Variable Genetic Operator Search for the Molecular Docking Problem

Salma Mesmoudi[1], Jorge Tavares[2], Laetitia Jourdan[3], and El-Ghazali Talbi[3]

[1] Laboratoire d'Informatique de Paris 6
104 avenue du Président Kennedy
75016 Paris, France
[2] CISUC, Informatics Engineering Department, University of Coimbra
Polo II - Pinhal de Marrocos
3030-252 Coimbra, Portugal
[3] INRIA Lille - Nord Europe Research Centre
40, Avenue Halley Bt.A, Park Plaza
59650 Villeneuve d'Ascq, France
salma.mesmoudi@lip6.fr, jorge.tavares@ieee.org, {laetitia.jourdan, el-ghazali.talbi}@inria.fr

Abstract. The aim of this work is to present a new hybrid algorithm for the Molecular Docking problem: Variable Genetic Operator Search (VGOS). The proposed method combines an Evolutionary Algorithm with Variable Neighborhood Search. Experimental results show that the algorithm is able to achieve good results, in terms of energy optimization and RMSD values for several molecules when compared with previous approaches. In addition, when hybridized with the L-BFGS local search method it attains very competitive results.

1 Introduction

The protein-ligand complex is the base of drug design. The *in vivo* and *in vitro* development of a new drug is a long and costly process. Thus, docking algorithms have been developed to provide an *in silico* approach to the problem. The aim of protein-ligand docking research is to predict the conformation of a ligand relative to a target protein. To correctly predict a good complex, a docking algorithm must be able to generate several complexes, and recognize which one is the best among these. Therefore, the molecular docking problem can be considered as the optimization of structural and energetic criteria described by an objective function. This function is based on the degrees of flexibility which represent the position and the conformation of the ligand and the receptor (protein).

In the last few years, several docking approaches have been developped [1, 2]. More recently, Evolutionary Algorithms (EA) have become a popular choice in molecular docking applications since they usually perform better than other algorithms [3]. The goal of this work is to improve the quality of molecular docking solutions. To accomplish this objective, we propose the hybridization of Variable Neighborhood Search (VNS) [4] with an EA. VNS is a relatively

recent algorithm which operates by changing the neighborhood of a current solution to explore other regions of the search space. Variable Genetic Operator Search (VGOS) operates in the same manner as the standard VNS with a main difference: instead of using a single solution with works with a population of solutions and each neighborhood is defined by a pair of genetic operators. Later, we will also add a local search method to our approach, to better understand and compare its performance with previous approaches. We apply to and test with this new algorithm to the molecular docking problem.

The rest of the paper is structured as follows. Section 2 presents the VGOS algorithm and related issues. In Section 3 we describe the results obtained with VGOS and the respective analysis. Finally, the main conclusions are discussed in Section 4.

2 Variable Genetic Operator Search

Evolutionary algorithms for the molecular docking problem can be found in the literature since 1993 [5]. A review of these efforts can be found in [6, 7].

The original idea of the VGOS is to change genetic operators every time the algorithm starts to converge towards a local minimum. This variation of operators is inspired by the concept used in Variable Neighborhood Search. VNS is a stochastic algorithm in which an ordered set of neighborhood structures $N_k(k = 1, \ldots, n)$ are defined [4]. When a local optimum is obtained, in the neighborhood $N_i(1 \leq i \leq n)$, that is better than the current solution, N_1 becomes the current neighborhood and the local optimum becomes the current solution. Otherwise, if the local optima is worst than the current solution the neighborhood N_{i+1} becomes the current one. These steps are repeated until all neighborhoods are explored without the improvement of the current solution.

2.1 Neighborhood structures

The important part of the work is supported by the first neighborhood structure (N_1). Therefore, we must be aware that the neighborhood structures order is important and, specifically, the first structure has to be chosen carefully. The other neighborhood structures play the role of checking the first neighborhood structure to escape from local minimum. The way to construct our neighborhoods is very important. In our case, we will base our VGOS on a population of solutions instead of a unique solution and every neighborhood is defined by different crossover and mutation operators. Thus, we make a hybridization between VNS and an EA. Generally, as we increase the number of visited neighborhoods, the chance to escape local optima increases. However, to facilitate the order choice of the neighborhoods, we only use standard genetic operators.

In our work, we adopt an experimental model which uses the main components from [8, 9]. The main reason is that *AutoDock* (the EA proposed in [9]) serves as a basis for the large majority of evolutionary-inspired approaches, and thus, the attained results here can be useful on a larger degree.

2.2 Encoding and Evaluation

During the docking process the protein remains rigid whilst the ligand is flexible. An individual represents only the ligand, as the encoding is an indirect representation. A genotype of a candidate solution is encoded by a vector of real-valued numbers which represent the ligand's translation, orientation and torsion angles. Cartesian coordinates represent the translation, three variables in the vector, whereas four variables defining a quaternion represent the orientation. A quaternion can be considered to be a vector (x, y, z) which specifies an axis of rotation with an angle θ of rotation for this axis. For each flexible torsion angle one variable is used. The phenotype of a candidate solution is composed of the atomic coordinates that represent the three-dimensional structure of the ligand. The atomic structure is built from the translation and orientation coordinates in the ligand crystal structure with the application of the torsion angles.

We use the same evaluation function as established in [9]. To identify appropriate binding conformations, it uses an aproximate physical model to compute the energy of a candidate conformation. It uses empirical free energy potentials composed of four commonly used energy terms. The first three terms are pairwise interatomic potentials that account for weak long-range attractive forces and short-range electrostatic repulsive forces. The overall docking energy of a given ligand molecule is expressed as the sum of intramolecular interactions between the complex and the internal energy of the ligand:

$$E_{total} = E_{vdw} + E_{H-bond} + E_{elec} + E_{internal} \tag{1}$$

The first three terms are the intermolecular energies: van der Waals force (E_{vdw}), hydrogen bonding (E_{H-bond}), and electronic potential (E_{elec}). The last term is the internal energy of the ligand. Lower energy means better docking stability. Thus, the aim of any docking method is to minimize the total energy (E_{total}) value.

2.3 Genetic operators

In this work, genetic operators are important since they define the neighborhoods of the algorithm. In[10, 11], several types of mutation and crossover operators are tested when applying them to the molecular docking problem according to their influence in representation properties, such as locality, heritability and heuristic bias. We use operators based on evolutionary strategies. They act in the following way: when undergoing mutation, the new value for a gene x' is obtained from the old value x by adding a random real number sampled from a distribution $U(0, 1)$:

$$x' = x + \sigma \times U(0, 1) \tag{2}$$

The common distribution used for $U(0, 1)$ is the standard Uniform distribution. However, we also replace this distribution with the standard Gaussian

distribution, $N(0,1)$ and, in the same way as the *AutoDock* approach, with a Cauchy distribution. The Cauchy distribution is:

$$C(x,\alpha,\beta) = \frac{\beta}{\pi\beta^2 + (x-\alpha)^2} \tag{3}$$

where $\alpha \leq 0, \beta > 0, -\infty < x < +\infty$ (α and β are parameters that control the mean and spread of the distribution). The two main mutation operators used are Gaussian and Cauchy mutation like in previous works (see e.g., [10, 12]). The third operator used is Uniform mutation.

In terms of crossover, we also apply three operators. Several common crossover operators, such as Whole Arithmetical crossover, Discrete crossover, and Blend-α crossover are used. For a complete description of their operation, we refer the reader to the following literature [13, 14].

2.4 Algorithm

The order notion is very important in the VGOS algorithm. Because of its importance, the exploration of the first neighborhood was performed by the best operators, thus the choice of the mutation and crossover is a crucial step [11, 10]. We use three kinds of mutation and crossover and each EA_{N_k} is based on a specified pair of operators. Each pair is applied on a neighborhood. Thus, the EA_{N_1} is based on Discret crossover and Gaussian mutation, the second EA_{N_2} is constructed from Whole Arithmetic crossover and Cauchy mutation, and finally, the third pair used in EA_{N_3} is Blend-α crossover and Uniform mutation.

Algorithm 1 Variable Genetic Operator Search

Require: a set of different neighborhood structures N_k for $k = 1, \ldots, max$

```
pop ⇐ initial population of random solutions
x_best ⇐ best_element(pop)
k ⇐ 1
repeat
    pop ⇐ EA_{N_k}(pop)
    x'_best ⇐ best_element(pop)
    if f(x'_best) > f(x_best) then
        k ⇐ 1
    else
        k ⇐ k + 1
    end if
until k = max + 1
return best element found
```

As described in algorithm 1, from the current population (pop), we identify the best individual (x_{best}) and the current evolutionary algorithm EA_{N_k} is applied to (pop), to obtain a new population. We put in $x\prime_{best}$ the best element

from this new population. If and only if a better solution 'better fitness (f)" has been found (i.e., f($x\prime_{best}$) > $f(x_{best})$), the search procedure EA_{N_1} is restarted from (pop). If no better solution is found (i.e., f($x\prime_{best}$) $\leq$ $f(x_{best})$) the algorithm moves to the next neighbood (next mutation/crossover) ($EA_{N_{k+1}}$). These operations will be repeated until all neighborhoods (operators) are visited and no improvement is found or the maximum number of generations is reached.

3 Experimentation and Analysis

To perform our experimentation we used instances from the *AutoDock* test suite. The suite is composed of eight protein-ligand complexes. Each complex contains a macromolecule (the protein) and a small substrate or inhibitor molecule (the ligand). The structures of these molecular complexes have been obtained from the Protein Data Bank (PDB). The essential information about the complexes in the test suite are included in Table 1.

Table 1. X-ray crystal structures used in the docking experiments.

Protein-ligand complex	PDB	Resolution	Torsion	Size
Alcohol dehydrogenase	1adb	2.4	14	21
Alpha-Thrombin	1bmm	2.6	12	19
Beta-Trypsin	3ptb	1.7	0	7
Carbonic anydrase	1nnb	2.3	9	16
Trypsin	1tnh	1.80	2	9
IGG1-KAPPA DB_3 FAB	2dbl	2.9	6	13
L-Arabinose-binding protein	7abp	1.67	4	11
HIV_{-1} Protease	1hvr	1.80	10	17

We evaluate a resulting ligand conformation by comparing it with the experimental structures using the standard Cartesian root-mean-square deviation (RMSD):

$$RMSD_{lig} = \sqrt{\frac{\sum_{i=1}^{n} dx_i^2 + dy_i^2 + dz_i^2}{n}} \tag{4}$$

where n is the number of atoms in the comparison and dx_i^2, dy_i^2 and dz_i^2 are the deviations between the crystallographic structure and the corresponding coordinates from the predicted structure lig on Cartesian coordinate i. RMSD values below or near 2.0Å can be considered to be a success and the ligand is classified as being docked. On the other hand, a structure with an RMSD just less than 3.0Å is classified as partially docked. Thus, lower values mean that the observed and the predicted structures are similar.

For the first experiments, we consider two versions of a standard evolutionary algorithm. The only difference between them is the local search method

whereas the first variant does not include any type of local search, mimicking the algorithm presented in [15]. The second evolutionary algorithm includes the Solis-Wets algorithm as local search, thus being equivalent to the algorithm contained in the *AutoDock* package [16]. As for the rest of the evolutionary algorithm components, the basic algorithm for our experiments follows what was described in the previous sections in terms of representation, fitness function and genetic operators. The representation and evaluation is the one used by *AutoDock*. The genetic operators used are the Whole Arithmetic crossover and the Gaussian-based mutation operator. Moreover, the evolutionary algorithm is a standard generational algorithm with stochastic tournament selection and weak elitism. Our proposed VGOS algorithm will be tested against these two variants. To elaborate the different neighborhoods, we use three mutation and crossover operators as explained in the previous section.

3.1 Settings

The parameter values were set heuristically, even though we did some additional tests and verified that, within a moderate range, there was no significant difference in the outcomes. In any case, we did not perform a comprehensive study on the influence of different parameter settings and it is possible that a careful fine-tuning of some values could bring slight improvements to the achieved results. For all experiments, the settings of the tested algorithms are the following: Population size: 150, Selection rate: 0.8; Crossover rate: 0.9; Mutation rate: 0.5; Tournament selection rate: 0.95; Tournament replacement rate: 0.9; Number of local search steps: 1000; Maximum number of fitness evalutions: 10 000 000. We should note that, although the maximum number of fitness evaluations is high, in the case that no improvement is found after 10 000 fitness evaluations, the algorithm stops. A local search step counts as a single fitness evaluation. For all experiments the number of runs is equal to 30.

3.2 Experiments

Table 2 shows us the comparison between VGOS and the standard evolutionary approaches, without local search and with the Solis-Wets method. For all molecules the table contains columns with the best energy (Best), the average (Avg) and standard deviation (Std) for the 30 runs. Bold values in the Best and Avg columns indicate the best value for that instance. The most important observation is that VGOS attains the best results for every single instance, in terms of average and for seven instances for the best energy value found. The exception was the complex 1nnb where the best energy was attained by the EA with the Solis-Wets method. These results are encouraging since it shows that VGOS is competitive against the EA that models the evolutionary algorithm used in the *AutoDock* package. The efficiency of VGOS can also be considered superior to the other approaches. For the instances 1adb and 1bmm, the larger ones, VGOS is the only approach to be able to consistently find good solutions since the average is negative in value, i.e., all the solutions found are closer to

Table 2. Summary of optimization results for VGOS according to energy evaluation.

Complex		EA + No LS			EA + Solis-Wets			VGOS		
Label	Size	Best	Avg	Std	Best	Avg	Std	Best	Avg	Std
1adb	21	-6.59	335.34	953.64	-3.85	192.36	559.51	**-17.33**	**-15.76**	1.32
1bmm	19	-3.94	5.00	11.68	-4.66	5.04	18.84	**-10.32**	**-4.98**	3.97
1hvr	17	-15.79	-8.67	9.39	-13.81	-10.13	4.07	**-16.32**	**-15.86**	0.27
1nnb	16	-4.53	-2.48	1.69	**-5.18**	-2.16	1.57	-5.17	**-3.74**	1.22
1tnh	9	-5.39	-2.73	1.36	-5.37	-2.49	1.14	**-5.84**	**-4.41**	1.38
2dbl	13	-10.03	-4.66	3.47	-11.43	-4.26	3.06	**-11.74**	**-9.67**	2.84
3ptb	7	-5.49	-3.66	1.02	-5.99	-4.16	1.18	**-6.23**	**-6.05**	0.23
7abp	11	-7.97	-6.78	0.90	-7.90	-6.84	1.14	**-8.87**	**-7.42**	2.64

Table 3. Summary of optimization results for VGOS according to RMSD values.

Complex		EA + No LS			EA + Solis-Wets			VGOS		
Label	Size	Best	Best-En	Avg	Best	Best-En	Avg	Best	Best-En	Avg
1adb	21	1.47	1.53	2.78	1.34	1.50	2.79	**0.06**	**0.33**	**0.41**
1bmm	19	2.06	2.06	4.18	1.29	1.29	3.89	**0.09**	**0.25**	**3.74**
1hvr	17	0.29	**0.29**	0.96	0.53	0.65	0.90	**0.24**	0.54	**0.42**
1nnb	16	0.44	0.76	1.84	0.62	**0.64**	2.76	**0.29**	0.65	**1.74**
1tnh	9	0.70	0.73	2.94	0.34	0.34	3.38	**0.07**	**0.16**	**1.32**
2dbl	13	0.57	0.78	2.42	0.53	0.53	2.64	**0.09**	**0.18**	**0.93**
3ptb	7	0.42	0.54	2.07	0.31	0.31	1.45	**0.04**	**0.14**	**0.23**
7abp	11	0.22	0.61	0.65	0.26	**0.45**	0.86	**0.21**	0.79	**0.56**

the best energy value. This is confirmed by the low value of the standard deviation. For the remaining instances, where all the approaches are able to find good solutions, VGOS has an average efficiency of 20% while the EA+Solis-Wets is around 41% and the simple EA is 67%. These values are obtained by calculating the distance of the runs average to the best solution found by the approach.

The same trend is also found when we consider RMSD values. Looking at table 3 we confirm the good performance of VGOS. The table contains for the three approaches and the eight problem instances: the best RMSD value found (Best), the RMSD value associated to the best energy value found contained in the previous table (Best-En) and the average of the RMSD values. The VGOS approach has the best values for the Best and Avg columns for every instance. Only in three occasions VGOS did not attain the best results for Best-En. The exceptions were 1hvr, 1nnb and 7abp. However, the important aspect to consider in this table is, overall, the RMSD values presented are very close zero. This indicates that solutions with a good structure are found. As expected, the instances where this is not verified are the ones where VGOS did not attain the best relationship between energy and RMSD (1hvr, 1nnb and 7abp).

3.3 Experiments with Local Search

The Solis-Wets algorithm, used in *Autodock* [16], is a direct search method with an adaptive step size, which performs a randomized local minimization of a given candidate solution. In [12], it is shown that this method is inefficient for the Molecular Docking problem in the context of evolutionary search. The recommended Local Search algorithm is the Broyden-Fletcher-Goldfarb-Shanno method (L-BFGS) [17].

L-BFGS is a powerful quasi-Newton conjugate gradient method, where both the function to minimize and its gradient must be supplied, but no a priori knowledge about the corresponding Hessian matrix is required. During local search, the maximum number of iterations that can be performed is specified by a parameter of the algorithm. However, the method stops as soon as it finds a local optimum, so the real number of iterations may be smaller than the specified value.

We will now hybridized VGOS with Solis-Wets and L-BFGS to see how the algorithm performs with an additional component. Furthermore, we will compare with the very efficient EA algorithm described in [12].

Table 4. Summary of optimization results for VGOS with LS according to energy evaluation.

Complex		EA + L-BFGS			VGOS + L-BFGS			VGOS + Solis-Wets		
Label	Size	Best	Avg	Std	Best	Avg	Std	Best	Avg	Std
1adb	21	-18.30	-14.00	4.80	**-22.80**	**-21.35**	1.50	-17.87	-16.28	3.43
1bmm	19	**-11.99**	-5.52	2.73	-11.91	**-5.64**	4.07	-10.39	-3.95	3.79
1hvr	17	**-32.43**	**-32.13**	0.28	-29.30	-28.80	0.35	-16.37	-16.01	0.52
1nnb	16	**-6.25**	-4.63	1.50	-6.09	**-5.01**	0.90	-5.23	-4.05	-10.39
1tnh	9	**-6.06**	**-5.54**	1.09	-5.83	-4.84	1.35	-5.84	-4.25	1.78
2dbl	13	-12.14	-10.04	2.99	**-12.19**	-9.71	3.70	-11.82	**-10.23**	2.74
3ptb	7	**-6.33**	**-6.22**	0.25	-6.22	-6.09	0.14	-6.23	-5.99	0.48
7abp	11	**-9.12**	**-8.67**	0.37	-8.69	-7.81	0.41	-8.81	-7.84	0.31

Table 4 compares the two variants of VGOS, with Solis-Wets and L-BFGS methods, against the EA with L-BFGS. The columns follow the same structure as table 2. Some important information can be drawn from the presented data. Regarding the best energy values the VGOS approaches are not competitive with the EA. With the exception of the problem instance 1adb, the EA attains all the best results. Although for most of the instances, the VGOS+L-BFGS approach stands close, it is unable to attain a better performance. VGOS with the Solis-Wets performs worse.

However, if we compare the results between VGOS and VGOS hybridized with the L-BFGS method, it is clear that the later performs better. This effect is not observable with the other VGOS hybrid. When looking at the Avg we find

Table 5. Summary of optimization results for VGOS with LS according to RMSD values.

Complex		EA + L-BFGS			VGOS + L-BFGS			VGOS + Solis-Wets		
Label	Size	Best	Best-En	Avg	Best	Best-En	Avg	Best	Best-En	Avg
1adb	21	0.30	0.51	1.64	**0.05**	**0.29**	**0.42**	0.07	0.35	0.46
1bmm	19	0.67	1.10	**3.63**	**0.06**	0.69	3.98	0.08	**0.25**	4.30
1hvr	17	0.50	0.71	0.61	0.25	**0.49**	0.46	**0.21**	0.64	**0.44**
1nnb	16	0.39	0.74	1.84	**0.20**	**0.10**	**1.04**	0.29	0.81	1.74
1tnh	9	0.20	0.94	1.16	0.10	0.67	**1.12**	**0.03**	**0.17**	2.21
2dbl	13	0.35	0.35	1.40	**0.06**	**0.22**	1.06	0.15	**0.22**	**0.75**
3ptb	7	0.24	0.51	0.93	0.02	0.24	**0.23**	**0.01**	**0.14**	0.36
7abp	11	0.35	0.82	0.76	**0.18**	**0.90**	**0.51**	0.25	**0.90**	**0.51**

a more balanced situation. The EA attains four of the best averages, VGOS+L-BFGS attains three and VGOS+Solis-Wets one. This primarily reflects the effect of the local search method used. The approaches using L-BFGS show a better performance than the one with the Solis-Wets method. This extendes and supports the findings of [12].

In the same way as before, table 5 displays the overview of the RMSD values, where each line displays the results for a complex. The differences between the three approaches are now between the EA and the VGOS hybrids. Both VGOS approaches attain among themselves the best results., in terms of Best, Best-En and Avg. The EA with the L-BFGS method was unable to obtain a single best result. The only exception was the average for the 1bmm complex. This indicates how well the VGOS algorithm is able to perfom in terms of RMSD, confirming the previous results (see table 3). However, there are no major differences between the two VGOS variants. The influence of a local search method is less visible. It was already shown that the L-BFGS method was more efficient with regards to energy optimization [12]. Thus, it was expected that VGOS with L-BFGS would not attain a significant difference from the other variant.

This table reinforces the importance of the relation between the RMSD value attained with the best energy. The lowest energy found is only good if it corresponds to a low RMSD value. From the Best-En column it is possible to conclude that the distribution of the best values is focused between the two approaches with VGOS. The conclusion we can draw from this is that, although the EA+L-BFGS is more capable to optimize in terms of energy minimization, the other methods (although with higher energies) can reach solutions with a more similar structure to the optimal case.

This is shown by the scatter plots in figures 1 and 2. The plots display for the 1adb and 1bmm complexes the position of the discovered solutions, the relation between energy and RMSD for an optimization run. The closer to the origin of the axis the better. Common patterns can be detected on both plots: there is a clear approximation flow to the origin, whereas the VGOS approaches clearly approximate more, with a higher concentration. The main differences between

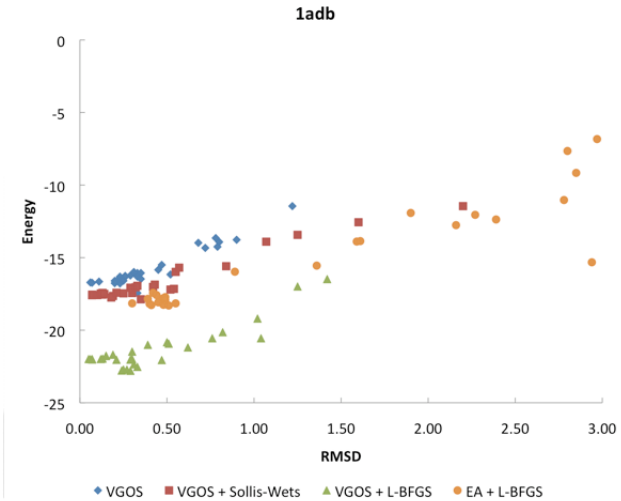

Fig. 1. Scatter plot of an optimization run for the 1adb complex.

the approaches are: 1) VGOS+L-BFGS is the only one that approximates the origin in terms of energy and RMSD (especially for 1adb complex); 2) the majority of the initial solutions for the EA are more distant. The main information from the these plots is that VGOS+L-BFGS drives more solutions with lower energy and RMSD values closer to the desired point.

4 Conclusions and Future Work

We proposed and investigated a new optimization method for the Molecular Docking problem which hybridizes Variable Neighborhood Search with an Evolutionary Algorithm. Variable Genetic Operator Search operates in the same manner as standard VNS , i.e., by changing the neighborhood of a current solution to explore other regions of the search space. The main difference is that instead of working with a single solution, it operates with a population of solutions. Later, we also add local search methods to our approach.

Results show that VGOS is able to attain consistently good results in terms of energy and RMSD values. The method is superior to previous approaches when considering the dual relationship between energy and structural optimization. Moreover, this is achieved by adding the L-BFGS method as local search. In spite of an EA with this local search method provides better results in terms of energy optimization, the VGOS approach is able to discover solutions with better RMSD values. For future work, we intend to study and understand the relation between the molecules topology and operators. With this knowledge, it is our aim to design more suitable operators adapted to each complex.

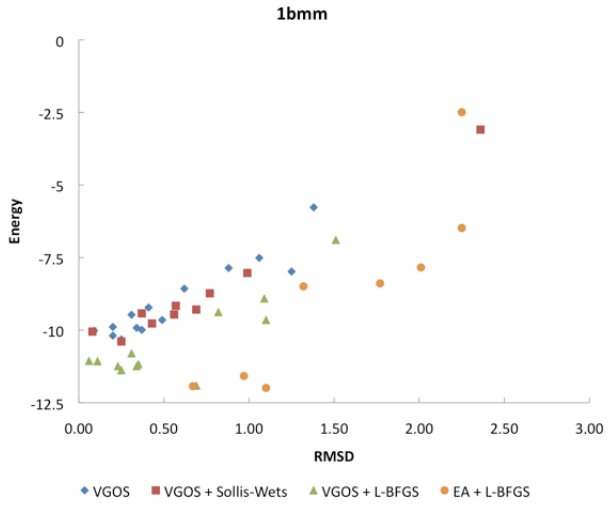

Fig. 2. Scatter plot of an optimization run for the 1bmm complex.

References

1. Diller, R.M., Verlind, L.M.J.: A critical evaluation of several global optimization algorithms for the purpose of molecular docking. J.Comput Chem **20** (1999) 1740–1751
2. McConkey, B., Sobelove, V., Edlman, M.: The performance of current methods in ligand-protein docking. Current Science **83** (2002) 845–855
3. Oduguawa, A., Tiwari, A., Fiorentino, S., Roy, R.: Multi-objective optimisation of the protein-ligand docking problem in drug discovery. In: ACM, GECCO'06,. (2006)
4. Maldenovic, N., Hansen, P.: Variable neighborhood search. Computers in Operations Reasearch **24** (1997) 1097–1100
5. Dixon, J.S.: Flexible docking of ligands to receptor sites using genetic algorithms. In: Proc. of the 9th European Symposium on Structure-Activity Relationships, Leiden, The Netherlands, ESCOM Science Publishers (1993) 412–413
6. Thomsen, R.: Protein-ligand docking with evolutionary algorithms. In Fogel, G.B., Corne, D.W., Pan, Y., eds.: Computational Intelligence in Bioinformatics. Wiley-IEEE Press (2008) 169–195
7. Moitessier, N., Englebienne, P., Lee, D., Lawandi, J., C.Corbeil: Towards the development of univeral, fast and highly accurate docking/scoring methods: a long way to go. British Journal of Pharmacology **153** (2007) 1–20
8. Thomsen, R.: Flexible ligand docking using evolutionary algorithms: investigating the effects of variation operators and local search hybrids. Biosystems **72** (2003) 57–73
9. Morris, G.M., Goodsell, D.S., Halliday, R.S., Huey, R., Hart, W.E., Belew, R.K., Olson, A.J.: Automated docking using a lamarkian genetic algorithm and an empirical binding free energy function. J. Comput. Chem. **19** (1998) 1639–1662

10. Tavares, J., Tantar, A.A., Melab, N., Talbi, E.G.: The influence of mutation on protein-ligand docking optimization: a locality analysis. In: Proceedings of the 10th International Conference on Parallel Problem Solving From Nature. (2008)
11. Tavares, J., Melab, N., Talbi, E.G.: An empirical study on the influence of genetic operators for molecular docking optimization. Technical Report RR-6660, INRIA Lille - Nord Europe research Centre (2008)
12. Tavares, J., Mesmoudi, S., Talbi, E.G.: On the efficiency of local search methods for the molecular docking problem. In: 7th European Conference on Evolutionary Computation, ML and Data Mining in BioInformatics. (2009)
13. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Spring (2003)
14. Lozano, M., Herrera, F., Krasnogor, N., Molina, D.: Real-coded memetic algorithms with crossover hill-climbing. Evolutionary Computation **12** (2004) 273–302
15. Thomsen, R.: Flexible ligand docking using evolutionary algorithms: investigating the effects of variation operators and local search hybrids. Biosystems **72** (2003) 57–73
16. Morris, G.M., Goodsell, D.S., Halliday, R.S., Huey, R., Hart, W.E., Belew, R.K., Olson, A.J.: Automated docking using a lamarckian genetic algorithm and and empirical binding free energy function. Journal of Computational Chemistry **19** (1998) 1639–1662
17. Liu, D.C., Nocedal, J.: On the limited memory method for large scale optimization. Mathematical Programming B (1989) 503–528