On the Efficiency of Local Search Methods for the Molecular Docking Problem

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

[1] Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139, USA
[2] INRIA Lille - Nord Europe Research Centre
40, Avenue Halley Bât.A, Park Plaza
59650 Villeneuve d'Ascq, France
jast@mit.edu
{salma.mesmoudi,el-ghazali.talbi}@inria.fr

Abstract. Evolutionary approaches to molecular docking typically hybridize with local search methods, more specifically, the Solis-Wet method. However, some studies indicated that local search methods might not be very helpful in the context of molecular docking. An evolutionary algorithm with proper genetic operators can perform equally well or even outperform hybrid evolutionary approaches. We show that this is dependent on the type of local search method. We also propose an evolutionary algorithm which uses the L-BFGS method as local search. Results demonstrate that this hybrid evolutionary outperforms previous approaches and is better suited to serve as a basis for evolutionary docking methods.

1 Introduction

In general terms, molecular docking is a search problem that aims to find the best conformation and orientation of a molecule relative to the active site of a second target molecule with the lowest energy [1]. The typical case is to have a protein as a receptor, fixed in a three-dimensional coordinate system, and a ligand, which can be repositioned and rotated to dock it with the receptor. The docking problem is very difficult since the relative orientation and conformations of the two molecules must be considered. With both molecules flexible, usually the active site of the protein and the ligand, the problem becomes harder. In fact, a higher degree of flexibility implies a considerable increase of the search space size.

Evolutionary algorithms have recently become one of the dominant search techniques for docking methods [2] and proved to be very successful. Despite the fact that numerous applications of evolutionary algorithms exist, the number of studies which focus on *understanding why* the algorithm's components are successful is scarce. To the best of our knowledge, the first attempt made concerning this important topic can be found in [3]. Several parameters (e.g., population size) and some genetic operators are empirically investigated, as well

as the efficiency of the use of a local search method. Some of our previous work reflects this concern of studying the effect of the components of an evolutionary algorithm on the search process [4, 5]. However, in [3] it is concluded that local search methods might not be very helpful in the context of molecular docking. The several experiments provided there show that an evolutionary algorithm without local search but with proper genetic operators can perform equally well or even outperform the best evolutionary algorithms with local search for this problem, e.g., [6, 2].

In this paper, we perform an empirical analysis on the evolutionary algorithm model usually adopted for molecular docking optimization [6, 2]. We show, contrary to [3], that local search methods can be useful for this problem, when the appropriate method is selected, regardless of the genetic operators used. Results confirm that the popular Solis-Wets algorithm is not an ideal local search method for molecular docking since the solution's quality and the algorithm's efficiency is similar to evolutionary algorithms without auxiliary methods. In contrast, an evolutionary algorithm with the L-BFGS method, a powerful quasi-Newton conjugate gradient method, improves the results attained by an evolutionary algorithm tailored to molecular docking considerably.

The rest of the paper is structured as follows. Section 2 contains an overview of the evolutionary algorithm's components used in our experimentation. In section 3 we present the experiments and their results. Finally, section 4 contains the main conclusions.

2 Evolutionary Algorithms and Molecular Docking

Since 1993, evolutionary algorithms for the molecular docking problem can be found in the literature [7]. A comprehensive review of these efforts can be found in [8, 2]. One of the most important works is the evolutionary algorithm proposed in [6], which is part of the package named *AutoDock*. The evolutionary algorithm is a conformational search method which uses an approximate physical model to evaluate possible protein-ligand conformations. It incorporates flexibility by allowing the ligand to change its conformation during the docking simulation. In addition, pairwise interactions between atoms are pre-calculated, which considerably speeds up the docking simulation. The approach uses an evolutionary algorithm with a local search method to search the space of possible protein-ligand conformations. When this method is applied, the genotype of the individuals is replaced with the new best solution found. This process is usually referred to as Lamarckian evolution. In our work, we adopt an experimental model which uses the main components from [6, 3]. The main reason is that *AutoDock* serves as a basis for the large majority of evolutionary-inspired approaches (e.g., see[2]) and thus, the attained results here can be useful on a larger degree.

2.1 Encoding

During the docking process the protein remains rigid whilst the ligand is flexible. An individual represents only the ligand. The encoding is an indirect represen-

tation. 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 [6]. Cartesian coordinates represent the translation which is defined by three variables in the vector. The orientation is determined by four variables in the vector, which define a quaternion. 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.

2.2 Evaluation

An energy evaluation function is used to evaluate each individual. The fitness for each candidate solution is given by the sum of the intermolecular interaction energy between the ligand and the protein, and the intramolecular energy that arises from the ligand itself [6]. *AutoDock* uses an empirical free energy potential composed of five terms. The first three terms are pairwise interatomic potentials that account for weak long-range attractive forces and short-range electrostatic repulsive forces. The fourth term measures the unfavorable entropy of a ligand binding due to the restriction of conformational degrees of freedom. The fifth and last term uses a desolvation measure. Further details of the energy terms and how the potential is derived can be found in [6].

2.3 Genetic Operators

Common crossover and mutation operators are applied on the population. In *AutoDock* a standard two-point crossover is used. Cut points only occur between related genes, i.e., separating translational values, orientation values and rotation torsion angles into separate blocks. This is done to avoid disruption of useful parts of the solution [6]. However, for real-valued encodings it is recommended that operators designed to deal with this type of encoding are used. In our previous work [5], we analyze several common crossover operators, such as Simple Arithmetical crossover, Whole Arithmetical crossover, Discrete crossover, Simulated Binary Crossover and Blend-α crossover, in terms of their effect on representation properties, i.e., locality, heritability and heuristic bias. Results showed that Whole Arithmetical crossover is a good operator for molecular docking, which confirms some experimentations discussed in [3].

Since the encoding is a real-valued vector, mutation is performed by using operators based on evolutionary strategies. The genetic operator acts 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 $U(0, 1)$:

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

where $N(0,1)$ is the standard Gaussian distribution. Here. an important aspect is determining the value for parameter σ. If it is set too low, exploitation overcomes exploration and if set too high *vice versa*. The value can be fixed or self-adapted (e.g., if an evolutionary strategy approach is used). In [3], the use of annealing schemes to control σ as a function of time, i.e., the number of generations, is proposed. The following scheme is used, scaled with 0.1:

$$\sigma(t) = \frac{1}{\sqrt{1+t}} \tag{2}$$

In the same manner as crossover, in [4, 5] we also show that this kind of mutation operator is best suited for molecular docking, especially when comparing it with operators based on Cauchy distributions.

2.4 Local Search

In this work, we study the impact of two local search methods. The first technique is the algorithm used in [6]: the Solis-Wets method. The Solis-Wets algorithm is a direct search method with an adaptive step size, which performs a randomized local minimization of a given candidate solution. The process starts with a candidate solution $x \in \Re^n$ and for each step a deviate $\epsilon \in \Re$ is chosen from a normal distribution. In the case a better solution is found by adding or removing ϵ from x, the current solution is replaced with the new one. Depending on whether a new solution if found or not, a success or a failure is recorded. If several successes occur in a row, the variance of the normal distribution is adapted for the search to move more quickly. If the opposite occurs, the variance is adapted to focus the search. This is accomplished through a parameter, ρ. Moreover, a bias term is applied to drive the search in successful directions. The method terminates when a certain lower-bound threshold for ρ is passed or when a maximum number of steps is reached.

The second method used is the Broyden-Fletcher-Goldfarb-Shanno method (L-BFGS) [9]. 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.

In a previous work concerning the analysis of local search methods for this problem [10], we analyzed the Solis-Wets algorithm and the simplex local search algorithm described by Nelder and Mead (NMS) for nonlinear, continuous function optimization [11]. The analysis was made in terms of locality and the attained results showed that both methods had a similar high impact on locality. As such, the performance of an evolutionary algorithm, as well as the solutions quality, is not considerably different when using one of these two methods. For this reason, the NMS method is not considered in this paper.

3 Experimentation and Analysis

Several instances from the *AutoDock* test suite are used to perform our optimization tests. 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). Table 1 provides information about the complexes names as well as their PDB identification.

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

To evaluate a resulting ligand conformation we compare it with the experimental structures using the standard Cartesian root-mean-square deviation (RMSD):

$$RMSD_{lig} = \sqrt{\frac{\sum_{i=1}^{n} dx_i^2 + dy_i^2 + dz_i^2}{n}} \tag{3}$$

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 a RMSD just less than 3.0Å is classified as partially docked. Thus, lower values mean that the observed and the predicted structures are similar.

We consider three versions of a standard evolutionary algorithm for our experimentation. The only difference between them is the local search method. The first variant does not include any type of local search, mimicking the algorithm presented in [3]. The second variant includes the Solis-Wets algorithm as local search, thus being equivalent to the algorithm contained in the *AutoDock* package [6]. The last evolutionary algorithm is our own proposal, which includes the L-BFGS method as local search. 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 described in the previous section. Moreover, the evolutionary algorithm is a standard generational algorithm with stochastic tournament selection and weak elitism.

3.1 Settings

The parameter values were set heuristically, even though we did some additional tests and verified that, within a moderate variation, 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: Number of runs: 30; 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 evaluations: 10 000 000. We should note that, although the maximum number of fitness evaluations is high, in the case where no improvement is found after 10 000 fitness evaluations, the algorithm stops. For both local search methods, a step counts as a single fitness evaluation.

3.2 Experiments

In table 2 we present an overview of the optimization results according to the energy fitness function. Each line displays the results for a complex, identified by the PDB label, with a given size. For each of three variants of the evolutionary algorithm, i.e., no local search (No LS), with the Solis-Wet algorithm (Solis-Wets) and the L-BFGS method (L-BFGS), a column is presented with three subcolumns. These contain the best energy value found during the 30 runs (Best), the average of the best energy values for 30 runs (Avg) and the corresponding standard deviation (Std). Bold values indicate the best values found.

A first observation of table 2 indicates a very important result: the evolutionary algorithm with the L-BFGS local search method achieves the best results for all instances. Moreover, it also attains the best results in terms of average and standard deviation. This clearly shows that the use of this local search method provides competitive results when compared to the alternative configurations. In fact, a closer look at the results reveals that the quality of the solutions found can be considerably. For the larger and more difficult complexes (1adb, 1bmm, and 1hvr) the order of magnitude can be the double, at least, when compared to the second best algorithm. For the large instance, 1adb, the L-BFGS attains an energy score of -18.30 whilst the variant without local search, the second best, only achieves -6.59. For the remaining complexes, especially the ones with a smaller size, the differences are not that striking. The complex 3ptb, which has the smaller size of the eight tested complexes, demonstrates this. The L-BFGS evolutionary algorithm has a best energy value of -6.33. The algorithm with the worst value is the no local search version with -5.49. In this case, there is a

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

Complex		No LS			Solis-Wets			L-BFGS		
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	**-18.30**	**-14.00**	**4.80**
1bmm	19	-3.94	5.00	11.68	-4.66	5.04	18.84	**-11.99**	**-5.52**	**2.73**
1hvr	17	-15.79	-8.67	9.39	-13.81	-10.13	4.07	**-32.43**	**-32.13**	**0.28**
1nnb	16	-4.53	-2.48	1.69	-5.18	-2.16	1.57	**-6.25**	**-4.63**	**1.50**
1tnh	9	-5.39	-2.73	1.36	-5.37	-2.49	1.14	**-6.06**	**-5.54**	**1.09**
2dbl	13	-10.03	-4.66	3.47	-11.43	-4.26	3.06	**-12.14**	**-10.04**	**2.99**
3ptb	7	-5.49	-3.66	1.02	-5.99	-4.16	1.18	**-6.33**	**-6.22**	**0.25**
7abp	11	-7.97	-6.78	0.90	-7.90	-6.84	1.14	**-9.12**	**-8.67**	**0.37**

difference of only 13.27% while for the larger complex (1adb), for the two same algorithms, the difference is 63.98%. This pattern is consistent when comparing the L-BFGS method to the other two. The larger the complexes, the larger the difference in results. The evolutionary algorithm with the L-BFGS clearly outperforms the other algorithms.

Another relevant aspect is the difference of performance between the two local search methods. As previously reported [3], an evolutionary algorithm without the Solis-Wets local search method can be efficient and competitive. Looking at table 2 we can agree with this conclusion since only in four instances (half the tested complexes) the Solis-Wets algorithm attains better results than without local search. Furthermore, the exact same behavior is exhibited in terms of the average of the best solutions found. This is an indication that these two variants of the evolutionary algorithm are more sensitive to the multi-modal landscape of the problem. The L-BFGS local search method enables the evolutionary algorithm to overcome the multiple local minima more easily.

In addition, the evolutionary algorithm with the L-BFGS method provides another indication that it is superior to the other versions. The column that displays the averages of the best solutions found during the 30 runs, shows for the L-BFGS method closer values to the best. Four instances have a proximity of values very small (0.92% for 1hvr, 1.7% for 3ptb, 4.99% for 7abp and 8.6% for 1tnh). The remaining ones have a small moderate distance (17.28% for 2dbl, 23.5% for 1adb and 25.88% for 1nnb) with the exception of 1bmm (distance of 53.97%). The proximity values between the best and the average are not reported for the other methods. The evolutionary algorithms without local search and with the Solis-Wets method show larger distances, the majority of them between 45% and 65% approximately. Although this indicates a more efficient local search method for the L-BFGS algorithm, it is important to point out that the method also reduces the population diversity substantially.

To establish if these differences are statistically significant, we performed the Wilcoxon rank sum test with significance value $\alpha = 0.01$. As expected, there are no significant differences between the evolutionary algorithms without local

search and the Solis-Wets method. Significant differences are found between the evolutionary algorithm with the L-BFGS local search method and the other two algorithms.

It is important now to relate the energy results with the RMSD values. Table 3 displays the overview of these values. In the same way as before, each line displays the results for a complex. For each of three variants of the evolutionary algorithm, a column is presented with three sub-columns. These are: the best RMSD value found during the 30 runs (Best), the RMSD value correspondent to the best energy value found during the 30 runs (Best-En) and the average of the RMSD values for 30 runs (Avg). Bold indicates the best displayed values.

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

Complex		No LS			Solis-Wets			L-BFGS		
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	**0.30**	**0.51**	**1.64**
1bmm	19	2.06	2.06	4.18	1.29	1.29	3.89	**0.67**	**1.10**	**3.63**
1hvr	17	**0.29**	**0.29**	0.96	0.53	0.65	0.90	0.50	0.71	**0.61**
1nnb	16	0.44	0.76	**1.84**	0.62	**0.64**	2.76	**0.39**	0.74	**1.84**
1tnh	9	0.70	0.73	2.94	0.34	**0.34**	3.38	**0.20**	0.94	**1.16**
2dbl	13	0.57	0.78	2.42	0.53	0.53	2.64	**0.35**	**0.35**	**1.40**
3ptb	7	0.42	0.54	2.07	0.31	**0.31**	1.45	**0.24**	0.51	**0.93**
7abp	11	**0.22**	0.61	0.65	0.26	**0.45**	0.86	0.35	0.82	0.76

In terms of the RMSD values, the differences between the three approaches are not as distinctive as in the energy case. An overview of table 3 reveals that the L-BFGS method does not contain all the best values. Nevertheless, it still contains the majority of the best RMSD values. However, this table contains two important pieces of information. The first one indicates how well the algorithm is able to perfom in terms of RMSD, i.e., can the evolutionary algorithm attain low RMSD values? The answer is given by the Best and Avg columns. From the table, it is possible to conclude that all algorithms can discover solutions with low RMSD values. For example, 0.22 without local search for the 7abp complex, 0.34 with Solis-Wet method and 0.20 with L-BFGS method for the 1tnh complex. Moreover, the Best column reveals that the three evolutionary algorithms are able to reach consistently good values. With the exception of the evolutionary algorithm without local search for the 1hvr complex, every other value is lower than 1.5 and for most cases, even lower than 1.0. This performance is reinforced by the Avg column since the RMSD average of the best solutions found during the 30 runs is low. However, in this case the L-BFGS method attains a better performance compared to the remaining algorithms. The RMSD average on the eight tested instances is 1.5 for the L-BFGS method. The Solis-Wets method has an average of 2.33 and without local search we have a slight improvement: 2.23.

The second important information gathered from this table is 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. For the best energy values in table 2, we present the corresponding RMSD values in the column Best-En. From this column it is possible to conclude that the distribution of the best values is more balanced between the two algorithms with local search methods. The Solis-Wets method has the best values for four instances and the L-BGFS algorithm for three instances. The remaining best value is for the evolutionary algorithm without local search. The first conclusion we can draw from this is that the L-BFGS method is more capable to optimize in terms of energy minimization while the other methods, although with higher energies, can reach solutions with a more similar structure to the optimal case. Nevertheless, the function being optimized is in terms of energy and the L-BFGS method shows that it is capable to reach more efficiently the desired area to search.

This is shown by the scatter plots in figure 1 and the full optimization runs presented in table 4. The plots display for the 1adb complex (without major differences, the same pattern is observed in the remaining instances) 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 all the plots: there is a clear approximation flow to the origin and there is a concentration between RMSD values of 2 and 4 with an energy level below 500000. The main differences between the plots are: 1) the degree of the concentration is higher for the L-BFGS method; 2) the majority of the initial solutions for the L-BFGS method have a RMSD below 8, thus they are presented in the plot; 3) although the Solis-Wet algorithm shows a small focus of solutions in the RMSD level below 2, this effect is stronger with the L-BFGS method. The main information from the these plots is that an optimization run with the L-BFGS method drives more solutions with lower energy and RMSD values closer to the desired point.

A perusal of table 4 helps to consolidate this view. The table presents the data from two complexes, 1adb and 1hvr, which show the best energy and corresponding RMSD for the 30 runs, for the three tested algorithms. The runs are sorted by the energy values. Important differences between the evolutionary algorithms are found. First, the L-BFGS method is able to consistently reach good energy values. This is very clear when compared to any of the other two methods, especially for the later runs. The L-BFGS method still provides good energy values while the Solis-Wets method and no local search display unacceptable energy values. In addition, the same kind of behavior is displayed when observing the RMSD columns. Although less strong in comparison to energy, the L-BFGS method is able to provide lower RMSD values for a longer number of runs. In the case of the 1hvr complex, the L-BFGS algorithm presents values always below 1.0 while for the remaining algorithms this is not true. Furthermore, for the 1adb complex, L-BFGS is the only method capable of attaining RMSD values below 1.0 and for almost half the runs (14 out of 30). Similar patterns are also observed in the remaining complexes optimization runs.

4 Conclusions

We investigated the use of local search methods when applied to the molecular docking problem. Previous research on this topic indicated that the most widely used local search method for docking could be inefficient and, with carefully selected genetic operators, an evolutionary algorithm is sufficient [3]. Our investigation partially supports this view. Although the Solis-Wets algorithm might not be the most suitable local search component in an evolutionary algorithm, other local search methods could prove to be efficient. In this paper, we proposed a standard generational evolutionary algorithm hybridized with the L-BFGS method, a powerful quasi-Newton conjugate gradient method. Results show that an evolutionary algorithm with this method as local search is superior to previous approaches [6, 3]. The optimization results are considerably better in terms of energy and RMSD values. The differences between this algorithm and the other tested approaches are statistically significant.

References

1. Neumaier, A.: Molecular modeling of proteins and mathematical prediction of protein structure. SIAM Review **39** (1997) 407–460
2. 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
3. Thomsen, R.: Flexible ligand docking using evolutionary algorithms: investigating the effects of variation operators and local search hybrids. Biosystems **72** (2003) 57–73
4. 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)
5. Tavares, J., Melab, N., Talbi, E.G.: An empirical study on the influence of genetic operators for molecular docking optimization. Technical Report RR-6660 (2008)
6. 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
7. 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
8. Moitessier, N., Englebienne, P., Lee, D., Lawandi, J., Corbeil, C.: Towards the development of universal, fast and highly accurate docking/scoring methods: a long way to go. British Journal of Pharmacology **153** (2007) 1–20
9. Liu, D.C., Nocedal, J.: On the limited memory method for large scale optimization. Mathematical Programming B (1989) 503–528
10. Tavares, J., Tantar, A.A., Melab, N., Talbi, E.G.: The impact of local search on protein-ligand docking optimization. In: Proceedings of the 8th International Conference on Hybrid Intelligent Systems. (2008)
11. Nelder, J.A., Mead, R.: A simplex method for function minimization. Computer Journal **7** (1965) 308–313

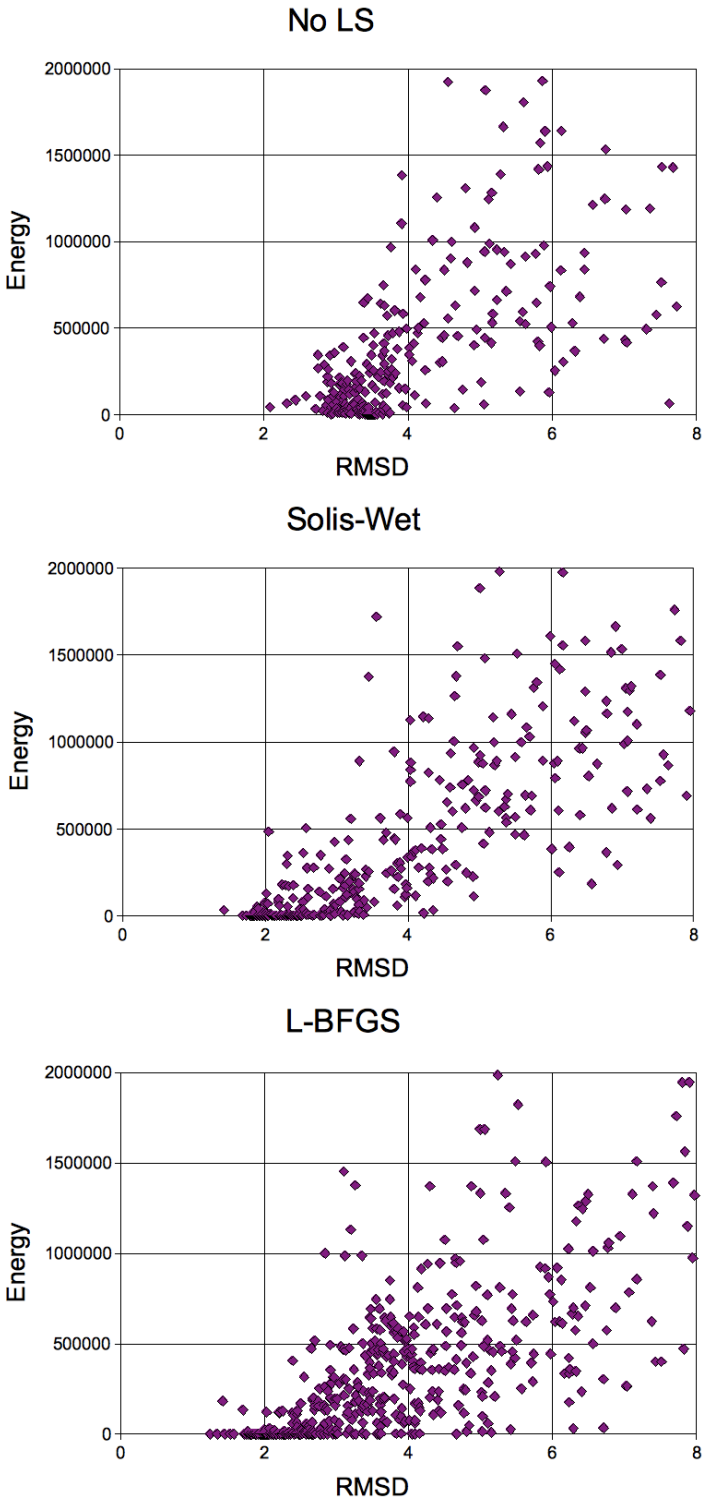

Fig. 1. Scatter plot of an optimization run for the 1adb complex with all variants.

Table 4. Full runs for the 1adb and 1hvr instances, sorted by best Energy values.

Runs	1adb						1hvr					
	No LS		Solis-Wets		L-BFGS		No LS		Solis-Wets		L-BFGS	
	Energy	RMSD	Energy	RMSD	Energy	RMSD	Energy	RMSD	Energy	RMSD	Energy	RMSD
1	-6.59	1.53	-3.85	1.50	**-18.30**	0.51	-15.79	**0.29**	-13.81	0.65	**-32.43**	0.71
2	-1.51	2.23	-0.28	2.06	-18.25	0.48	-15.58	0.73	-13.70	0.66	-32.42	0.62
3	-1.19	1.81	2.50	1.34	-18.18	0.40	-15.52	0.48	-13.36	0.65	-32.34	0.62
4	0.04	1.47	7.30	1.98	-18.15	**0.30**	-15.30	0.67	-13.26	0.68	-32.31	0.70
5	8.25	2.21	8.67	2.75	-18.08	0.45	-15.22	0.71	-13.16	0.53	-32.30	0.58
6	9.42	2.35	9.44	3.50	-18.27	0.41	-14.42	0.86	-13.11	0.61	-32.27	0.55
7	12.08	2.64	9.45	3.18	-18.15	0.55	-14.17	0.89	-12.96	0.66	-32.26	0.55
8	22.59	1.85	14.13	3.28	-17.92	0.49	-13.96	0.75	-12.92	0.66	-32.26	0.60
9	32.85	3.29	14.25	1.40	-17.84	0.39	-13.88	0.84	-12.81	0.66	-32.25	0.59
10	36.87	2.39	15.43	2.10	-17.79	0.48	-13.75	0.85	-12.80	0.88	-32.25	0.69
11	37.38	2.79	16.30	2.40	-17.72	0.49	-13.26	0.80	-12.59	0.57	-32.20	0.63
12	40.45	1.71	16.45	1.39	-17.58	0.44	-12.98	0.92	-12.41	0.73	-32.19	0.69
13	47.21	1.83	17.55	2.76	-17.43	0.42	-12.90	0.89	-12.36	0.86	-32.08	0.56
14	55.59	2.86	24.19	2.32	-15.97	0.89	-12.73	1.01	-12.04	1.23	-32.03	0.50
15	56.95	4.32	43.29	2.21	-15.58	3.71	-12.72	0.93	-11.58	0.72	-31.97	0.56
16	63.86	1.99	49.75	2.22	-15.55	1.36	-12.51	0.77	-11.48	0.76	-31.35	0.72
17	64.04	2.89	59.49	4.12	-15.32	2.94	-12.51	1.18	-11.09	0.98	-32.34	0.60
18	65.29	2.56	76.65	1.91	-13.89	1.59	-12.37	1.20	-10.85	0.93	-32.21	0.65
19	72.67	3.60	88.04	3.19	-13.86	1.61	-12.26	0.94	-10.66	1.12	-32.20	0.56
20	91.42	4.18	93.40	2.29	-12.76	2.16	-9.31	1.05	-10.10	1.16	-32.15	0.58
21	115.83	2.81	98.69	7.91	-12.37	2.39	-8.28	0.95	-10.07	0.94	-32.13	0.58
22	162.20	2.85	100.14	3.24	-12.05	2.27	-7.90	1.06	-9.82	0.86	-31.28	0.67
23	169.17	2.80	101.90	2.58	-11.92	1.90	-7.05	0.93	-9.70	1.16	-32.30	0.57
24	213.00	2.87	137.47	2.76	-11.03	2.78	-5.89	1.06	-9.62	1.06	-32.23	0.58
25	252.60	3.76	154.18	2.08	-9.16	2.85	-4.50	1.52	-7.24	1.38	-32.17	0.60
26	257.58	2.75	225.49	2.14	-8.02	3.92	-2.32	0.93	-5.76	1.30	-32.06	0.58
27	299.62	2.56	301.42	3.69	-7.65	2.80	1.89	1.49	-3.40	1.07	-32.22	0.58
28	423.42	3.42	314.32	3.21	-6.83	2.97	6.15	1.07	-1.21	1.12	-32.20	0.60
29	2900.60	4.21	721.81	3.84	-2.25	3.52	14.49	1.42	-0.57	1.29	-32.07	0.59
30	4558.50	4.85	3053.11	4.42	-2.13	3.83	24.45	1.55	0.61	1.17	-31.44	0.74
Avg	335.34	2.78	192.36	2.79	**-14.00**	**1.64**	-8.67	0.96	-10.13	0.90	**-32.13**	**0.61**
Std	953.64	**0.87**	559.51	1.26	**4.80**	1.26	9.39	0.28	4.07	0.25	**0.28**	**0.06**