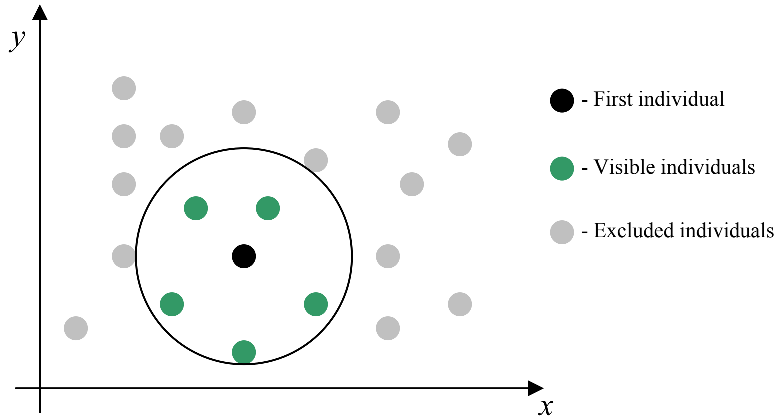

Figure 2 - Selection mechanism: visibility radius

The selection mechanism ensures that an individual can only mate with individuals visible to him. Thus, the first individual is selected accordingly with whichever selection method is being used. If no individuals are found inside the visibility radius of the first individual selected, a slight mutation in his coordinates is executed and a new first individual is selected.

The mate for the crossover process is selected from the set of individuals nearby the first selected individual, i.e., the individuals present on his visibility radius. In this work, tournament selection is used in both steps of the selection mechanism. The sizes of these tournaments are specified parameters of the algorithm.

4.3. Offspring placement

After applying the crossover operator, the new individuals are placed in a position visible to at least one of the parents. Thus, for each descendant, one parent is randomly selected and the new individual is placed at a random position inside his visibility radius, like shown in Figure 3.

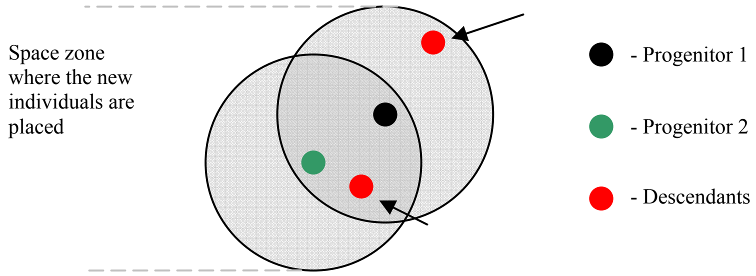

Figure 3 - Offspring placement considering descendants visibility radius

4.4. Mutation operator

Although there is no reference to the behaviour of the mutation operator in the architecture proposed by Grant Dick [2], here we introduce a simple mechanism that, when undergoing mutation, can do small modifications to both the *x* and *y* coordinates of the individuals. The introduction of this mechanism becomes from the concept that, once the individual is no longer the same after mutation, he should not occupy the same exactly position.

The new values for *x* and *y* can differ from the previous values in ½ of the visibility radius. To prevent the individual to divert too much from the original position, the algorithm can only change the coordinates of each individual once per generation, despite of how many atoms are changed during this process. Huge modifications to the individual's position can cause his isolation from the rest of the population and, in consequence, hinder him from being selected for the next generation.

5. EXPERIMENTAL RESULTS

The methods inherent to the sdGA were added to the EA described in section 2.2.

This section is divided into three subsections, reflecting the three different stages of the experimental process. The first section shows some preliminary tests for the sdGA architecture. The main purpose of this set of experiments was to make sure that the implementation had been correctly made and the sdGA was running without bugs. However, it is relevant to include here a portion of those first results because they were determinant to identify some parameters with influence on creating and maintaining niches during the algorithm cycle. In the second section, a parameter analysis to the sdGA is presented. In section 5.3 we analyze the efficiency of the algorithm in what concerns its ability to discover good quality solutions.

All the experiments had been performed for the 50 atom cluster and using 14 for the β value. This value was selected (instead of β=6, for instance) because it corresponds to a short range interaction and presents a more challenging optimization situation, since the search space has many local minima [5]. Also it allows an easier comparison with results from previous experiments for Morse clusters [5], [10].

The pictures shown in this section represent the individuals distribution on the bi-dimensional space – each dot in the picture correspond to an individual. Although several runs were performed with each setting, the pictures presented stand for one single run illustrating the individuals distribution for a specific situation. Results presented are nevertheless representative of the behavior of the algorithm.

5.1. Preliminary Tests

In the first set of experiments, small variations on the tournaments size were made. The algorithm ended after 1000000 evaluations and the individuals were distributed on a 100x100 space with a visibility radius of 10. Figure 4 and Figure 5 illustrates the results observed for the indicated tournaments size. The creation of niches seems to be prejudiced when working with a higher selection pressure. This issue will be carefully analyzed in the next section.

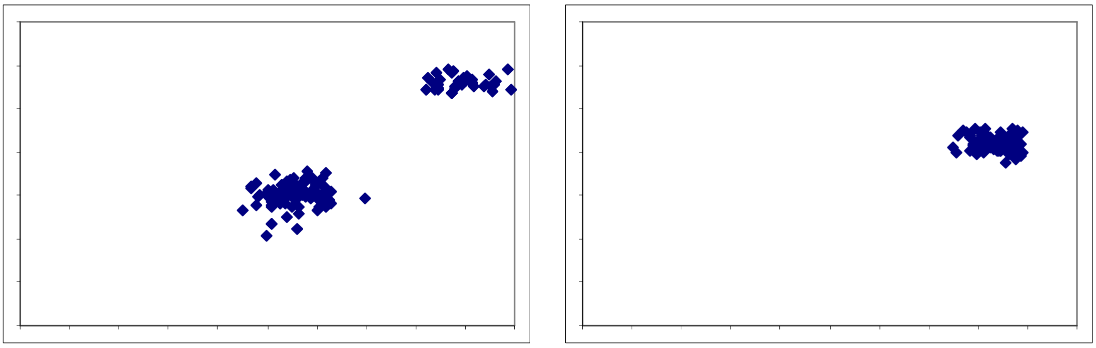

Figure 4 - Tournament 1-2 and 1-3

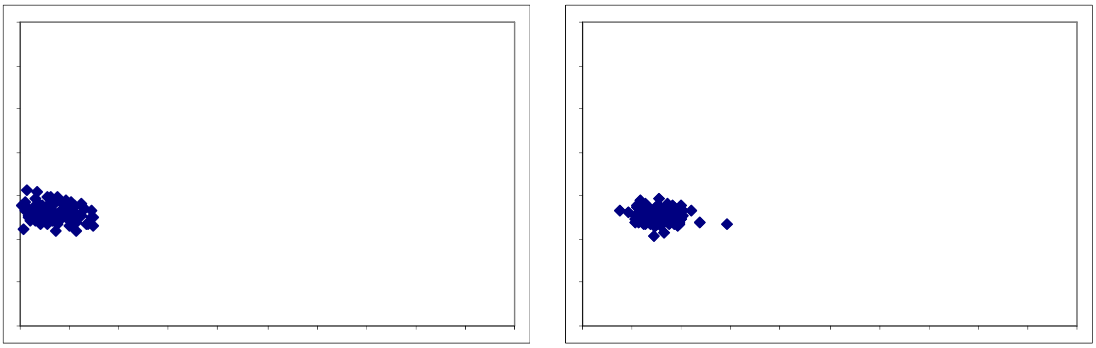

Figure 5 - Tournament 3-1 and 3-3

Beyond the capacity of creating niches, it is important to understand the behavior of the architecture in what concerns to his ability of maintain niches throughout the algorithm cycle. The following experiments were fundamental to identify some parameters with influence on the maintenance of niches. The figures shown correspond to the distribution frames of 100000 evaluations (until the niches are extinguished), sampled from experiments with 2000000 for populations of 100 individuals distributed on 100x100 spaces.

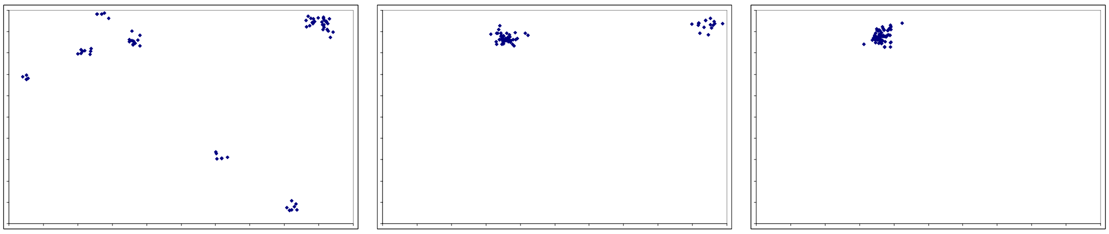

Figure 6 - Visibility 5 / Tournament 2-2

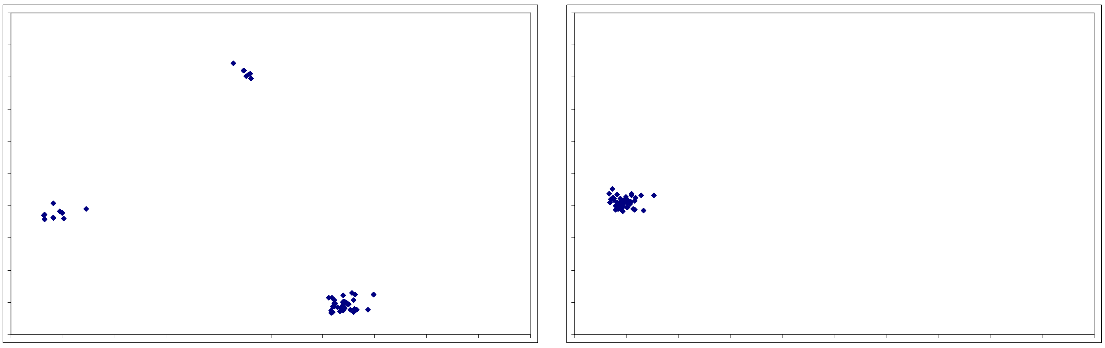

Figure 7 - Visibility 5 / Tournament 3-3

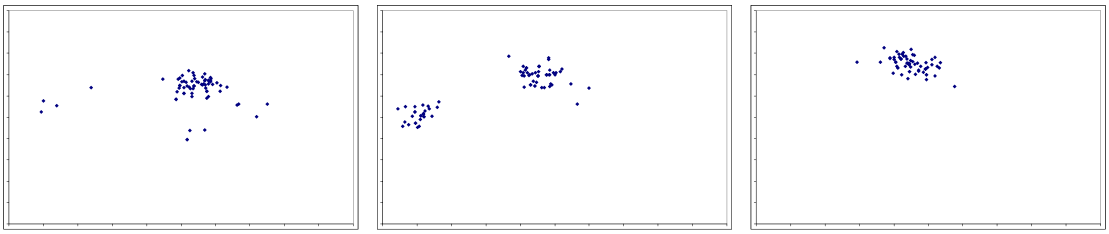

Figure 8 - Visibility 10 / Tournament 2-2

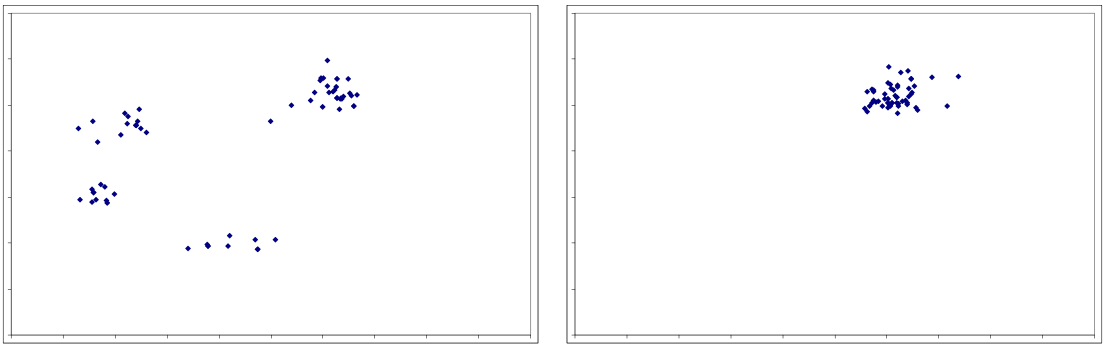

Figure 9 - Visibility 10 / Tournament 3-3

Regarding the results shown in Figure 6, Figure 7, Figure 8 and Figure 9, the visibility radius – or the compromise between visibility and the space dimensions – seems to have direct influence on the proximity between individuals inside the same niche. It is also notorious that the higher is the selection pressure, the faster the whole population is concentrated in the same niche.

In other experiments with higher visibility values (15, 20 and 25), the presence of niches was never detected. One possible justification for such incapacity of creating niches is the fact that a higher visibility radius leads to higher distances between individuals inside the same niche. In consequence, the dimensions of each niche are inevitably increased, compelling the invididuals to scatter themselves all over the bi-dimensional space, instead of creating niches.

The last results presented in this section were sampled from experiments with the same basic configuration of the previous tests, increasing the population size to 200 individuals and fixing the visibility value in 5. Various combinations of tournaments size and space dimensions were tested. The results presented in Figure 10, Figure 11, Figure 12, Figure 13, Figure 14 and Figure 15 are the distribution frames for 100000 and 500000 experiments.

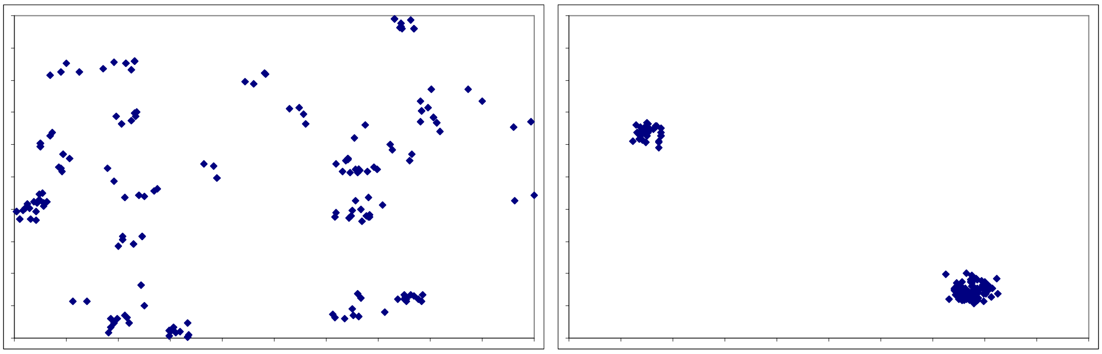

Figure 10 - Space 100x100 / Tournament 1-2

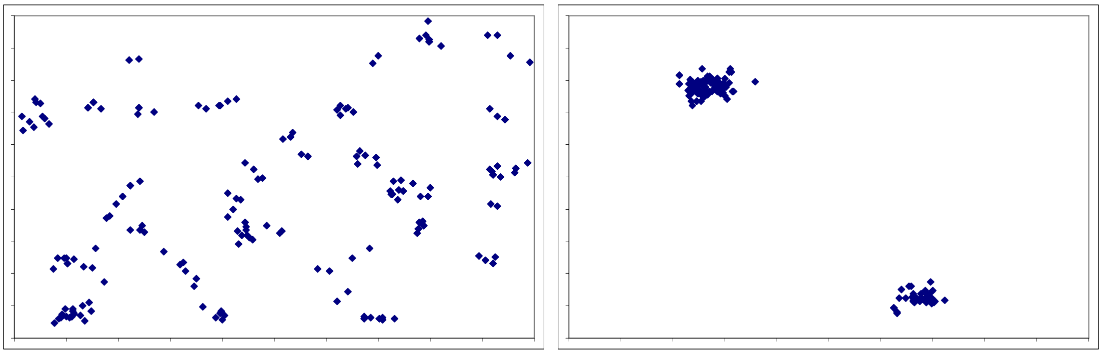

Figure 11 - Space 100x100 / Tournament 1-3

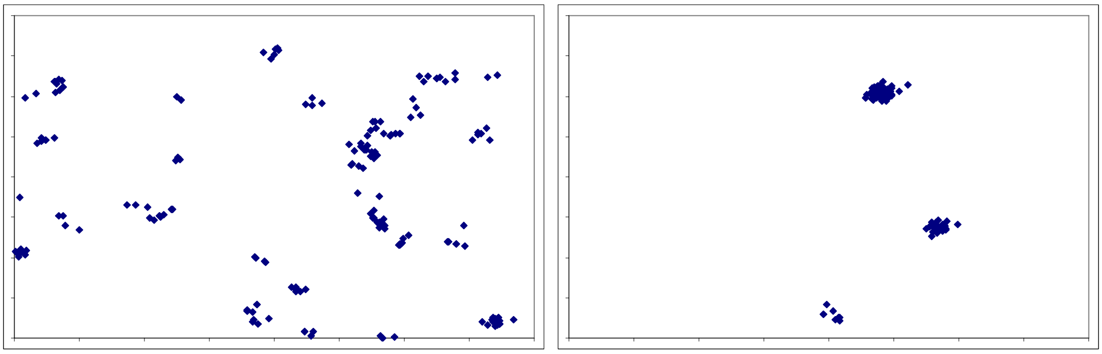

Figure 12 - Space 150x150 / Tournament 1-2

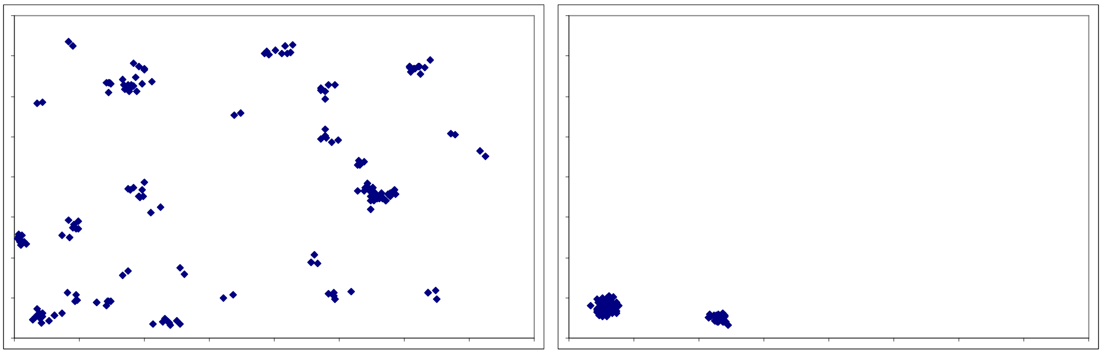

Figure 13 - Space 150x150 / Tournament 1-3

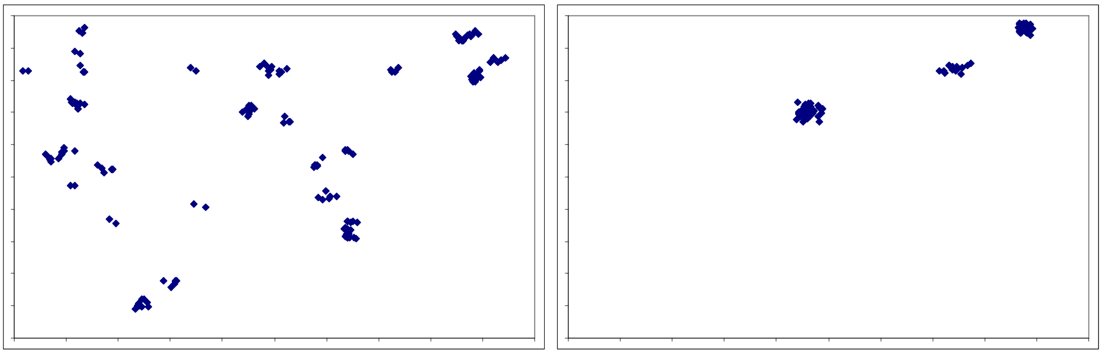

Figure 14 - Space 200x200 / Tournament 1-2

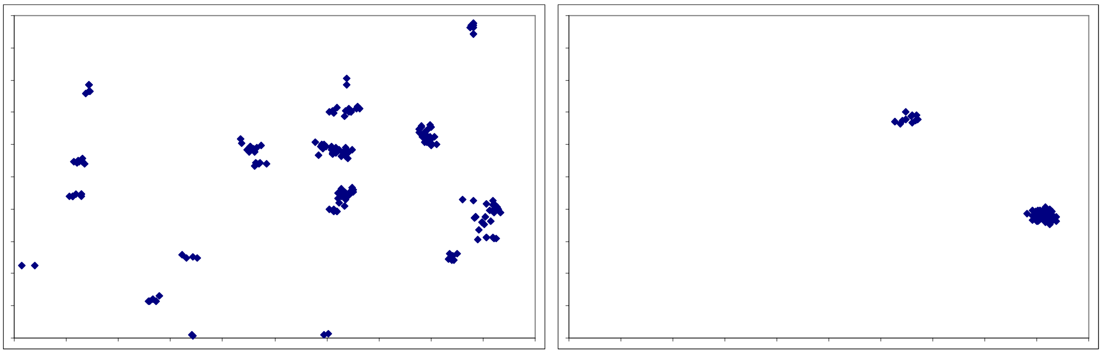

Figure 15 - Space 200x200 / Tournament 1-3

The standard results observed in this set of experiments starts to show a reasonable capacity for maintaining niches throughout the algorithm execution. Although only the first 500000 evaluations were considered, in some isolated runs the niches had only extinguished a few frames later in the 100000 scale. However, the standard behavior observed indicates that the population tends to inhabit the same niche after 500000 evaluations.

As told before, the preliminary tests were essential to the comprehension of the architecture sensibility to the variation of some basic parameters. This test period was also important to perform adjustments in the algorithm – as, for instance, decreasing the weight of the mutation operator when modifying the individuals position.

5.2. Parameter Analysis

In this section all the experiments ran for 4000000 generations and with a fixed crossover rate of 70%. These values were chosen regarding to the previous work with Morse clusters [17]. The number of evaluations was extended for a more accurate observation of niches behavior.

For a better understanding of the results presented in this section, Figure 16 shows the standard evolution of the niches for the specified algorithm lifetime. The individuals distribution is exposed in frames of 500000 evaluations. The following settings were adopted in this base experiment: Population size: 200; Mutation rate 10%; Tournament size: 1-2; Space dimensions: 300x300; Visibility: 5.

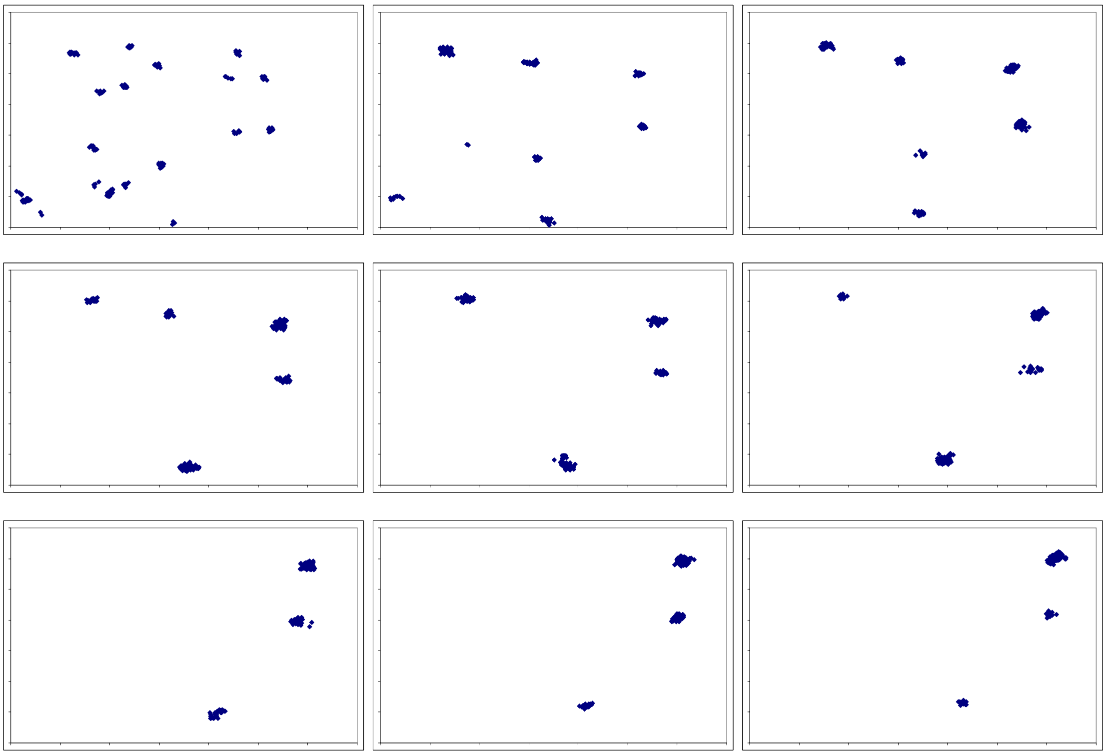

Figure 16 - Standard niches evolution

In the next subsections we will vary some parameters to analyze how they modify the algorithm behavior.

5.2.1. Population Size

For the first analyzed parameter, the following settings were adopted: Population sizes: {50, 100, 200, 250}; Mutation rate: 10%; Tournament size: 1-2; Space dimensions: 300x300; Visibility: 5. The next figures show the last three frames of 1000000 evaluations and illustrate the standard behavior for the indicated population sizes.

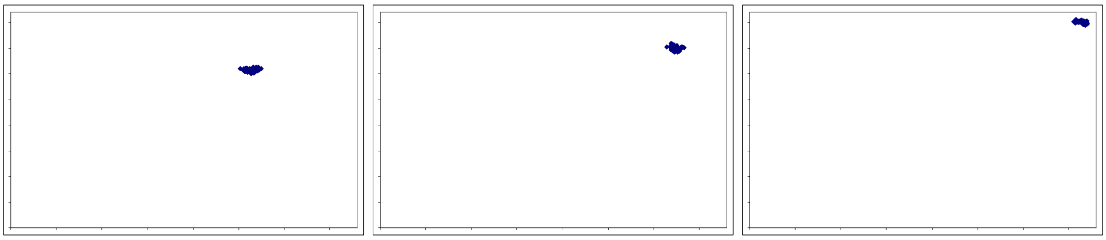

Figure 17 - Population of 50 individuals

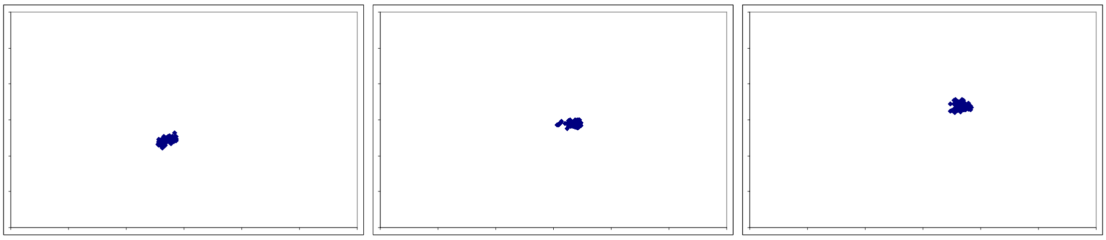

Figure 18 - Population of 100 individuals

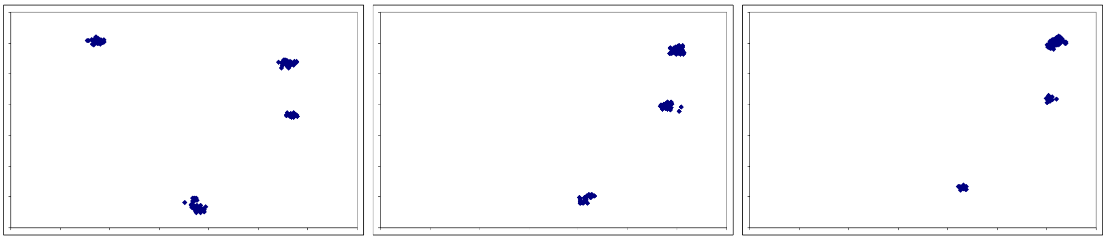

Figure 19 - Population of 200 individuals

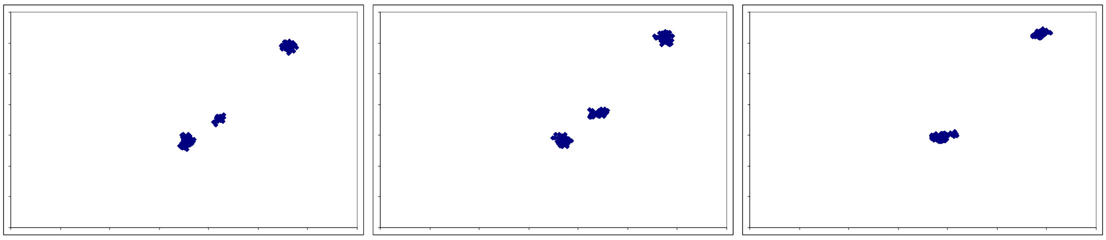

Figure 20 - Population of 250 individuals

The results attained reveal that the niches maintenance is directly related to the quantity of individuals of the population. Like shown in Figure 17 and Figure 18, the standard behavior for the experiments with 50 and 100 individuals was the extinction of the niches before the 2000000 evaluations. This fact was observed in all the repetitions for these experiments. For the 200 individuals population, the standard behavior is presented in Figure 19, with the niches well defined after the 4000000 evaluations. Even so, in a few runs for this experiment the niches disappeared before the end of the execution. The behavior presented for 250 individuals, in Figure 20, was regular in all the runs for that experiment.

5.2.2. Selection – Tournaments Size

The results shown here correspond to the final distribution of the population for the 4000000 evaluations. For each configuration two experiments were chosen to illustrate the standard behavior. For all the experiments, the following settings were taken: Population size: 200; Mutation rate: 10%; Tourney sizes: {1-2, 2-2, 1-3}; Space dimensions: 300x300; Visibility: 5.

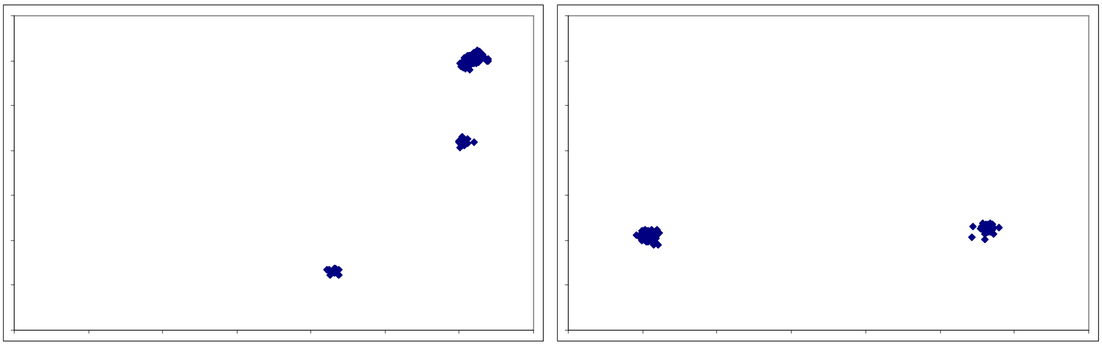

Figure 21 - Tournament 1-2

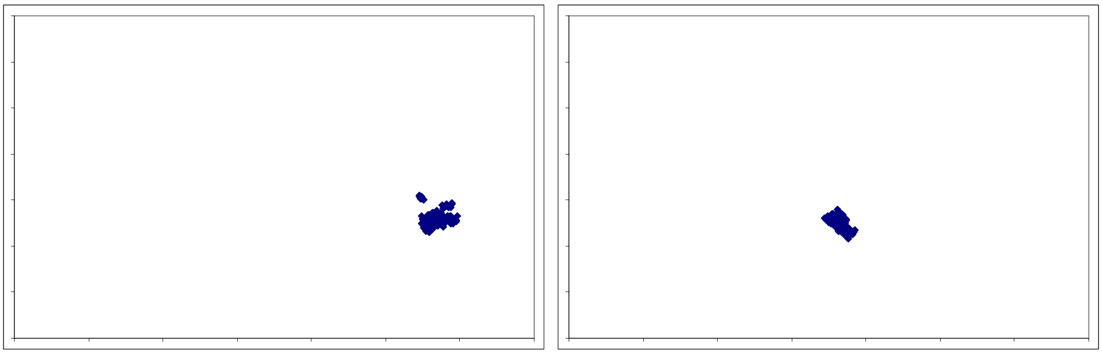

Figure 22 - Tournament 2-2

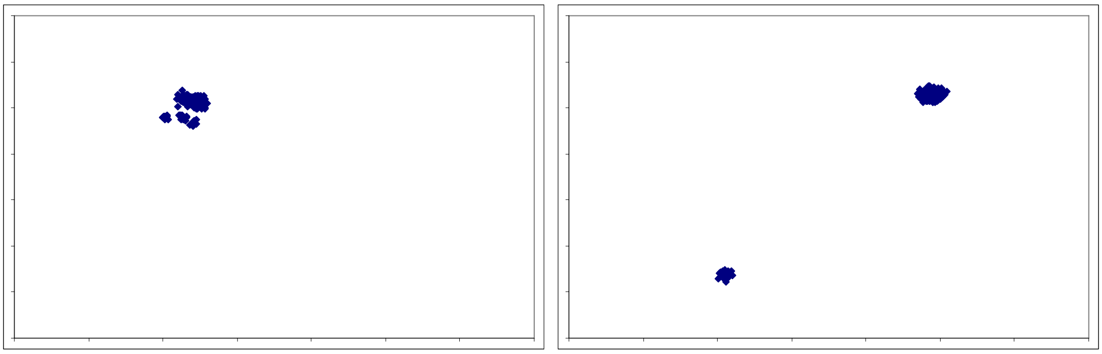

Figure 23 - Tournament 1-3

Regarding the results on Figure 21, Figure 22 and Figure 23, it is perceptible that the selection pressure has some influence on the niches behavior, essentially when the size of the first tournament is increased. In some way, these results confirm the first observations during the preliminary tests. The size of the second tournament, although it has almost no influence on the niches creation, seems to lead the individuals to a faster convergence to specific space locations.

For a better understanding of the influence of the size of the second tournament, the experiments with tournaments 1-2 and 1-3 were repeated for more 30 runs. The distributions of the next pictures correspond to the frames in the beginning, in the middle and in the end of the 4000000 evaluations.

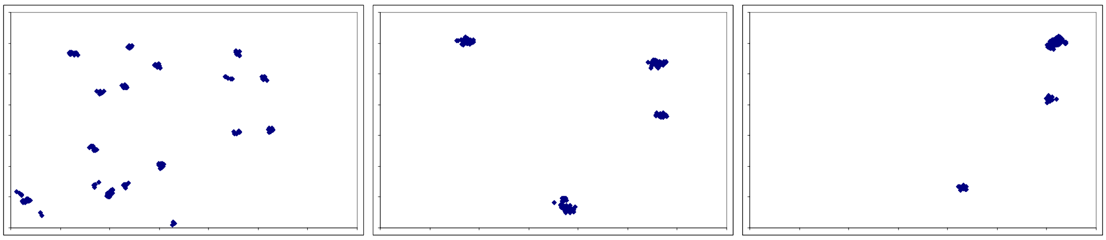

Figure 24 - Tournament 1-2

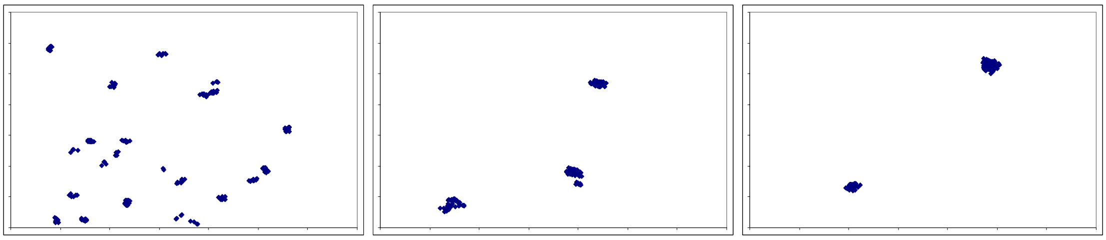

Figure 25 - Tournament 1-3

Figure 24 and Figure 25 show the detected standard distribution for these experiments. Although the distributions might have some slight differences, they are not significant enough to consider that they are influencing the creation and maintenance of niches.

The results presented in this section show that only the size of the first tournament is determinant for the creation of niches. Even so, the size of the second tournament must be regarded, once a higher selection pressure can help the niches to converge to good solutions. On the other hand, if the selection pressure is too high, it might be prejudicial to the diversity inside the niches, leading to a premature convergence.

5.2.3. Mutation Rate

In the experiments performed to analyze the influence of the mutation rate in the niching process, the following settings were adopted: Population size: 200; Mutation rates: {0.01, 0.05, 0.1, 0.15}[1]; Tourney size: 1-2; Space dimensions: 300x300; Visibility: 5. The next results correspond to the standard behavior observed in the last three 1000000 evaluations frames for the indicated mutation rates.

---

[1] Mutation rates are higher than the standard values adopted for evolutionary optimization. Nevertheless previous experiments showed that fairly high rates improve the efficiency of the algorithm when searching for clusters with low PES.

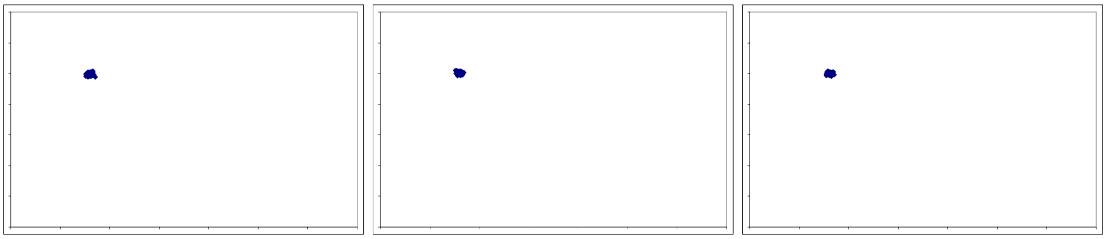

Figure 26 - Mutation rate 1%

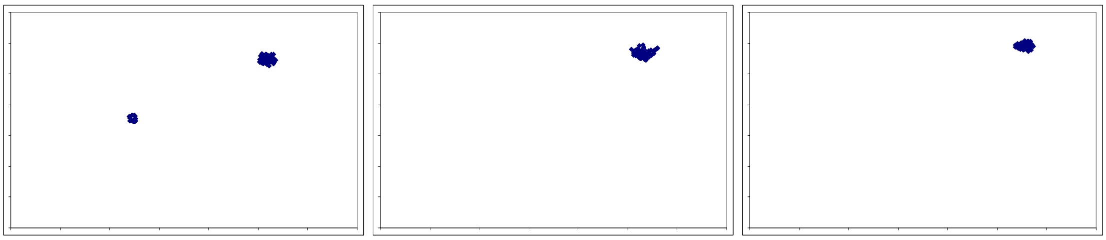

Figure 27 - Mutation rate 5%

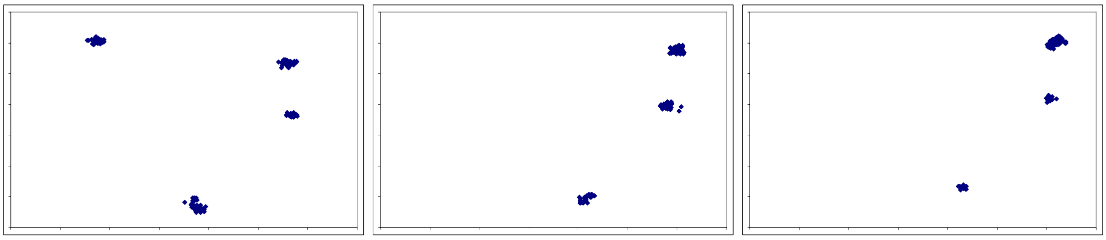

Figure 28 - Mutation rate 10%

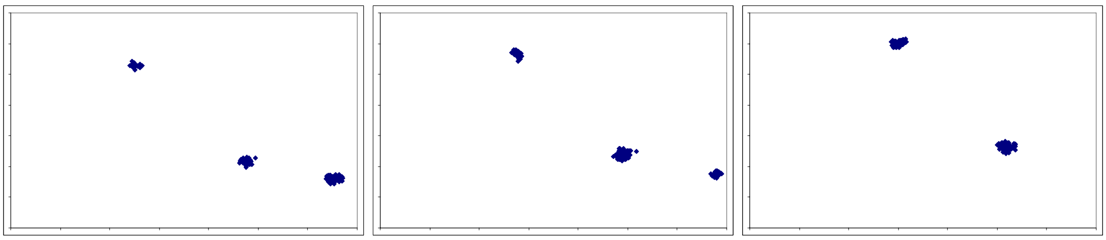

Figure 29 - Mutation rate 15%

The results show that small modifications in the mutation rate can influence the maintenance of niches. When a mutation rate of 1% is used (Figure 26) , niches usually disappear before 1000000 evaluations. For the experiments with the 5% mutation rate, the niches tend to extinguish before the 200000 evaluations, as illustrated in Figure 27. For the last two mutation rates observed, in Figure 28 and Figure 29, the standard behavior is the maintenance of the niches throughout the 4000000 evaluations. Even so, the experiments with a 15% rate showed a more consistent behavior in what concerns the niches maintenance: in all the runs performed with this setting, several niches survived until the end of the optimization.

5.2.4. Space Dimensions and Visibility

The last analyzed parameter is the relation between the space dimensions and the defined visibility radius. Therefore, we opted to fix the visibility value, and vary the space dimensions to analyze the impact of their relation on the niching process. The following settings were adopted for these experiments: Population size: 200; Mutation rate: 10%; Tournament size: 1-2; Space dimensions: {150x150, 300x300, 500x500}; Visibility: 5.

The results shown in Figure 30, Figure 31 and Figure 32 correspond to the standard behavior observed in the last three 1000000 evaluations frames for the indicated space dimensions.

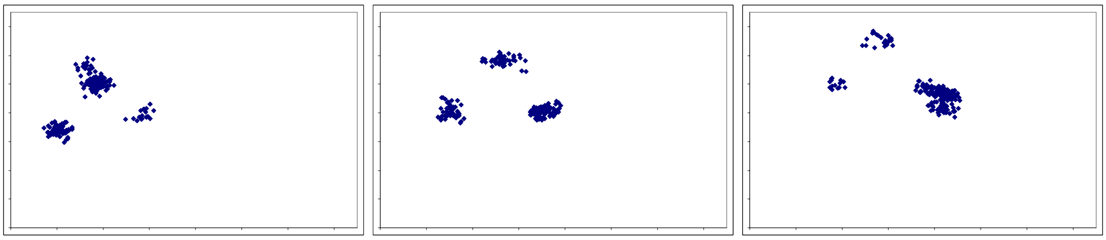

Figure 30 - Space 150x150

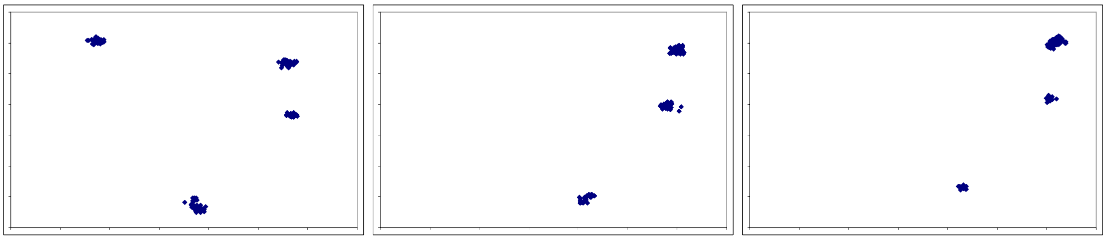

Figure 31 - Space 300x300

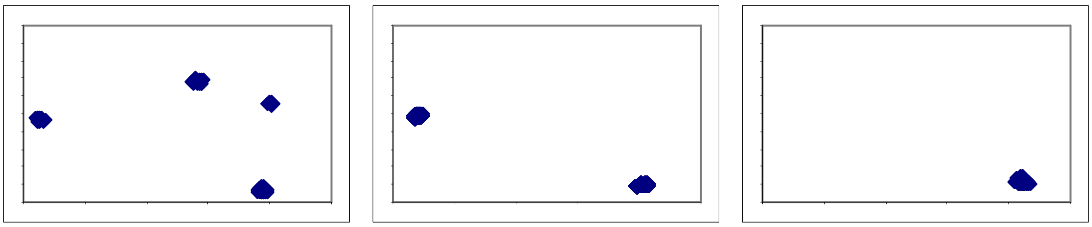

Figure 32 - Space 500x500

In a few experiments with the 500x500 space, the niches prevailed until the end of the execution. Even so, that was not the standard behavior detected. These results show that the space dimensions influence, fundamentally, the proximity between niches. When working on smaller spaces, the disappearing – or fusion – of niches can be dictated by the proximity of niches populating close areas. Although that was not the standard behavior, it was a phenomena observed in a few experiments with space 150x150.

Considering this disadvantage of using small spaces and the standard behavior detected for the 500x500 space, where the niches tend to disappear after the 3000000 evaluations, the space 300x300 and visibility 5 figures out to be the most solid tested configuration.

5.3. Solutions Quality and Statistics

The most significant issue in this section is the ability of the niches to explore different search spaces, thus achieving distinct solutions. To understand how efficiently niches are exploring different solutions, various experiments were made using the default settings. In this section, four experiments (#1, #2, #3 and #4) are analyzed. The easiest way to determine if the search space is different is to compare the best individual and the average fitness of each niche. Once the algorithm is using all his local optimization power, the best solutions found in each niche must correspond to the nearest local optimum.

5.3.1. Solutions Quality

Figure 33 shows the final individuals distribution for two repetitions of the experiment where niches prevailed till the end of the execution. The best and average fitness are shown for each niche.

Despite all the variations tested in the previous sections, the settings used for these experiments are mainly considered the default settings for the algorithm: Runs: 30; Population size: {250, 1000}; Number of evaluations: {4000000, 6000000}; Crossover rate: 70%; Mutation rate: 10%; Tourney size: 1-2; Space dimensions: 300x300; Visibility: 5; Optimization rate: 100%; Optimization length (iterations): 200.

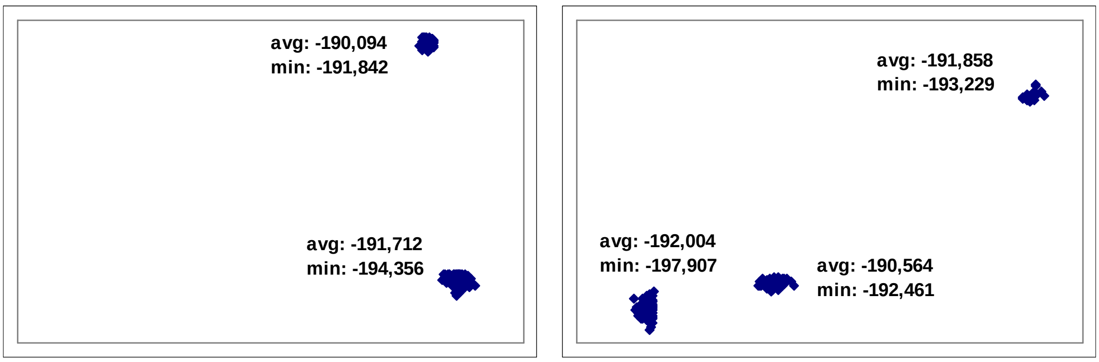

Figure 33 - Experiments with 4.000.000 evaluations (#1)

Figure 34 shows the final individuals distribution for experiments with the same default settings from the experiments before, except for the number of evaluations: 6000000.

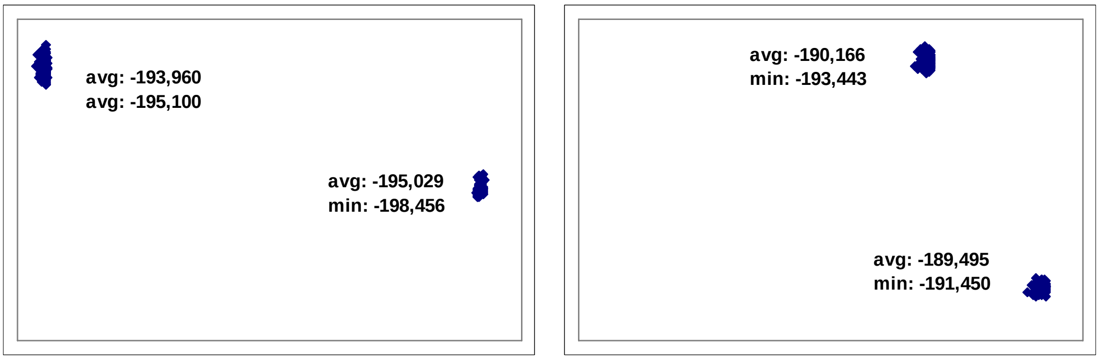

Figure 34 - Experiments with 6.000.000 evaluations (#2)

Regarding these results, for both the tested number of evaluations, different niches seems to explore different portions of the search space. Also the average fitness found in each niche appears to follow the best solution found in the niche, thus emphasizing that each niche is exploring a distinct portion of the solutions space.

The results presented in Figure 35 and Figure 36 were sampled from experiments with the same default settings from the experiments above, but for populations of 1000 individuals. Again, two values for the number of evaluations were experimented: 4000000 and 6000000 evaluations.

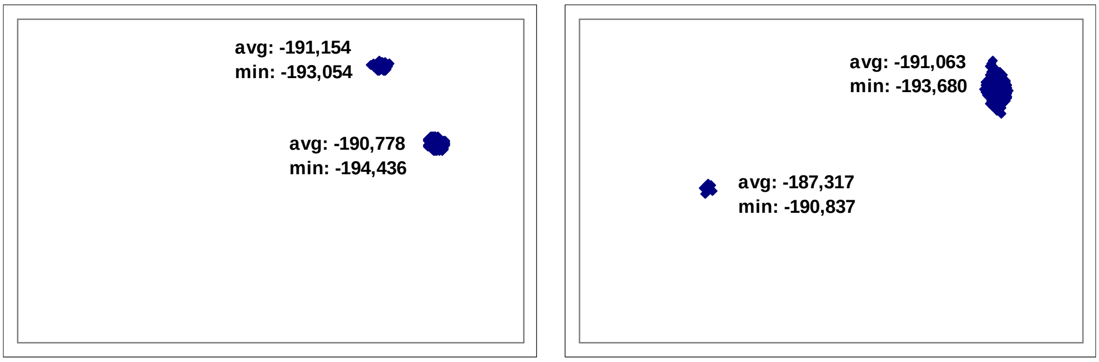

Figure 35 - Experiments with 4.000.000 evaluations (#3)

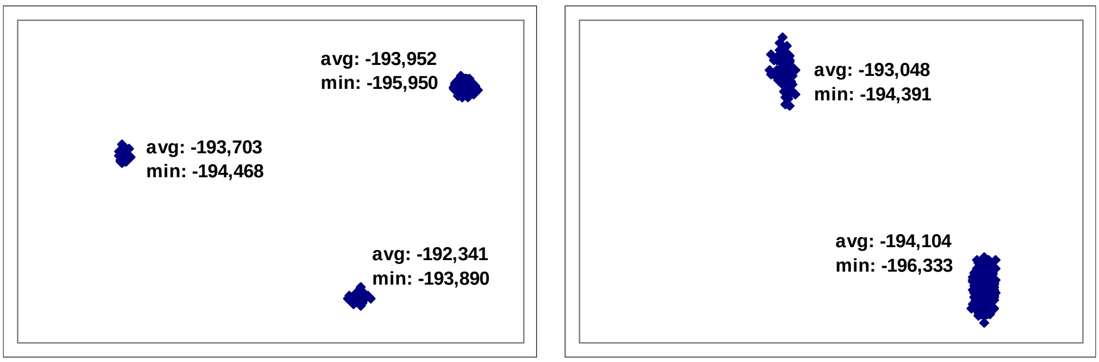

Figure 36 - Experiments with 6.000.000 evaluations (#4)

Like in experiments #1 and #2 (Figure 33 and Figure 34) , in experiments #3 and #4 different niches seem to explore different portions of the solutions space. Like before, the average fitness for each niche tends to follow the best individual found, suggesting that individuals in the same niche are looking for the same solution.

5.3.2. Looking for Good Solutions

The solutions space when looking for Morse clusters is plenty of local minima for each number of particles. However, some specific structures are known as good alternatives to the best solutions [15].

The values presented on Table 1 (column "Sols."), correspond to the fitness of the best known solution and the solutions considered good alternatives for N50. The column "M." contains the number of times that each one of these structures were found in an experiment without niching techniques taken as reference [17] for the analysis presented in this section. The remaining columns map the number of times these values were achieved on the four experiments described in the last section.

Table 1 – Reference structures for N50

Sols.	Morse	Exp #1	Exp #2	Exp #3	Exp #4
-191,240	0	1	0	0	0
-197,907	4	3	2	1	3
-198,456	2	2	1	1	0

Considering the amount of times the algorithm found these specific structures, the chosen reference experiment shows already a good behavior. Only the worst PES cluster was never achieved. The two structures with lower PES were found 4 and 2 times, more than in each one of the four sdGA experiments.

The experiments #1 and #2, for populations with 250 individuals, revealed, as expected, a better performance than the experiments #3 and #4 when looking for these three PES structures. This fact can be explained because these last two experiments ran for populations of 1000 individuals, thus decreasing the number of evaluations per individual.

It is also relevant that the niching architecture tested is capable of outputting a greater variety of local minima than the standard EC algorithm: the sdGA returns all the best individuals found in his visibility radius, while the standard algorithm returns only the best individual of the whole population. When there are well defined niches in the population, the probability of the sdGA to return a large set of local minima increases. All the analyzed sdGA experiments were capable of returning more good quality local minima than the reference experiment. Even if the most of these local optima have not the same importance of the three considered solutions for N50, this ability of returning a large set of good solutions is a central issue for the sdGA functioning and should not be mistreated.

5.3.3. Statistical Analysis

A statistical significance analysis T-Test (significance level of 0.05), between the reference experiment and the four experiments described in the last subsection, was performed. The p-values attained for each experiment are shown in Table 2. Also the best solutions found in the 30 runs for these experiments are presented.

Table 2 – Statistical Analysis

Morse	Exp #1	Exp #2	Exp #3	Exp #4
-191,50	-196,43	-193,44	-191,88	-196,47
-198,46	-194,36	-194,86	-190,89	-196,47
-195,15	-194,92	-192,03	-197,91	-194,82
-195,55	-195,14	-196,47	-194,80	-195,04
-192,71	-195,16	-195,25	-196,95	-196,79
-198,46	-195,25	-195,25	-197,91	-196,47
-197,91	-198,46	-196,77	-196,95	-197,91
-196,95	-197,91	-191,40	-196,02	-195,95
-191,69	-198,46	-192,71	-195,69	-196,33
-195,93	-191,24	-194,97	-196,79	-195,10
-195,08	-195,42	-196,46	-196,46	-196,46
-191,82	-196,47	-197,91	-195,20	-194,47
-197,91	-195,25	-196,45	-195,82	-195,16
-195,77	-195,81	-195,81	-191,67	-196,95
-197,91	-192,35	-191,85	-197,91	-196,95
-194,06	-196,70	-191,68	-196,02	-196,47
-197,91	-195,16	-194,95	-191,73	-195,46
-192,03	-194,80	-195,07	-195,55	-197,91
-196,47	-192,28	-191,37	-195,09	-192,35
-194,71	-197,91	-196,47	-196,45	-197,91
-191,81	-196,45	-197,91	-195,06	-195,81
-195,20	-196,46	-195,43	-196,44	-196,47
-194,71	-195,08	-194,87	-196,46	-195,13
-191,66	-192,71	-191,79	-195,16	-192,35
-196,45	-191,22	-192,35	-196,46	-195,14
-192,28	-197,91	-191,47	-194,82	-196,79
-196,11	-196,47	-198,46	-196,45	-195,46
-191,26	-194,47	-195,14	-197,91	-196,79
-195,55	-195,20	-195,42	-197,91	-192,35
-196,12	-194,86	-195,16	-194,46	-196,70
T-TEST	**0,50759**	**0,56949**	**0,24187**	**0,10289**

Regarding this results, it is clear that there are no significant differences in using the base algorithm or the sdGA approach. The T-Test shows a higher deviation for experiments #3 and #4. This can be due to the fact, already mentioned, that these two experiments were performed using 1000 individuals for same number of evaluations, thus decreasing the evaluations per individual.

6. CONCLUSIONS

In this work we presented some experiments regarding the viability of applying the sdGA approach to a complex problem such as Cluster Geometry Optimization. Results show that this architecture is able to create and maintain niches, thus increasing the likelihood of discovering good solutions in a single optimization run.

Experimental results also revealed some sensitivity to the variation of some algorithm parameters, in what concerns the preservation of niches.

6.1. Parameter Analysis

The population size figured out to have a great influence in the niching process. Thus, increasing the number of individuals also increases the capacity of the algorithm to create and maintain niches. In turn, small populations are not able to maintain niches for half the lifetime of the algorithm, which can be considered an evidently bad performance. The population size was found the parameter with more influence in the niching process.

The tourney size during the selection process is also a considerable influent parameter. The size of the second tournament has almost no influence on the maintenance of niches, although the quality of the solutions inside each niche can be affected by this parameter. On the other hand, the size of the first tournament can be considered critical for the maintenance of niches. The results shown revealed that, for the studied population size, and even when using a low selection pressure, this parameter can seriously compromise the creation of niches. Even so, it remains the doubt about the influence of the tournaments size when using bigger populations.

Another parameter with some influence on the formation of niches is the mutation rate. When using lower mutation rates, the variations on the individuals positions induced by this operator are hindered. The great proximity between individuals inside the same niche observed can be associated with this inhibition. However, it does not seem to be the reason to the rapid extinction of niches, remaining this issue open for future research. On the other hand, using higher mutation rates can lead to a huge dispersion of the individuals at each generation, increasing the possibilities of a niche to extinguish.

The last analyzed parameter, the relation between the space dimensions and the individuals visibility, also revealed some influence on the niching process, even though this influence is more perceptible in the maintenance of niches rather than in the formation process. In contradiction to the initial expectations, the compromise between these two parameters figures out to be the least influent from the analyzed parameters. The tested variations showed that using smaller spaces tends to increase the proximity between niches. Despite the fact that this proximity increases the probability of the niches to fuse themselves with the others, those revealed the capacity to cohabit even in the limits of proximity. In the tested spaces with bigger dimensions, although the niches seem to take some advantage of the protection provided by the hypothetic distance between niches, the niching process seems to be harmed, once the results showed difficulties on creating and maintaining niches. One possible explanation for such incapacity can be, when the algorithm starts, the inevitable dispersion of the individuals, reducing the number of members inside the visibility radius of each individual.

6.2. Solutions Quality

After understanding how far the sdGA architecture was able to run an acceptable number of evaluations for this problem without loosing niches, the main issue turns again to be the initial motivation for this work: the capacity of looking for good solutions simultaneously on cluster geometry optimization.

The results show that the architecture is capable of achieving the initial aim of creating niches (and explore more efficiently the search space) and still look for good solutions. In what concerns the quality of the solutions, it seems that the sdGA can be applied without loosing quality on the optimization. In the specific mission of finding the three reference solutions for N50, there are no clear advantages on using the niches architecture. Even so, the sdGA revealed to be capable of finding a substantial number of local minima with good quality if only the fitness of the individuals is considered.

7. REFERENCES

[1] Deaven, D. M. e Ho, K. - Molecular Geometry Optimization with a Genetic Algorithm. *Phys. Rev. Lett.*, 1995;

[2] Dick, G. - The Spatially-Dispersed Genetic Algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, 2003;

[3] Dick, G. – The Spatially-Dispersed Genetic Algorithm: An Explicit Spatial Model For Gas. In *Proceedings of the Congress in Evolutionary Computation (CEC-2003)*, 2003;

[4] Dick, G. – An Empirical Investigation into Correlation Functions in a Spatially-Dispersed Evolutionary Algorithm. In *6th Annual Colloquium of the Spatial Information Research Centre (SIRC-2004)*, 2004;

[5] Doye, J. P. K. e Wales, D. J. - Structural Consequences of the Range of the Interatomic Potential. A Menagerie of Clusters. *J. Chem Soc. Faraday Trans.*, 1997;

[6] Goldberg, D.E. - *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989;

[7] Hartke, B. - Application of Evolutionary Algorithms to Global Cluster Geometry Optimization. In *Applications of Evolutionary Computation in Chemistry*, *Structure and Bonding*, 2004;

[8] Hartke, B. – Global Geometry Optimization of Atomic and Molecular Clusters by Genetic Algorithms, In L. Spector et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*;

[9] Hartke, B. - Global Cluster Geometry Optimization by a Phenotype Algorithm with Niches: Location of Efusive Minima, and Low-Order Scaling with Cluster Size. *J. Comp. Chem.*, 1999;

[10] Johnston, R. L. – Evolving Better Nanoparticles: Genetic Algorithms for Optimising Cluster Geometries, *Dalton Transactions*, 2003;

[11] De Jong, K. A. - *An analysis of the behaviour of a class of genetic adaptive systems*. Ann Arbor, 1995;

[12] Liu, D. C. e Nocedal, J. - On the Limited Memory Method for Large Scale Optimization. *Mathematical Programming B*, 1989;

[13] Morse, P. – Diatomic Molecules According to the Wave Mechanics. II. Vibrational Levels. *Phys Rev.*, 1929;

[14] Roberts, C., Johnston, R. L. e Wilson, N. - A Genetic Algorithm for the Structural Optimization of Morse Clusters. *Theor. Chem. Acc.*, 2000;

[15] Wales, D.J. et al. - The Cambridge Cluster Database, URL: http://www-wales.ch.cam.ac.uk/CCD.html, accessed between January and December 2006;

[16] Zeiri, Y. - Prediction of the Lowest Energy Structures of Clusters Using a Genetic Algorithm. *Phys. Rev*, 1995;

[17] Pereira, Francisco B., Marques, M. C., Leitão, T., Tavares, J. – Analysis of Locality in Hybrid Evolutionary Cluster Optimization. In *Proccedings of the Congress in Evolutionary Computation (CEC-2006)*, 2006;