A Self-Adaptive Local Search Coordination in Multimeme Memetic Algorithm for Molecular Docking

Pablo F. Leonhart[0000−0002−7372−7919], Pedro H. Narloch[0000−0002−8142−9399], and Márcio Dorn[0000−0001−8534−3480]

Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
[mdorn]@inf.ufrgs.br

Abstract. Molecular Docking is a methodology that deals with the problem of predicting the non-covalent binding of a receptor and a ligand at an atomic level to form a stable complex. Because the search space of possible conformations is vast, molecular docking is classified in computational complexity theory as a `NP`-hard problem. Because of the high complexity, exact methods are not efficient and several metaheuristics have been proposed. However, these methods are very dependent on parameter settings and search mechanism definitions, which requires approaches able to self-adapt these configurations along the optimization process. We proposed and developed a novel self-adaptive coordination of local search operators in a *Multimeme Memetic Algorithm*. The approach is based on the *Biased Random Key Genetic Algorithm* enhanced with four local search algorithms. The self-adaptation of methods and radius perturbation in local improvements works under a proposed probability function, which measures their performance to best guide the search process. The methods have been tested on a test set based on `HIV`-protease and compared to existing tools. Statistical test performed on the results shows that this approach reaches better results than a non-adaptive algorithm and is competitive with traditional methods.

Keywords: Self-adaptation · Multimeme Memetic Algorithms · Molecular Docking

1 Introduction

Molecular Docking (`MD`) is a computer-aided approach used in Drug Discovery to predict the conformation of a small molecule (ligand) inside a larger molecule binding site (receptor), measuring the binding affinity between these molecules. Due to the massive number of possible conformations that a molecule can assume, the `MD` problem is considered as a `NP`-Hard one by computational complexity theory. Thus, the development of search methods with the capability to explore the conformational search space is essential. Due to the complexity related to the conformational search space, exact methods are not efficient since

they can not retrieve a solution in a viable time. In this manner, metaheuristics, like *Memetic Algorithms* (`MA`), employing algorithms as global and local search procedures, became interesting approaches for finding good solutions in a possible computational time. Regardless of the possibility of using metaheuristics to explore the search space, they are very dependent on parameter setting and search mechanism definition. In this way, self-adaptive mechanisms are reliable techniques to give the algorithm the option to self-decide which parameter, or search mechanism, should be used along the optimization process [1].

Besides the search mechanism definition, another important component of `MD` methods is the energy function responsible for describing the interaction between receptor-ligand, evaluating different physicochemical aspects related to the binding process. Also, the scoring function is important for the search method, since it is used to distinguish and rank different solutions regarding energy. Different scoring functions are available in the literature [2], but the `RosettaLigand` scoring function had achieved interesting results in the last years [3][4] in `MD` predictions. In light of these facts, we propose an `MD` method based on `RosettaLigand` scoring function. To explore the conformational search space, a novel self-adaptive multimeme algorithm based on *Biased Random Key Genetic Algorithm* (`BRKGA`) enhanced with four local-search algorithms is evaluated and compared with state-of-art methods. Our contributions are related to the self-adaptive mechanism used to select the local search strategy to be applied, and the radius perturbation for each docking case. Results obtained showed that our approach reaches better energy results in comparison with the non-adaptive `BRKGA` version and traditional methods such as `AutoDock Vina` [5], and `jMetal` [6]. The remaining of this paper is organized as follows: Section 2 describe the main concepts related to the molecular docking problem and related works. Section 3 presents the proposed self-adaptive approach. In Section 4 are analyzed the achieved results. Finally, Section 5 presents the conclusions and future works from this research.

2   Preliminaries

Molecule Flexibility: An important aspect related to the `MD` problem is the flexibility associated with each molecule. It is possible to classify the `MD` approaches in rigid-ligand and rigid-protein; flexible-ligand and rigid-protein; and flexible-ligand and flexible-protein. These approaches dictate how the problem is computationally encoded. Therefore, the molecule flexibility is directly related to the problem's dimensionality, where more flexible molecules have more degrees of freedom. Thus, leading to a more complex optimization search space. For the flexible-ligand with rigid-receptor scheme (used in this work), a possible solution for the problem can be described by $7 + n$ variables, where three of them are the ligand translation values (T_x, T_y, and T_z), followed by four values corresponding the ligand's orientation (Q_x, Q_y, Q_z, and T_w), and n varies according to the dihedral angles present in the ligand.
Scoring Function: In order to compute the binding affinity between a ligand and a receptor, scoring functions are used to score and rank the steric and elec-

trostatic interactions of both molecules. As the energy measurement is a computationally expensive operation [7], approximation methods have been used. It is possible to categorize these functions in force-field based, empirical and semi-empirical, and knowledge-based functions [8]. In this paper, we have used a knowledge-based energy function known as `RosettaLigand`, which considers statistical analysis of the *Protein Data Bank* (`PDB`) in its composition [3] in addition to physicochemical terms [4]. The `RosettaLigand` score function was tested in different comparative studies, showing good results when compared with well-known scoring functions such as `AutoDock` [3].

Related Works: The multidimensionality and complexity of the conformational search space avoid that any known computational technique optimally solves the `MD` problem. Many methods based on metaheuristics have been applied to attempt to get optimal solutions to these problems. The major of algorithms utilized in `MD` are *Genetic Algorithms* (`GA`), *Differential Evolution* (`DE`), and *Particle Swarm Optimization* (`PSO`). Hybrid strategies also have been applied, like in Rosin et al. [9] where was implemented a `GA` hybridized with *Simulated Annealing* (`SA`) and *Solis-Wets* as `LS` strategies to explore the binding conformational search space. In Tagle et al. [10] three variations of `SA` were employed as local search procedure in a *Memetic Algorithm*. Recently, Leonhart et al. [11] proposed a hybridization of `BRKGA` with three variations of *Hill-climbing* and `SA` algorithms working as `LS` chained operators applied in test cases of `HIV`-protease. The approach proposed by Krasnogor et al. [12] and applied to bioinformatics problems (prediction of structures) is also relevant to highlight. The authors developed a simple, but efficient, inheritance mechanism (`SIM`) for a discrete combinatorial search. The strategy consists of encoding the memetic material in the individual's representation, where this material indicates the `LS` to be applied. During the evolution, the crossover is responsible for choosing what method attach to the offspring, according to the parent with better fitness. Jakob [13] proposed another self-adaptive approach to schedule local search methods over a function probability. In this idea, all `LS` have equal chance to be selected at the beginning of the process. During the evolutionary process, the probabilities of applying each algorithm are updated, according to the relative fitness gain and the required evaluations considered. In Domínguez-Isidro et al. [14] is proposed an adaptive local search coordination, based on a cost-benefit scheme, for a multimeme `DE` for constrained numerical optimization problems. Also, in Jin et al. [1] is proposed a `MA`, combining `GA`, `DE`, and *Hill-climbing*, working under a strategy that use weights to measure the contribution of the algorithms according to their improvement over the individuals of the population and in the stage of evolution. In this study, we investigate the possibility of applying a self-adaptive memetic algorithm, based in a proposal probability function and inspired on the `BRKGA` to coordinate the use of different local search methods, and exploring different neighborhoods by adapting the radius perturbation.

Memetic Algorithms: A Memetic Algorithm (`MA`) is an evolutionary algorithm which is composed of global and local procedures [15]. Global search algorithms can explore the whole search space, while, the exploitation of a neighborhood

solution is attributed to a local search method, which can obtain good precision [16]. They are inspired by Darwinian's principles of natural evolution and Dawkins' notion of a *meme*, defined as a unit of cultural evolution that can perform local refinements.

Biased Random Key Genetic Algorithm: In this study, our global search strategy is inspired on the *Biased Random Key Genetic Algorithm* (BRKGA) initially proposed by Gonçalves et al. [17]. The algorithm combines aspects of GA and random keys to encode the solutions by real values. In our approach, however, we adopt real-coded values to represent each gene, instead of the keys. In BRKGA, the set of solutions is ordered by fitness value and divided in *castes*, called *elite* and *non-elite* groups. The initial population is randomly generated and the individuals ordered in the castes. The crossover operator gets one parent from each set to create the offspring following a probabilistic parameter, that prefers more genes from the elite group. The mutation procedure is responsible for generating new individuals instead of modifying them. Thus, the next population will be composed by the caste 'A' (elite set) from the current population; the caste 'B', formed by all offsprings; and the caste 'C', which are the mutated solutions. Then, the new set is ordered to update the individuals in the castes.

Search Space Discretization: The MD problem needs a definition of the search space around the binding site of the receptor. In this work, we adopt the idea proposed by Leonhart et al. [11] representing this area as a cube formed by smaller cubes, where the central atom of the ligand defines the center of this box. Thus, with this point and the area's volume is possible to set the size of smaller cubes. This division has the objective to explore the whole ligand-receptor binding search space better. At the beginning of the memetic execution, the population is generated and equally distributed between the subareas of the cube. To keep the diversity, at least one individual must be in each area. This feature builds a local competition, where the better solutions dispute location between them. Procedures like crossover, mutation and local search operators could move solutions between regions, following the evolutionary process, but respecting the percentual of individuals non-migrants. The crossover can create offsprings from different positions of their parents according to the cube of each one. In mutation process, eventually, are filled empty cubes, and generated solutions in any area of the search space. Also, the local search methods can modify solutions of a subcube, depending on the approach to visit the neighborhood solutions.

Local Search Methods: Local search is a heuristic method which moves from solution to solution in the candidate's space by performing local changes until getting a local optimum or reach another limit. However, this movement is only possible if a neighborhood relation is defined in the search space. The neighborhood generation is made taking into account three aspects: (i) order of gene visitation, i.e., which position would be first considered to explore; (ii) radius perturbation, what means how much each gene must be modified; and (iii) the direction of search, that indicates if the radius value would be added or subtracted from the value encoded in the individual.

The *Hill Climbing algorithm* (HC) [18], also known as descent improvement, is an

old and simple local search that starts with an initial solution and replace it, at each iteration, by the best neighbor found that improves the value of objective function. The procedure ends when there is no solution better than the current one, reaching a local optimum. This algorithm presents variations in the way that the neighborhood is explored. The order of solution's generation (deterministic or stochastic) is one of them, as well as the strategy selection of the next solution, known as a pivoting rule. We highlight three variants of `HC` (pivoting rules) to select the best neighbor: (i) *Best Improvement* (`BI`), which performs a fully and deterministic search in the whole neighborhood. This strategy could be time-consuming because it evaluates all the movements, although ensuring the selection of the best solution in every iteration. (ii) *First Improvement* (`FI`) this variation swap the current solution by the first best neighbor found. The neighbors are evaluated in a deterministic way following a pre-defined order of candidates' generation. The approach is faster than `BI` because it does not visit the whole neighborhood, just in the worst case, meaning the end of improvements. (iii) *Stochastic Hill Descent* (`SHD`) is similar to `FI`. The swap rule of solutions is the same, but the order of neighbors' visitation is randomized at each iteration of the process. This characteristic ensures an equal selection of candidates from all regions of the search space.

Another local search method is the *Simulated Annealing* (`SA`), a stochastic algorithm which accepts, in some moments, worst solutions [19]. The objective is to avoid local optimums and delay the search convergence. The process starts with an initial solution and at each iteration generates a random neighbor. The candidate is accepted if better than the current solution, and if it is worst, there is a rule with a decreasing probability to allow this solution. Thus, the procedure begins running like a random walk accepting many solutions, but when the probability decreases, the method is more similar to the `HC` algorithm.

A recent feature in LS is the concept of *Local Search Chains* proposed by Molina et al. [20]. The idea is to share the state of search between different local search's applications. The `LS` call may not explore all neighborhood of a solution, so the final strategy parameter values achieved by the candidate will be the initial state of a possible subsequent local search application in this individual. This strategy allows the `LS` operators to be extended in some promising search zones, avoiding different algorithms to evaluate the same candidates by chaining the searches. According to [20] there are some aspects to consider in the LS chains' management. A fixed intensity search is one of them, `LS` intensity stretch (I_{str}), which ensures that every local search algorithm has the same computational effort applied. Another one is to save the configurations (visitation order) that guided the search and the current state at the end of the call. In the `HC` methods, for instance, the last neighbor generated is saved in all variants, in the `FI` and `SHD` approaches are also kept the visitation's order of the neighborhood.

Adaptive Memetic Algorithms (`MAs`): Recent studies have applied hybridization with adaptation in `MAs`. The adjustment of parameters and operators has presented a promising area of computation in this evolutionary algorithm. This approach can self-adjust to a given problem without previous knowledge by utilizing acquired information to adapt itself with the search progress. The challenge

to design a robust and efficient `MA` has some questions to be considered, like: (i) where and when the local search should be applied; (ii) which individuals should be improved and how must be chosen; (iii) the computational effort required in each `LS` call; (iv) the equilibrium of ratio between global and local search. Several adaptive memetic algorithms have addressed these questions [1,12,13].

Multimeme Memetic Algorithm (`MMA`): The *Multimeme Memetic Algorithm* (`MMA`) was originally proposed by Krasnogor and Smith [21]. In a `MA` only one `LS` method is used, while in the `MMA` a set of local searchers is employed. The idea of the algorithm is self-adaptively choose from this set which method to use for different instances, phases of the search or individuals in the population. In their approach, the individual was represented by its genetic and memetic material, where the last one specifies which meme will be used to perform local search.

3   The Proposed Method

In this paper, we propose and test a `MMA` algorithm for the `MD` problem. Our algorithm self-adaptively choose which local search operators to apply in different individuals. We represent a solution only by its genetic material, i.e., encoding values to perform conformational operations with the ligand structure,which includes the translation and rotation of the molecule, and internal angle rotations. Thus, from the definition of our solution representation, we have a neighborhood size equal to $2n$, where n represents the genes' number in an individual. We have adapted and implemented the three variations of the HC algorithm and the Simulated Annealing (for more details see section 2). Also, we incorporate the LS chaining to enjoy best the benefits of each algorithm applied. The main contribution of this study is the self-adaptive coordination of the local search methods and the radius perturbation, which works under a probabilistic rule composed of two main elements, inspired in the studies of Jakob [13,22] and Domínguez-Isidro et al. [14]. Equation 1 show these two terms: the ratio fitness gain (rfg), and the ratio of success application (rs_{app}). The first element it is a measurement of how much each `LS` operator improves the fitness of solutions, considering the history of values reached and the effort applied in each one. The idea is to ponder `LS` operators (`LSO`), performing few iterations and considering improvements in the individuals. The second term is simple and represents the benefit of the `LS`, by counting how many times solutions were improved.

$$rfg = \left| \frac{f_{original} - f_{LS}}{f_{original} - f_{mean}} \right| \qquad rs_{app} = \frac{n_{improvements}}{n_{app}} \qquad (1)$$

The rfg is obtained from the `LS` application to one individual, where $f_{original}$ is the individual fitness before the local improvement, f_{LS} is the fitness reached by the algorithm, and f_{mean} is the fitness average from all solutions of the caste 'A' in the moment of `LS` call. The objective is to measure the effectiveness of the local operator over a solution comparing the improvement with the best individuals from the population at that moment. The rs_{app} is the ratio between the number of improvements ($n_{improvements}$) and the number of applications (n_{app}) of a given `LSO`. To keep the methods competitive and cooperative, inspired

in the study of Jin et al. [1], we adopt *weights* to measure the contribution of each algorithm, the Equation 2 shows the proposal:

$$weight_{rfg} = \left(\frac{\sum_{i=1}^{n_{app}} rfg_i}{n_{app}} \right) * w_{rfg} \qquad weight_{rs_{app}} = rs_{app} * w_{rs_{app}}$$

$$weight_X = weight_{rfg} * weight_{rs_{app}}$$
(2)

To measure the effectiveness of a `LS` we consider the history of improvements by calculating the average rfg and multiplying by a fixed value, represented by w_{rfg}. In the same way, at the success ratio is attributed another weight ($weight_{rs_{app}}$). These weight values added is equal to zero. Finally, with the multiplication of both, we get $weight_X$, where X is any `LS` operator. After a local search application, the respective weight is calculated, and then the probabilities of each one are obtained by Equation 3 shows:

$$prob_X = \frac{weight_X}{\sum_{j=1}^{n_{LS}} weigth_j}$$
(3)

The probability of a given local search is obtained by calculating its contribution percentage in the weights of all `LS` methods (n_{LS}). Thus, each operator has a value between 0 and 1, allowing to mount a roulette wheel to sort and obtain an algorithm to be applied. It is important to highlight that at the beginning of the `MMA` execution all `LSO` has the same evaluations number to fairly calibrate the initial weights. Along the execution, the best method will get a higher probability, so having more opportunity to be applied. Algorithm 1 shows the pseudo-code of our approach. The input values P, P_e, P_m, n, I_{ls} represent the population size, the elite size, the mutant population size, the number of individual genes, and the minimum number of individuals to run a `LS` call, respectively. During the crossover and mutation processes, if the intensity stretch of global search is reached, the local search function is called. The individuals to be improved are selected from the most populated cubes (see space discretization in section 2) . After each `LS` application, its contribution is calculated and the probabilities adjusted for the next call. The execution's output is the best solution found until satisfies the maximum of energy evaluations.

4   Experimental Results

To test our `MA` approach, we use a test set with 16 complexes containing receptor and ligand molecules. We perform the tests into the following stages: (i) comparison of the proposed `MMA` with other two self-adaptive algorithms considering a fixed radius perturbation; (ii) the `MMA` approach with three variations of radius search; (iii) finally, the self-adaptive algorithm was compared with other methods in the state-of-art.

Benchmark: The 16 selected structures are based on the `HIV`-protease receptor and was previously classified in [23]. The set was obtained from the `PDB` database and divided into four groups, following the ligand structure size. The `PDB` code and the range of crystallographic resolution in Ångströms (Å) are:

Algorithm 1 Pseudocode of the developed `MMA` algorithm.

Input: P, P_e, P_m, n, I_{ls}
Output: Best Solution X
1: $P \leftarrow$ initialize with n vectors of real-coded values
2: **while** not reach the maximum of energy evaluations **do**
3:      Evaluate solutions of P and divide in P_e and $P_{\bar{e}}$
4:      $P^+ \leftarrow P_e \bigcup crossover() \bigcup mutation()$ ▷ Making the next population
5:      **if** currentEvaluations is multiple of `LS` intensity **then**
6:          $prepareLocalSearch()$ ▷ Individuals improved are introduced in P^+
7:      **end if**
8: **end while**
9: **return** best solution X

Local search preparation

10: Make a list of potential candidates to apply `LS`   ▷ According to I_{ls} value
11: **while** not enough evaluations in the I_{str} **do**
12:     individual = getNextIndividual()   ▷ Iterating over the previous list
13:     method, radius = chooseOperators()   ▷ Sort values to determine which one
14:     $runLocalSearch(individual, method, radius)$
15:     rfg and rs_{app} are calculated   ▷ According to method and radius perturbation
16:     updateProbabilities()   ▷ Probabilities of `LSO` are updated under the weights
17: **end while**

$small$ (`1AAQ`, `1B6L`, `1HEG`, `1KZK`, $1.09 - 2.50$); $medium$ (`1B6J`, `1HEF`, `1K6P`, `1MUI`, $1.85 - 2.80$); $large$ (`1HIV`, `1HPX`, `1VIK`, `9HVP`, $2.00 - 2.80$) size inhibitors, as well as $cyclic\ urea$ (`1AJX`, `1BV9`, `1G2K`, `1HVH`, $1.80 - 2.00$) inhibitors. `PYMOL` was used to remove molecules such as solvent, non interacting ions, and water from the target structures.

The *AutoDockTools* was used to add partial charges and hydrogens to structure, as well, to define the maximum number of torsional angles in the ligand. This maximum number of torsions was set to 10, but can be less according to the ligand structure/size, and their selection considers the angles that fewer move atoms, keeping freeze the ligand center [23]. After, the *Open Babel tool* [24] was used to convert and generate necessary files to be manipulated by our algorithm and then used by the *PyRosetta* to load the complexes and evaluate the energy score. As already mentioned, the search space was represented by a cube including the binding site of the receptor, the size of this box was set in 11 Å for each axis, and the grid spacing defined as 0.375 Å. From this definition, the initial ligand conformation and position in the cube are randomized for each algorithm run, to take no advantage of the known crystal structure.

Parametrization: Considering the random values for the ligand in each run is fair that every algorithm starts from the same point. Thus, for each instance in each run, the initial population and the ligand structure was the same. In `BRKGA` we adopted the recommended values to the parameters according to [17]. The population size is 150 individuals, where 20% is the elite group, 30% is the mutation set, and the remaining is crossover offsprings. The elite allele inheri-

tance probability on the random choice is $0.5 - 0.7$ to crossover operation and the percentual of individuals non-migrants is set to 30% also. The LS parameters were defined as 0.5 to radius perturbation, in the first experiments, after this value varied in 0.1 and 0.25 also. The intensity stretch was equal to 500, and the global/local search ratio equal to 0.5, following the values adopted in [20]. Finally, the MMA weights were defined as 0.6 to the historical ratio fitness gain and 0.4 to the ratio of LS success application. Since the use of a stochastic method, we execute 11 (first and second stages) and 31 (third stage) independent runs per instance with a stop criterion of 1,000,000 energy evaluations. So, we are acquiring statistical confidence in the present results.

Achieved Results: We divide our tests into three stages described as following. Firstly, we ran the MMA approach, working under our proposal probability function, and compared with another two simple self-adaptation methods. The SIM technique [21] is one of them, and the other one is a random selection of the LS operators at the moment to be used. In the SIM approach we start the population equally distributing the LS methods to each solution, and during the iterations, the crossover and mutation procedures changes this memetic material. The random approach just sorts a method when the local search must be applied. All algorithms were applied over half of the instances with 11 runs each one. In this first test, we are interested in verifying the behavior of the approach based on probabilities against the inheritance and random methods. It is important to note that the radius perturbation is fixed in 0.5 to every gene. This value was chosen empirically in preliminaries tests with the generation and evaluation of random individuals. We evaluated the results by its energy in *kcal/mol* and with the *Root Mean Square Deviation* (RMSD), which is a structural measure of the ligand get by the algorithm with the crystallographic experimental structure of the molecule. Table 1 presents the best energy found and the corresponding RMSD, as well as the average and standard deviation of each one, based on the best solutions generated in each configuration.

Table 1. Achieved results from the MMA, based on the probability function, with an inheritance (SIM) and random (RAND) approaches. Cells highlighted in gray shows the best solution obtained, and cells in blue, the best solutions average, for each instance.

ID	Method	Best solution		11 runs average		ID	Best solution		11 runs average	
		Energy	RMSD	Energy	RMSD		Energy	RMSD	Energy	RMSD
1AJX	RAND	-250.16	1.02	-242.06 ± 3.61	10.49 ± 4.04	1HPX	-356.80	2.88	-349.66 ± 3.61	4.86 ± 1.73
	SIM	-250.64	1.40	-243.21 ± 4.17	9.11 ± 4.47		-355.18	2.25	-349.30 ± 2.77	5.10 ± 1.36
	MMA	-250.20	3.21	-241.34 ± 2.86	11.48 ± 2.68		-357.00	2.73	-350.15 ± 3.90	4.89 ± 1.19
1B6J	RAND	-327.38	2.05	-322.45 ± 5.03	4.69 ± 2.83	1K6P	-383.38	5.26	-380.48 ± 2.56	6.42 ± 1.60
	SIM	-326.64	2.13	-321.75 ± 2.82	5.58 ± 2.41		-383.63	4.32	-381.08 ± 1.58	5.48 ± 1.33
	MMA	-329.11	1.11	-322.75 ± 3.12	3.87 ± 2.71		-383.37	3.95	-380.91 ± 2.57	4.95 ± 1.52
1G2K	RAND	-456.64	1.06	-449.38 ± 6.43	4.40 ± 2.90	1KZK	-440.35	1.58	-435.47 ± 4.49	4.17 ± 2.10
	SIM	-456.13	1.30	-451.27 ± 6.67	3.32 ± 2.53		-441.05	1.54	-435.40 ± 9.85	3.25 ± 2.12
	MMA	-456.70	0.83	-452.69 ± 5.78	2.68 ± 2.06		-441.03	1.53	-435.30 ± 9.21	3.29 ± 1.72
1HEG	RAND	358.31	8.12	359.47 ± 1.25	8.21 ± 1.32	1VIK	175.39	3.59	182.95 ± 11.34	3.74 ± 1.70
	SIM	358.04	5.74	359.52 ± 1.36	7.96 ± 2.30		173.33	2.29	191.48 ± 21.64	5.29 ± 3.65
	MMA	358.05	8.75	359.20 ± 0.70	8.05 ± 1.24		173.94	2.15	191.45 ± 21.94	4.32 ± 2.75

The SIM method has a slight advantage over our proposal, considering just the best energies found, but looking for the average, we get the best values in half instances. Also, regarding RMSD, the probability approach gets the best solutions

in five test cases, and about the average, the best results were divided between the algorithms. We applied a nonparametric test for multiple comparisons procedure, the *Dunn test* [25], to analyze the statistical significance in the achieved results. The test indicated that there is no difference between the approaches compared. In the second stage, we have applied our `MMA` algorithm shifting the radius perturbation of the `LS` with the values: $0.10, 0.25$, and 0.50. The idea of this stage is to verify if different modifications in the genes might improve the potential of the applied local search method. Table 2 summarizes the achieved results over the same benchmark utilized in the first test. In the same way, we evaluate the energy and `RMSD` from the best solutions, as well as the averages and standard deviations. Results show that we have an almost equal distribution of the best values found between the three variations, comparing just the best energy reached or the average. Thus, we conclude that all variations in the radius search could improve the `LS` process, for example, a bigger value can be more adequated at the beginning of the search, while a smaller one would be better in the algorithm refinement stage.

Table 2. Achieved results from the `MMA` approach with three variations of the `LS` radius perturbation. Third and fourth columns contains the lowest energy (*kcal/mol*) and its `RMSD` values for each version. Cells shaded in gray highlight the best solution obtained, and cells shaded in blue, the best solutions average, for each instance.

ID	Radius Search	Best solution Energy	RMSD	11 runs average Energy	RMSD	ID	Best solution Energy	RMSD	11 runs average Energy	RMSD
1AJX	0.10	-251.18	1.39	-243.06 ± 3.89	10.00 ± 4.05	1HPX	-355.42	1.24	-348.88 ± 2.10	5.50 ± 1.69
	0.25	-250.54	3.09	-242.69 ± 4.71	9.68 ± 4.07		-356.61	3.43	-350.85 ± 3.30	4.80 ± 1.34
	0.50	-250.20	3.21	-241.34 ± 2.86	11.48 ± 2.68		-357.00	2.73	-350.15 ± 3.90	4.89 ± 1.19
1B6J	0.10	-328.58	2.10	-325.19 ± 2.47	3.50 ± 2.45	1K6P	-383.59	4.96	-380.35 ± 2.57	5.70 ± 1.66
	0.25	-328.13	2.21	-321.98 ± 3.60	5.84 ± 2.57		-383.52	0.92	-380.80 ± 2.43	6.04 ± 2.45
	0.50	-329.11	1.11	-322.75 ± 3.12	3.87 ± 2.71		-383.37	3.95	-380.91 ± 2.57	4.95 ± 1.52
1G2K	0.10	-456.69	0.78	-443.59 ± 9.81	5.04 ± 2.51	1KZK	-440.93	1.56	-434.06 ± 9.59	4.51 ± 1.92
	0.25	-456.58	1.28	-452.81 ± 5.60	2.88 ± 2.04		-441.12	0.84	-438.83 ± 2.28	3.15 ± 2.07
	0.50	-456.70	0.83	-452.69 ± 5.78	2.68 ± 2.06		-441.03	1.53	-435.30 ± 9.21	3.29 ± 1.72
1HEG	0.10	357.60	9.53	359.46 ± 1.62	8.96 ± 2.41	1VIK	173.89	2.28	185.06 ± 13.93	4.68 ± 3.62
	0.25	358.10	5.57	359.41 ± 1.03	8.02 ± 1.92		172.99	2.16	191.22 ± 18.04	4.95 ± 2.39
	0.50	358.05	8.75	359.20 ± 0.70	8.05 ± 1.24		173.94	2.15	191.45 ± 21.94	4.32 ± 2.75

Also, the application of test Dunn showed that only in three instances (1B6L, 1G2K and 1HEF) there is a statistical difference between the three variations of the MMA algorithm. With these preliminary results, we have decided to extend and apply our proposal probability function to the choice of which radius perturbation to use during the optimization process, combining it with the previous variation in the `LS` methods. In this third stage, the weights and probabilities of the radius pool are updated after a local search execution. Similarly, the application and evaluation of the `LS` algorithms are made with any radius perturbation applied. Thus, we compare the `MMA` approach against a `BRKGA` version without `LS`, and another two `MA` versions with `SHD` and `SA`. Additionally, we compare these results with `AUTODOCK VINA` [5], `DOCKTHOR` [26], and `jMETAL` [6], a multiobjective docking approach. Table 3 shows the achieved results. The comparison of energy values was made only between our implemented methods because of the energy function utilized, which differs from other methods. The `RMSD` comparison was made between all algorithms. The values show that the `MMA` is better

than `BRKGA`, when we compare the average of energy and `RMSD`. Comparing the `SHD` and `SA` approaches with the self-adaptive, it is possible to notice that values are very similar. Considering the average `RMSD` values, the `MMA` got better results only in the `1AAQ` complex in comparison with the state-of-art algorithms. The best method overall was the `DOCKTHOR` with better `RMSD` values in 81% of the instances used in this work.

In the same way, we applied the Dunn test in this stage. Table 4 shows the significant levels of energy and `RMSD`, adopting a significance of $\alpha < 0.05$, guaranteeing a confidence of 95% in the analysis. Cells above the main diagonal (highlighted) shows the p-values comparing the energy results, and the remaining cells show the `RMSD` comparisons. Comparing `MMA` with `BRKGA` we find significant difference in all instances regarding energy and 50% in `RMSD`. Also, looking for the differences between the self-adaptive approach and `SHD` and `SA`, there are significant energy values only for complexes `9HVP` and `1AAQ`, respectively. Thus, the `MMA` shows to be better than a `BRKGA` method, but is equivalent with `MA` versions.

5   Conclusion and Future Work

The statistical test performed shows that the multimeme approach improves the results for the `MD` problem instead use only an evolutionary algorithm. Although, the comparisons with memetic implementations showed to be equivalent techniques. This is explained because in the `MMA` version is utilized, in a distributed way, four `LS` algorithms instead only one in the full search process. In this case, we may be losing computational effort with methods that do not contribute much to find local optimums. Also, in the `MA` is adopted a fixed radius whereas in the multimeme is utilized three values to be self-adapted. This feature is interesting because the methods can explore different neighborhoods in different stages of the evolution, and then compensate for a possible loss of processing. So, this pool of local search methods combined with the pool of radius perturbation confirms to be an interesting initial variation to reach reasonable solutions to this problem, but still, need improvements.

This work brings contributions to the use of a self-adaptive memetic computational technique. Also, the proposal and application of a function to evaluate, in execution, the local search impacts, to best adapt the parameters and guide the search process. Further investigations might consider different global search algorithms, e.g., *Differential Evolution* and *Particle Swarm Optimization*, as well as methods to run as local searches, such as *Solis and Wets* and *Nelder-Mead*. Besides that, the use of another objective function to evaluate the ligation energy, such as AutoDock Vina, it is interesting to confirm the performance of the algorithm. The results of minimum energy would be different but the ability of MMA will be the same.

About the self-adaptive approach is possible to change the pool of `LSO`, using only the `SHD` and `SA` methods, for instance. The values for radius perturbation can also be modified, as well as different values applied to each gene, i.e., the translation operation may have a specific value different of the rotation and

Table 3. Comparison results of MMA with BRKGA, two MA methods, AUTODOCK VINA, DOCKTHOR, and jMETAL. The lowest energy (*kcal/mol*) and its RMSD *Å* values are shown (best values highlighted in gray), as well as their respective averages and standard deviation (cells shaded in blue for the best values) for 31 runs of each algorithm.

ID	Radius Search	Best solution Energy	RMSD	11 runs average Energy	RMSD	ID	Best solution Energy	RMSD	11 runs average Energy	RMSD
1AAQ	BRKGA	36.70	1.14	42.75 ± 5.89	3.12 ± 2.47	1HIV	-164.16	2.92	-150.52 ± 3.10	6.65 ± 1.11
	SHD	36.88	1.13	37.94 ± 0.47	1.09 ± 0.14		-164.94	3.41	-163.67 ± 0.82	2.28 ± 0.55
	SA	37.04	1.14	39.12 ± 2.68	1.59 ± 1.51		-165.15	3.39	-163.19 ± 2.67	2.65 ± 1.04
	MMA	36.72	1.24	38.76 ± 3.73	1.64 ± 1.70		-165.12	3.38	-163.35 ± 3.58	2.85 ± 1.39
	VINA	3.93	9.75	9.07 ± 5.05	9.06 ± 0.86		-0.29	7.49	12.01 ± 18.70	8.09 ± 0.92
	DOCK	3.52	0.94	5.12 ± 1.41	9.25 ± 4.92		55.13	0.29	55.32 ± 0.14	0.29 ± 0.04
	JMETAL	-16.00	2.90	-12.56 ± 3.48	2.11 ± 1.15		-32.00	7.93	-18.14 ± 3.68	2.58 ± 1.45
1AJX	BRKGA	-246.85	4.40	-239.68 ± 1.87	12.24 ± 1.50	1HPX	-348.40	5.28	-346.81 ± 1.59	5.87 ± 1.04
	SHD	-250.51	0.98	-243.68 ± 4.39	8.59 ± 4.75		-357.34	3.15	-352.56 ± 3.99	4.04 ± 1.36
	SA	-250.62	1.22	-242.21 ± 3.84	10.32 ± 4.22		-356.79	2.97	-351.63 ± 3.73	4.66 ± 1.49
	MMA	-250.89	3.08	-243.00 ± 4.56	9.42 ± 4.57		-357.49	3.52	-349.83 ± 3.40	5.08 ± 1.17
	VINA	-10.74	1.52	-9.82 ± 0.42	6.21 ± 2.82		-9.08	5.74	-6.52 ± 2.18	6.18 ± 1.03
	DOCK	48.68	0.85	49.33 ± 2.30	0.88 ± 0.09		88.64	7.81	96.05 ± 9.69	7.07 ± 2.11
	JMETAL	-18.00	4.40	-12.84 ± 4.35	3.88 ± 1.64		-18.00	4.86	-11.73 ± 4.77	3.84 ± 0.94
1B6J	BRKGA	-328.69	1.11	-316.83 ± 7.49	5.85 ± 2.33	1HVH	428.01	8.94	434.16 ± 2.60	10.33 ± 2.22
	SHD	-328.70	1.09	-322.42 ± 3.26	5.06 ± 2.76		427.29	8.38	429.98 ± 2.45	8.65 ± 1.62
	SA	-328.27	1.18	-320.81 ± 3.99	5.94 ± 2.22		427.00	8.42	430.81 ± 2.26	8.66 ± 1.80
	MMA	-329.00	1.13	-322.17 ± 3.89	5.63 ± 2.84		427.95	8.41	431.74 ± 2.50	9.06 ± 2.79
	VINA	-9.12	2.89	-2.39 ± 3.64	6.92 ± 3.08		-8.65	7.34	-7.04 ± 1.12	5.98 ± 1.71
	DOCK	38.78	0.61	38.96 ± 0.11	0.61 ± 0.06		127.18	8.34	130.22 ± 2.76	6.10 ± 1.58
	JMETAL	-18.00	3.47	-11.45 ± 6.79	2.49 ± 0.71		-18.00	2.44	-12.45 ± 4.42	2.89 ± 0.52
1B6L	BRKGA	-317.46	3.09	-312.62 ± 3.11	6.64 ± 1.94	1K6P	-382.38	4.42	-375.89 ± 4.85	6.88 ± 1.76
	SHD	-319.21	2.80	-316.64 ± 1.83	3.54 ± 1.99		-383.78	0.83	-368.36 ± 67.28	5.39 ± 1.96
	SA	-318.57	2.96	-315.72 ± 2.34	4.42 ± 2.41		-383.08	5.36	-380.13 ± 1.85	6.29 ± 1.56
	MMA	-319.56	2.48	-315.66 ± 2.25	5.10 ± 2.43		-384.23	4.07	-381.27 ± 1.60	5.73 ± 1.93
	VINA	-12.71	0.89	-12.02 ± 1.29	2.22 ± 3.07		-5.16	5.22	-0.53 ± 3.20	7.45 ± 2.09
	DOCK	30.51	0.46	30.72 ± 0.13	0.43 ± 0.02		143.30	1.73	150.40 ± 6.76	2.10 ± 1.88
	JMETAL	-16.00	2.17	-13.14 ± 3.68	2.17 ± 0.63		-20.00	2.38	-14.69 ± 4.77	3.20 ± 0.85
1BV9	BRKGA	-80.07	0.74	-16.33 ± 44.55	7.27 ± 2.86	1KZK	-440.32	1.91	-422.16 ± 17.60	5.68 ± 2.14
	SHD	-80.64	0.67	-57.17 ± 18.63	6.83 ± 4.84		-441.16	1.54	-438.45 ± 2.99	2.83 ± 1.97
	SA	-80.41	0.70	-53.44 ± 22.76	6.90 ± 4.59		-441.03	1.55	-435.52 ± 5.89	3.94 ± 2.35
	MMA	-80.81	0.71	-53.52 ± 16.20	7.48 ± 4.36		-441.19	1.54	-437.01 ± 6.58	3.47 ± 2.31
	VINA	14.56	5.78	20.65 ± 2.64	8.41 ± 1.49		-9.85	2.37	-8.05 ± 0.83	5.73 ± 2.57
	DOCK	55.69	0.89	56.59 ± 1.04	0.94 ± 0.05		27.61	0.79	27.79 ± 0.16	0.75 ± 0.13
	JMETAL	-20.00	2.20	-14.61 ± 5.64	2.45 ± 1.31		-24.00	9.02	-10.44 ± 12.79	7.95 ± 1.63
1G2K	BRKGA	-455.77	1.34	-436.89 ± 9.44	6.04 ± 2.05	1MUI	-36.39	1.83	-26.57 ± 6.68	5.85 ± 3.02
	SHD	-456.71	0.88	-452.56 ± 5.60	2.83 ± 2.28		-36.79	1.82	-34.99 ± 1.86	1.72 ± 0.68
	SA	-456.62	0.78	-450.82 ± 6.38	3.43 ± 2.36		-36.57	1.93	-33.67 ± 2.64	2.53 ± 2.21
	MMA	-456.78	1.10	-451.14 ± 7.35	3.21 ± 2.52		-36.85	1.95	-33.83 ± 3.38	3.35 ± 2.75
	VINA	-10.01	4.41	-9.34 ± 0.86	6.05 ± 1.77		-8.22	6.68	-6.29 ± 1.15	7.48 ± 1.09
	DOCK	15.46	0.36	16.10 ± 1.49	0.43 ± 0.17		17.06	0.34	17.42 ± 0.15	0.46 ± 0.20
	JMETAL	-20.00	2.60	-12.84 ± 5.31	3.01 ± 1.69		-20.00	4.65	-13.04 ± 4.16	2.57 ± 0.98
1HEF	BRKGA	175.78	13.48	176.69 ± 0.57	12.25 ± 0.56	1VIK	174.70	2.18	228.25 ± 50.90	6.85 ± 2.82
	SHD	173.69	11.43	175.26 ± 1.00	11.79 ± 0.67		173.46	2.20	186.06 ± 16.52	3.82 ± 2.81
	SA	173.69	11.36	175.46 ± 0.87	12.00 ± 0.80		173.35	2.16	187.99 ± 18.04	4.59 ± 2.93
	MMA	173.58	11.40	174.81 ± 1.19	11.66 ± 0.33		173.34	2.15	182.83 ± 13.13	4.20 ± 2.66
	VINA	-1.79	9.36	1.42 ± 2.26	8.77 ± 0.56		96.97	11.19	140.08 ± 29.70	9.18 ± 2.02
	DOCK	67.68	1.75	68.07 ± 0.36	1.83 ± 0.15		165.54	1.38	267.46 ± 123.25	2.97 ± 2.60
	JMETAL	-22.00	6.60	-10.74 ± 6.25	5.81 ± 1.67		-40.00	4.90	-19.12 ± 14.44	4.01 ± 0.98
1HEG	BRKGA	357.76	6.91	361.20 ± 1.45	8.67 ± 1.68	9HVP	359.99	1.09	372.04 ± 14.29	4.98 ± 3.39
	SHD	357.18	6.45	359.00 ± 0.59	7.53 ± 1.36		360.26	1.46	362.91 ± 4.20	3.11 ± 3.44
	SA	356.76	5.56	359.11 ± 1.32	7.77 ± 1.82		360.09	1.05	362.94 ± 4.75	2.50 ± 2.46
	MMA	356.91	9.52	358.99 ± 1.11	8.19 ± 1.49		359.97	1.11	363.17 ± 4.78	3.36 ± 3.40
	VINA	-5.85	5.50	-5.51 ± 0.26	6.00 ± 0.98		-3.74	10.09	0.50 ± 4.03	9.39 ± 0.95
	DOCK	58.37	2.75	60.50 ± 1.91	4.02 ± 1.49		25.96	1.19	26.04 ± 0.05	1.21 ± 0.05
	JMETAL	-14.50	3.16	-9.15 ± 3.90	5.24 ± 1.73		-22.00	4.17	-14.89 ± 4.13	2.72 ± 1.01

Table 4. Analysis of results with a significant level equal to $p < 0.05$, cells above the main diagonal show the p-values for energy, and the entries below the RMSD values.

ID	Method	BRKGA	SHD	SA	MMA	ID	BRKGA	SHD	SA	MMA	ID	BRKGA	SHD	SA	MMA	ID	BRKGA	SHD	SA	MMA
1AAQ	BRKGA	---	0.00	0.43	0.00	1BV9	---	0.00	0.00	0.00	1HIV	---	0.00	0.00	0.00	1KZK	---	0.00	0.00	0.00
	SHD	0.00	---	0.10	0.72		1.00	---	1.00	1.00		0.00	---	1.00	0.44		0.00	---	0.39	1.00
	SA	0.00	1.00	---	0.00		1.00	1.00	---	1.00		0.00	1.00	---	0.67		0.05	0.31	---	0.67
	MMA	0.02	1.00	1.00	---		1.00	1.00	1.00	---		0.00	0.64	1.00	---		0.00	1.00	1.00	---
1AJX	BRKGA	---	0.00	0.01	0.00	1G2K	---	0.00	0.00	0.00	1HPX	---	0.00	0.00	0.00	1MUI	---	0.00	0.00	0.00
	SHD	0.00	---	1.00	1.00		0.00	---	0.86	1.00		0.00	---	1.00	1.00		0.00	---	0.45	1.00
	SA	0.34	0.88	---	0.64		0.00	1.00	---	0.32		0.03	0.73	---	1.00		0.00	1.00	---	1.00
	MMA	0.00	1.00	0.70	---		0.00	1.00	1.00	---		0.27	0.10	1.00	---		0.02	0.18	1.00	---
1B6J	BRKGA	---	0.00	0.20	0.00	1HEF	---	0.00	0.00	0.00	1HVH	---	0.00	0.00	0.01	1VIK	---	0.00	0.00	0.00
	SHD	1.00	---	1.00	1.00		0.01	---	1.00	1.00		0.03	---	1.00	0.10		0.00	---	1.00	0.68
	SA	1.00	1.00	---	0.71		0.19	1.00	---	0.44		0.01	1.00	---	1.00		0.08	0.78	---	0.18
	MMA	1.00	1.00	1.00	---		0.00	1.00	0.44	---		0.11	1.00	1.00	---		0.01	1.00	1.00	---
1B6L	BRKGA	---	0.00	0.00	0.00	1HEG	---	0.00	0.00	0.00	1K6P	---	0.00	0.00	0.00	9HVP	---	0.34	0.08	0.00
	SHD	0.00	---	1.00	0.83		0.12	---	1.00	1.00		0.09	---	1.00	1.00		0.54	---	1.00	0.02
	SA	0.00	1.00	---	1.00		0.28	1.00	---	1.00		1.00	1.00	---	0.21		0.18	1.00	---	0.12
	MMA	0.05	0.13	1.00	---		1.00	0.83	1.00	---		0.06	1.00	1.00	---		0.39	1.00	1.00	---

dihedral angles. Also, in the our probability function there are issues to be explored since the weights utilized in both terms until other details. The option by considering the average fitness of caste 'A' can be replaced by the fitness value of the best solution only, or the average of all solutions, or the average of the best individuals from each subcube. Another possibility is to implement a time window to evaluate the LS contributions, considering the last one hundred thousand evaluations, for example. Finally, future works can explore many features to turn this approach better to find solutions to molecular docking problem.

Acknowledgements

References

1. Jin, X., Zhihua, C., Wenyin, G.: An adaptive strategy to adjust the components of memetic algorithms. In: 2014 IEEE 26th International Conference on Tools with Artificial Intelligence. pp. 55–62 (Nov 2014)
2. Chen, Y.C.: Beware of docking! Trends Pharmacol. Sci. 36, 78–95 (2015)
3. Combs, S.A., Deluca, S.L., Deluca, S.H., Lemmon, G.H., Nannemann, D.P., Nguyen, E.D., Willis, J.R., Sheehan, J.H., Meiler, J.: Small-molecule ligand docking into comparative models with rosetta. Nature Protocols 8, 1277–1298 (2013)
4. Davis, I.W., Baker, D.: Rosettaligand docking with full ligand and receptor flexibility. Journal of Molecular Biology 385, 381–392 (2009)
5. Trott, O., Olson, A.J.: Autodock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J. Comput. Chem. 31(2), 455–461 (2010)

6. López-Camacho, E., Godoy, M.J.G., Nebro, A.J., Aldana-Montes, J.F.: jmetal-cpp: optimizing molecular docking problems with a c++ metaheuristic framework. Bioinformatics (2013)
7. Dar, A.M., Mir, S.: Molecular Docking: Approaches, Types, Applications and Basic Challenges. Journal of Analytical & Bioanalytical Techniques 08, 8–10 (2017)
8. Kitchen, D.B., Decornez, H., Furr, J.R., Bajorath, J.: Docking and scoring in virtual screening for drug discovery: Methods and applications. Nature Reviews Drug Discovery 3(11), 935–949 (2004)
9. Rosin, C.D., Halliday, R.S., Hart, W.E., Belew, R.K.: A comparison of global and local search methods in drug docking. In: In Proceedings of the Seventh International Conference on Genetic Algorithms. pp. 221–228. Morgan Kaufmann (1997)
10. Ruiz-Tagle, B., Villalobos-Cid, M., Dorn, M., Inostroza-Ponta, M.: Evaluating the use of local search strategies for a memetic algorithm for the protein-ligand docking problem. In: 2017 36th International Conference of the Chilean Computer Science Society (SCCC). pp. 1–12 (Oct 2017)
11. Leonhart, P.F., Spieler, E., Braun, R., Dorn, M.: A biased random key genetic algorithm for the proteinligand docking problem. Soft Comput. pp. 1–22 (2018)
12. Krasnogor, N.: Studies on the Theory and Design Space of Memetic Algorithms. Ph.D. thesis, University of the West of England (2002)
13. Jakob, W.: A general cost-benefit-based adaptation framework for multimeme algorithms. Memetic Computing 2(3), 201–218 (Sep 2010)
14. Domínguez-Isidro, S., Mezura-Montes, E.: A cost-benefit local search coordination in multimeme differential evolution for constrained numerical optimization problems. Swarm Evol. Comput. 39, 249 – 266 (2018)
15. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts - towards memetic algorithms (1989)
16. Krasnogor, N., Smith, J.: A tutorial for competent memetic algorithms: model, taxonomy, and design issues. EEE Trans. Evol. Comput. 9(5), 474–488 (Oct 2005)
17. Gonçalves, J.F., Resende, M.G.C.: Biased random-key genetic algorithms for combinatorial optimization. Journal of Heuristics 17(5), 487–525 (Oct 2011)
18. Aarts, E., Lenstra, J.K. (eds.): Local Search in Combinatorial Optimization. John Wiley & Sons, Inc., New York, NY, USA, 1st edn. (1997)
19. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220 4598, 671–80 (1983)
20. Molina, D., Lozano, M., Sánchez, A.M., Herrera, F.: Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains. Soft Comput. 15(11), 2201–2220 (2011)
21. Krasnogor, N., Smith, J.: Emergence of profitable search strategies based on a simple inheritance mechanism (2001)
22. Jakob, W.: Towards an adaptive multimeme algorithm for parameter optimisation suiting the engineers' needs. In: Parallel Problem Solving from Nature - PPSN IX. pp. 132–141. Springer Berlin Heidelberg (2006)
23. Morris, G.M., Huey, R., Lindstrom, W., Sanner, M.F., Belew, R.K., Goodsell, D.S., Olson, A.J.: Autodock4 and autodocktools4: Automated docking with selective receptor flexibility. J. Comput. Chem. 30(16), 2785–2791 (2009)
24. O'Boyle, N.M., Banck, M., James, C.A., Morley, C., Vandermeersch, T., Hutchison, G.R.: Open babel: An open chemical toolbox. J. Cheminf. 3(1), 1–14 (2011)
25. Dunn, O.J.: Multiple comparisons using rank sums. Technom. 6(3), 241–252 (1964)
26. de Magalhães, C.S., Almeida, D.M., Barbosa, H.J.C., Dardenne, L.E.: A dynamic niching genetic algorithm strategy for docking highly flexible ligands. Inf. Sci. 289, 206–224 (2014)