SAND99-0225C

Optimization in Computational Chemistry and Molecular Biology, pp. #-#
C. A. Floudas and P. M. Pardalos, Editors

Improved Evolutionary Hybrids for Flexible Ligand Docking in Autodock

William E. Hart
Sandia National Laboratories
P.O. Box 5800, MS 1110
Albuquerque, NM 87185 USA
wehart@sandia.gov
(619) 844-2217

Chris Rosin
Department of Molecular Biology, MB-5
The Scripps Research Institute
La Jolla, CA
crosin@scripps.edu

Richard K. Belew
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA
rik@cs.ucsd.edu

Garrett M. Morris
Department of Molecular Biology, MB-5
The Scripps Research Institute
La Jolla, CA
garrett@scripps.edu

Abstract

In this paper we evaluate the design of the hybrid EAs that are currently used to perform flexible ligand binding in the Autodock docking software. Hybrid evolutionary algorithms (EAs) incorporate specialized operators that exploit domain-specific features to accelerate an EA's search. We consider hybrid EAs that use an integrated local search operator to refine individuals within each iteration of the search. We evaluate several factors that impact the efficacy of a hybrid EA, and we propose new hybrid EAs that provide more robust convergence to low-energy docking configurations than the methods currently available in Autodock.

Keywords: drug docking, ligand binding, hybrid evolutionary algorithms, global optimization, local optimization.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

1 Introduction

Computational methods for molecular docking are valuable tools for structure-based drug discovery. Methods for automated docking fall into two broad categories: matching methods and conformational search methods. Matching methods attempt to find a good docking based on the geometry of the docking molecule and receptor site. The DOCK program [12] was one of the first matching methods developed, and current versions of it are still used. Conformational search methods typically model the ligand in greater detail, and they often allow conformational flexibility in either the ligand or receptor site, or both. These methods employ a simulation or optimization method to search through the space of ligand-receptor configurations.

Autodock [7, 15] is an example of this approach to molecular docking. It uses a physically detailed model that allows for a fixed receptor site and flexible ligand. Autodock employs a rapid grid-based method for energy evaluation and precalculates ligand-protein pairwise interaction energies so that they may be used as a look-up table during the conformational search. Autodock has been successfully applied to a variety of applications [6] using a simulated annealing search method.

More recently, evolutionary algorithms (EAs) have been incorporated into Autodock and applied to standard test problems [17, 14]. EAs have become a popular choice for heuristic search in docking applications [1, 24], and in our evaluation of EAs with Autodock they consistently perform better than simulated annealing. The molecular docking problem solved by Autodock is a challenging global optimization problem, and the EAs perform a better global search across the range of positional, orientational and conformational parameters for flexible ligands. Two forms of EAs can currently be used with Autodock: a genetic algorithm [5] and a hybrid EA that uses local search. The hybrid EAs apply local search in each iteration to refine points. Rosin et al. [17] and Morris et al. [14] show that this local refinement can significantly improve the performance of the EA.

In this paper we reconsider the design of these hybrid EAs. Specifically, we evaluate several factors that may impact the efficacy of these methods. First, we describe a new local search method that has more robust convergence properties than the method previously used with Autodock. Next we consider the length of local search, which impacts the balance between global sampling and local refinement in a hybrid EA. Finally, we consider the initial step length used by the local search method, which can be dynamically initialized using population statistics from the EA. We empirically evaluate the effects of these factors on the performance of hybrid EAs using standard test problems. Our results incdicate that hybrid EAs using the more robust local search are usually better, and that running the local search method longer improved the search. Initializing the local search step length did not appear to be an important factor, although using this approach can avoid certain worst-case scenarios where the fixed initial step length is poorly initialized.

2 Autodock

Autodock docks small flexible substrate molecules to large rigid macromolecules like proteins [15]. A candidate docking gives specific positions and orientations for the protein and a small molecule. Autodock uses an approximate physical model to compute the energy of a candidate docking, and uses a heuristic search to minimize this energy. This method makes most sense when there is a single docked configuration that is at a much lower energy than other configurations, so that we expect this low-energy configuration to be the consistent result of physical interaction between the two molecules. If the prediction of this configuration is to be accurate, the energy function must have its global minimum at or near this physical configuration.

Heuristic search operates on the configuration of the small molecule, assuming (without loss of generality) a fixed position for the protein. The small molecule can take any position around the protein, and can have any orientation. Global orientation is expressed as a quaternion, which can be thought of as a vector giving an axis of rotation, along with an angle of rotation about this axis. The small molecule may also have several internal rotatable bonds so that its shape is somewhat flexible. The representation of a candidate docking consists of 3 coordinates giving the position of the small molecule, followed by the 4 components of the quaternion specifying the overall orientation of the small molecule, followed by one angle for each of the rotatable bonds.

The docking potential used in Autodock 3.0 is an empirical free energy potential. This energy potential is composed of five terms (see Morris et al. [14] for further details). The first three are pairwise interatomic potentials that account for short-range electrostatic repulsive forces and long-range weak van der Waals attractive forces. The standard Lennard-Jones 12-6 potential is used for the van der Waals forces, and a 12-10 potential is used for hydrogen bonds. The next term measures the unfavorable entropy of a ligand binding due to the restriction of conformational degrees of freedom, using a measure that is proportional to the number of sp^3 bonds in the ligand. The last term uses a desolvation measure adapted from Stouten et al. citeStoFroNakSan93 which works well with the precalculated grid formulation used by Autodock.

To account for internal energy in a flexible small molecule with internal rotatable bonds, we calculate the same energy contributions summed over all pairs of atoms within the small molecule. This sum is added to the total energy evaluation. This penalizes conformations of the small molecule that are energetically unfavorable independent of their interaction with the macromolecule.

To save time when computing energy of interaction with the macromolecule, 3-D potential grids are computed for each atom type before optimization begins. Interaction energy is computed as described above at each point in the grid. Then, when calculating total energy during optimization, the energy contribution of an atom is obtained via trilinear interpolation of its position within the grid specific to its atom type, based on the values at the nearest 8 points in the grid. Calculation of the energy due to pairwise interactions within the small molecule does not make use of these grids.

Computation of the grids for energy evaluation requires knowledge of the (assumed fixed) 3-D positions of each atom in the protein; these positions are usually obtained by X-ray crystallography. We also require the structure of the small molecule, along with the locations of internal rotatable bonds. Small molecules tend to be chemically simple, so that we can determine their structure (at least up to the degrees of freedom represented by the rotatable bonds) from their chemical composition alone. Partial charges are required to calculate electrostatic interaction potentials, but these partial charges can be computed from the structure with molecular modelling software such as MOPAC. So, it is possible to use Autodock to test many candidate small molecules against a single target protein, after obtaining the structure of this protein experimentally. This makes Autodock an important computational tool in the initial stages of drug design.

3 Optimization Methods

Docking is a difficult global optimization problem, and a variety of different optimization strategies have been proposed to solve docking problems (e.g. see [24, 1]). Simulated annealing [11, 23] is the first optimization method that was used to perform docking with Autodock [7, 15]. Simulated annealing borrows from the natural metaphor of cooling metal in an attempt to globally minimize functions. Simulated annealing operates much like a steepest descent algorithm, but where a steepest descent algorithm rejects all inferior points, simulated annealing may accept an inferoir point with probability p. This probability p is based on the inferiority of the alternate point and a temperature parameter:

$$p = e^{-\frac{\Delta E}{kT}},$$

where ΔE is the size of the energy gain, and k is the Boltzmann constant. A cooling schedule lowers temperature during the course of optimization. Simulated annealing does a more global search in early iterations, when high temperature allows transitions over energy barriers from one valley to another. In later iterations, the temperature becomes low, which places more focus on a local optimization within the current basin of attraction.

Subsequently, genetic algorithms [5, 2] have been used by Hart [8], Rosin et al. [17] and Morris et al. [14] to perform docking with Autodock. Genetic algorithms are evolutionary algorithms (EAs), which perform a multi-point search based on the mechanisms of natural evolution. Specifically, EAs utilize stochastic competition and multi-point recombination, which reflect the mechanisms of natural selection and sexual recombination. Figure 1 provides a basic overview of the main steps of an EA. In each iteration, an EA uses stochastic competition to select a subset of points from its current *population* of points. This subset is used to generate new points using evolutionary operators like recombination and mutation. Recombination generates a new point from two points, often forming a point that represents a convex combination of these two points. Mutation generates a point by varying a subset of the point's parameters. The set of new points generated by the evolutionary operators are typically used to form the population for the next iteration of EA. Although the stochastic competition tends to focus an EA's search, these methods often perform a robust global search because they sample across multiple points in a search domain.

```
Uniformly generate $P_0$
For $t = 1,2,\ldots$
   Stochastic competition generates $\overline{P}_t$ from $P_{t-1}$
   Apply recombination and mutation to points in $\overline{P}_t$ to generate $P_t'$
   Apply local search to points in $P_t'$ to generate $P_t''$
   Compose $P_t$ from $P_t''$ and $P_t'$
EndFor
```

Figure 1: High-level description of the major steps in an EA. Local search is not used in a canonical EA, but it is used in a hybrid EA.

Hart [8], Rosin et al. [17] and Morris et al. [14] also consider hybrid EAs that use local search. These EAs apply local search to a subset of the points generated by the evolutionary operators in an EA in each iteration. The motivation for these hybrids is that these methods could decompose the search by allowing the EA to globally sample across the range of possible docking configurations while the local search method quickly minimizes points to find locally optimal configurations. Hart, Kammeyer and Belew [8, 10] argue that these types of hybrid EAs are better global optimizers than either EAs or local search separately, and Törn and Žilinskas [22] note that most successful global optimization methods also apply the same principle of distinguishing the mechanisms for global and local search.

Global-local search hybrids may be especially effective for docking. We believe that there are multiple locations on the surface of the macromolecule where the small molecule could dock, and multiple orientations of the small molecule that are energetically plausible. Local search can reveal which of these locations and orientations is best by fitting the small molecule as closely as possible to the macromolecule within a small local neighborhood of a coarse location and orientation. But we do not expect smooth hills in energy from one location and orientation to a very different one, so that global search is required to choose among these. A global-local search hybrid can effectively sample distant locations and orientations with global search, and get accurate evaluations of each using local search.

The particular hybrid EAs that we have applied to docking are *Lamarckian* hybrid EAs. In Lamarckian hybrid EAs, the points used to start a local search are replaced by the final point generated by the local search. This is in contrast to a *Darwinian* hybrid EA, which simply gives the starting point the value of the final point (which generally increases the probability of selecting the point in the subsequent stochastic competition). Although Darwinian hybrid EAs are more biologically plausible, our previous work [10] leads us to use Lamarckian GA-LS hybrids here.

The method used to perform local search in these hybrid EAs is Algorithm 1 from Solis and

Wets [19]. The Solis-Wets method is a direct search method that performs a randomized local minimization. Each step starts with a current point x. A deviate δ is chosen from a normal distribution whose standard deviation is given by a parameter Δ_t. If either $x+\delta$ or $x-\delta$ is better, a move is made to the better point and a "success" is recorded. Otherwise a "failure" is recorded. After several successes in a row, Δ_t is increased to move more quickly. After several failures in a row, Δ_t is decreased to focus the search. Additionally, a bias term is included to give the search momentum in directions that yield success. This method is typically terminated if Δ_t falls below a given threshold Δ_{lb}.

An important feature of this type of local search is that it doesn't rely on gradient information. This is particularly important for docking with Autodock because the docking potential is not differentiable throughout the entire search domain. Autodock's grid-based intermolecular energy has a gradient that is undefined whenever an atom is on a grid boundary, and has discontinuities as atoms move across grid boundaries. This would make gradient-based local search a poor choice for Autodock.

4 Hybridization Issues

Although hybrid EAs using local search have been successfully applied to a range of applications, basic principles have not been formulated to guide the development of effective hybrids. This is particularly true for hybrid EAs applied to continuous search domains like the docking problem in Autodock. In this section we discuss three hybridization issues that may affect the performance of the hybrid EAs used in our docking experiments.

Local Search Robustness Since direct search methods do not explicitly employ derivative information, it is not possible for a direct search method to terminate with a guarantee that the final point is near a stationary point (where the gradient is zero). However, a basic expectation of a local search method for a hybrid EA is that it robustly converges to a stationary point. Solis and Wets [19] note that their algorithm will converge to a global minimizer if the step scale parameter is not adapted. However, in Algorithm A the step scale is adapted, and they argue that "a proof of convergence is unlikely."

A class of direct search methods that do have robust convergence properties are pattern search methods. Pattern search methods have been analyzed by Torczon and Lewis [21, 13], and they provide a general framework for describing a wide variety of direct search methods. In a general sense, pattern search methods sample the objective function from a given *pattern* of points that represent offsets from the current best point. If there is a better point in this pattern, then it is accepted as the new iterate and the sampling is repeated about it. If not, then the scale of the pattern is reduced (e.g. by halving it), and the function is again sampled about the best point. Lewis and Torczon's analysis provides a set of conditions which, if satisfied, guarantee a weak stationary point convergence. For unconstrained pattern search methods, they show that

$$\liminf_{t\to\infty} \| \bigtriangledown f(x_t) \| = 0,$$

where $\bigtriangledown f$ is the gradient of f.

A formal description of the pattern search algorithm used in our experiments is given in Figure 2. The pattern used in this algorithm is a set of offsets that form a positive basis [13]. Specifically, this method uses offsets generated by the $n+1$ directions from the centroid of a regular simplex to each corner of the simplex [20]. These offsets lie at the corners of a triangle in two dimensions, the corners of a tetrahedron in three dimensions, and so on. Our implementation of pattern search uses a **shuffle** method to randomly select the order in which these offsets are considered.

```
Given $x_0$ and $\Delta_0$
Let $\{v_1, \ldots, v_{n+1}\}$ be the canonical minimal positive basis
For $i = 1, \ldots, n+1$
  $r_i = i$
For $t = 1, 2, \ldots$
  shuffle($r$)
  $s_t = \{0\}^n$
  For $i = 1, \ldots, n+1$
    $s = v_{r_i}$
    If($f(x_t + \Delta_t s) < f(x_t)$)
      $s_t = \Delta_t s$
      Break
    EndIf
  EndFor
  $x_{t+1} = x_t + s_t$
  If ($f(x_t) == f(x_t + s_t)$)
    $\Delta_{t+1} = \Delta_t / 2$
  Else
    $\Delta_{t+1} = 2\Delta_t$
  EndIf
  If ($\Delta_{t+1} < \Delta_{lb}$)
    Break
EndFor
```

Figure 2: Pseudo-code for the pattern search algorithm.

Local Search Length When local search is used in a hybrid EA, the local search is typically truncated before the termination criteria stop the local search (e.g. when the step length becomes too small). Thus the maximum length of local search is a parameter that can fundamentally impact the dynamics of the hybrid EA. An EA hybrid with long local searches will execute fewer iterations of the EA than an EA hybrid with short local searches (if both terminate after the same number of function evaluations). Consequently, the length of local search affects the balance between the amount of global sampling and local refinement performed by the hybrid EA.

In previous work, we have used both short and long local search lengths for docking in Autodock [17, 14]. We previously experimented with very short and very long local search lengths in Autodock, and found them to yield similar performance [17]. However subsequent experiments with the new energy potential in Autodock suggest that this may be an important factor, particularly if the local search is only terminated when it exceeds the local search length.

Initial Search Scale A fundamental feature of optimization problems on continuous domains is that the scale of changes in the objective function can vary dramatically in different regions of the domain. Consequently, a basic requirement of an optimization method is that it dynamically adapt the scale of its search in order to match the scale of changes in the objective function. This implies that hybrid EAs using local search need to dynamically adapt the initial scale of the local search method. If the initial scale of the local search is not adapted, then the scale of the entire hybrid is limited by how quickly the local search method adapts its scale. This limits the utility of local search, and in cases where the local search is truncated it can even prevent the local search method from productively refining a point.

Thus hybrid EAs that use direct search methods for local search need to adapt the initial local search scale to reflect the characteristics of the current search. To our knowledge, Mühlenbein, Schomisch and Born [16] are the only authors to have reported an initialization strategy for hybrid EAs using a direct search method. Their initialization strategy uses a statistic of the population spread to initialize all local searches with the same value. The spread of the population reflects the degree to which the EA's search has focused on a particular region of the search domain. If we assume that the local search should be as focused as the EA's search, then this represents a natural parameter for the initial local search step scale. Their method initializes Δ_0 to

$$\Delta_0 = \frac{|x_1 - x_m|}{\sqrt{n}},$$

where x_1 is the point with the best fitness in the population and x_m is the point with the median fitness.

5 Methods and Experiments

5.1 Search Algorithms

Our experiments compare the empirical performance of hybrid EAs on a set of standard docking problems. The hybrid EAs that we evaluate are the hybrid GA using SW that is provided in Autodock [14] and hybrid GAs using the Solis-Wets method (SW) or pattern search method (PS) that are provided by the SGOPT optimization library [9], which was integrated into Autodock 3.0 for this study. In all cases, the experimental parameters for the hybrid GAs were the same as those used by Morris et al. [14]. The GAs used a population of 50 points, applied a two-point crossover with a probability of 80% and applied a Cauchy mutation operator with a probability of 2%. When the Cauchy mutation operator is applied to a dimension of a point, it adds a Cauchy random variable with parameters $\alpha = 0$ and $\beta = 1$. The hybrid EAs provided by SGOPT scale the mutation steps to 0.01 times the range of the dimension. Stochastic competition was performed using proportional selection, where the baseline for computing the proportions was the worst point in the last 10 iterations [14]. Elitism was also used to keep the best point found so far.

Local search was performed on randomly chosen points in each iteration with a probability of 6%. Using local search infrequently is motivated by our preliminary work with this application where we varied the probability of local search [8]. The local search operator was either SW or PS. The SW method used in Autodock implicitly scales the search in each dimension to a value that is approximately 0.01 times the range of the dimension. The lower bound on Δ_t is effectively zero for this method due to an incorrect implementation of SW in Autodock; SGOPT correctly implements this lower bound. SW performs contraction after four consecutive failures and it performs expansion after four consecutive successes. The SW and PS methods provided by SGOPT also scale the search in each dimension. The initial search scale is 0.1 times the range of each dimension, and the SW and PS methods were terminated when the search scale fell below 0.001.

In each experiment, 20 trials were done with different random seeds. The hybrid EAs were terminated after 1.5 million function evaluations; this enables a comparison with previous work [17, 14]. Using a fixed number of function evaluations provides a reasonable basis of comparison for this problem because the calculation of the docking potential is the most expensive step in this optimization problem.

For notational convenience, we refer to the different hybrid EAs using the notation [ea]-[ls]. The values for ea are AD, the GA provided by Autodock, GA0, the GA provided by SGOPT with fixed initial step length and GA1, the GA provided by SGOPT using the initialization of Mühlenbein et al. [16]. The value of ls is SW300, SW3000, PS300 or PS3000, which refer to the choice of local search method and the maximum length of local search.

5.2 Experiments

A test suite of six cases was used in all of the experiments. Each test case consists of a macromolecule and a small substrate molecule. The salient features of the six test cases are summarized in Table 1. The different test cases were selected to test various aspects of the energy function [15].

Ligand/Protein Complex	PDB Shorthand	Number of Torsions	Number of Dimensions
β-Trypsin/Benzamidine	3ptb	0	7
Cytochrome P-450cam/Camphor	2cpp	0	7
McPC-603/Phosphocholine	2mcp	4	11
Streptavidin/Biotin	1stp	5	12
HIV-1 protease/XK263	1hvr	10	17
Influenza Hemagglutinin/sialic acid	4hmg	11	18

Table 1: Summary of test cases.

The number of torsion angles is an important feature of these test cases because it determines the dimensionality of the search space. The representation used in each experiment consisted of a triple of Cartesian coordinates, a four dimensional quaternion, and the torsion angles. Thus, the dimensionality of the search space is 7+(number of torsion angles). The range of the coordinates defines a cube that is 23 angstroms long in each dimension. The quarternion parameters lie within $[-1,1]^3 \times [0,2\pi]$, and each torsion angle lies within $[-\pi,\pi]$; the points in the initial population have each parameter generated randomly in its range.

For each method on each test case, we consider the minimum energy produced by the search. Because we have crystallographic structures of the true docked complex for each test case, we also measure the absolute accuracy of the final docked configuration. This is done by taking the square root of the average squared deviation of the atoms in the predicted configuration from the crystallographic configuration.

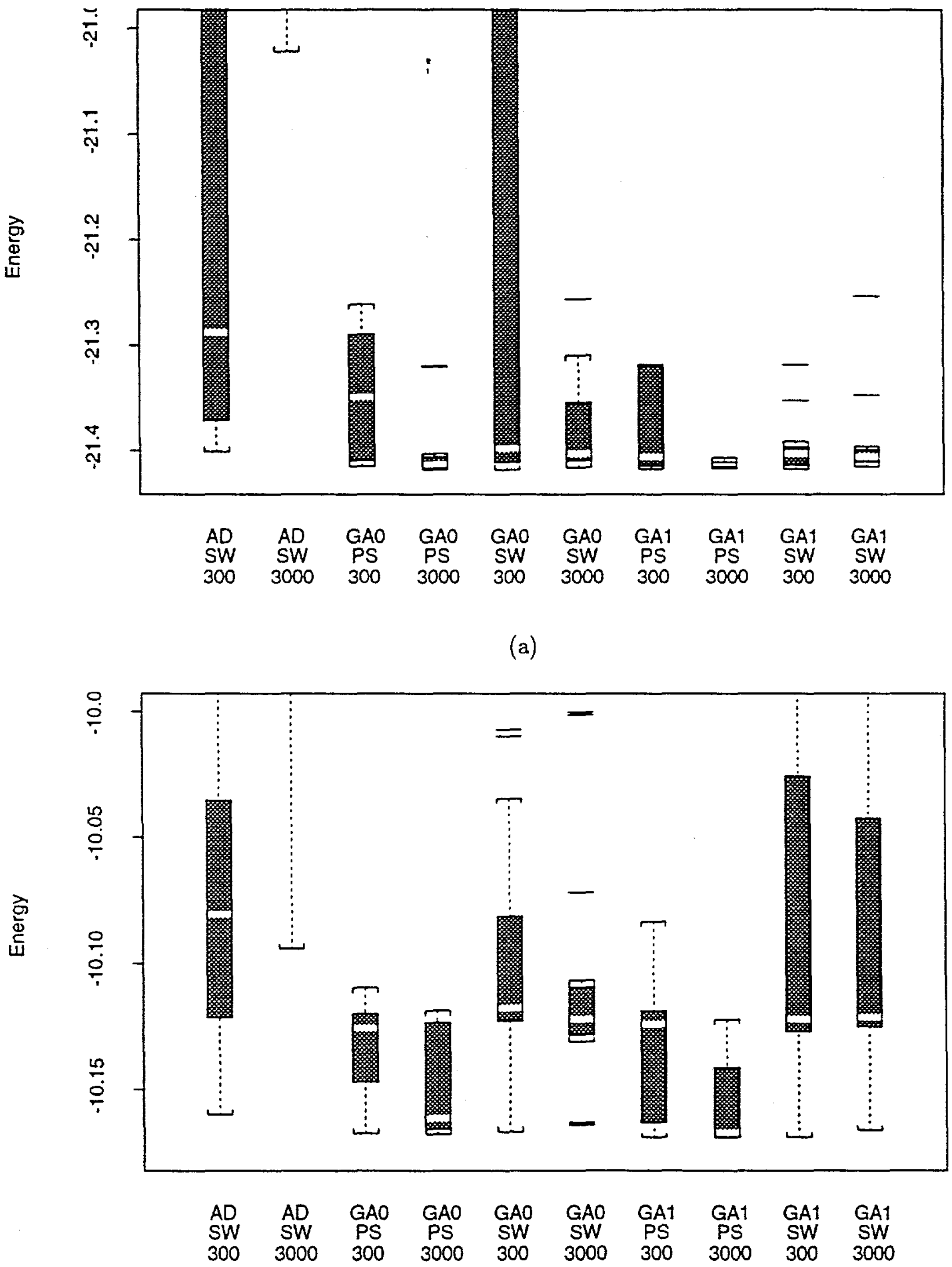
6 Results

Figures 3, 4 and 5 show boxplots of the final docking energies for each of the optimization experiments for each test case. Boxplots are a convenient method of summarizing data that provide a visual indication of the spread and skewness of the data. The dark bar in the boxplots show range between the first and third quartile; one quarter of the data is below the first quartile, and three quarters of the data is below the third quartile. The white line inside the dark bar represents the median. The whiskers at the top and bottom of each boxplot indicate the spread of the data up to 1.5 times the range of the first and third quartile.

We applied the nonparametric Kruskal-Wallis test [4] to identify significant differences between the hybrid EAs (at the 5% level). This test is appropriate because the data does not appear to be normally distributed, and because we are making multiple pairwise comparisons among more than 2 samples. These statistics are discussed in the following sections, which evaluate the effects of the hybridization issues that we discussed in Section 4.

6.1 Autodock vs SGOPT

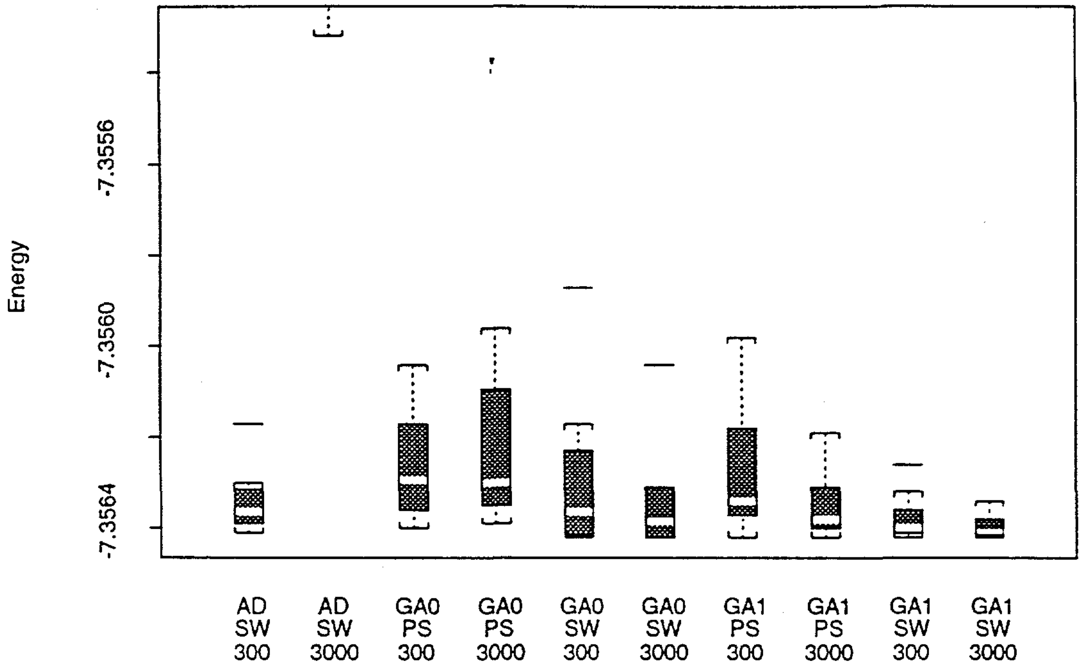
We can make a direct comparison between the hybrid EAs using SW provided by Autodock and SGOPT. Figure 6 shows boxplots for the hybrid EAs using SW for all six test cases. For each test case, the trials for the hybrid EAs are ranked, and the boxplots show the distribution of ranks



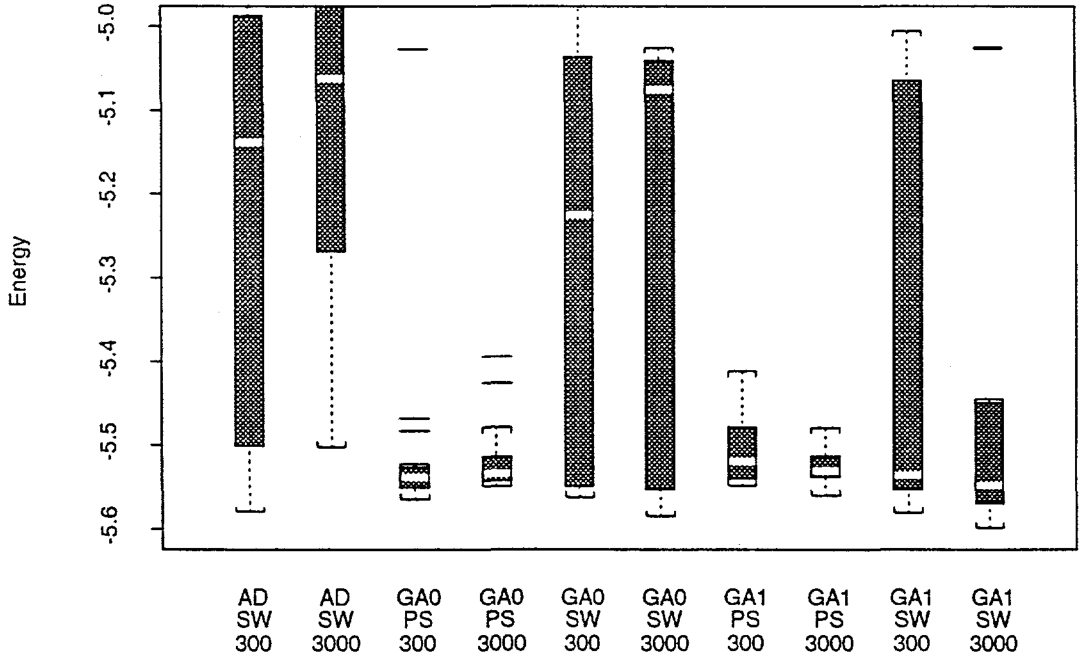
(a)

(b)

Figure 3: Final docking potentials for hybrid EAs: (a) 1hvr and (b) 1stp.

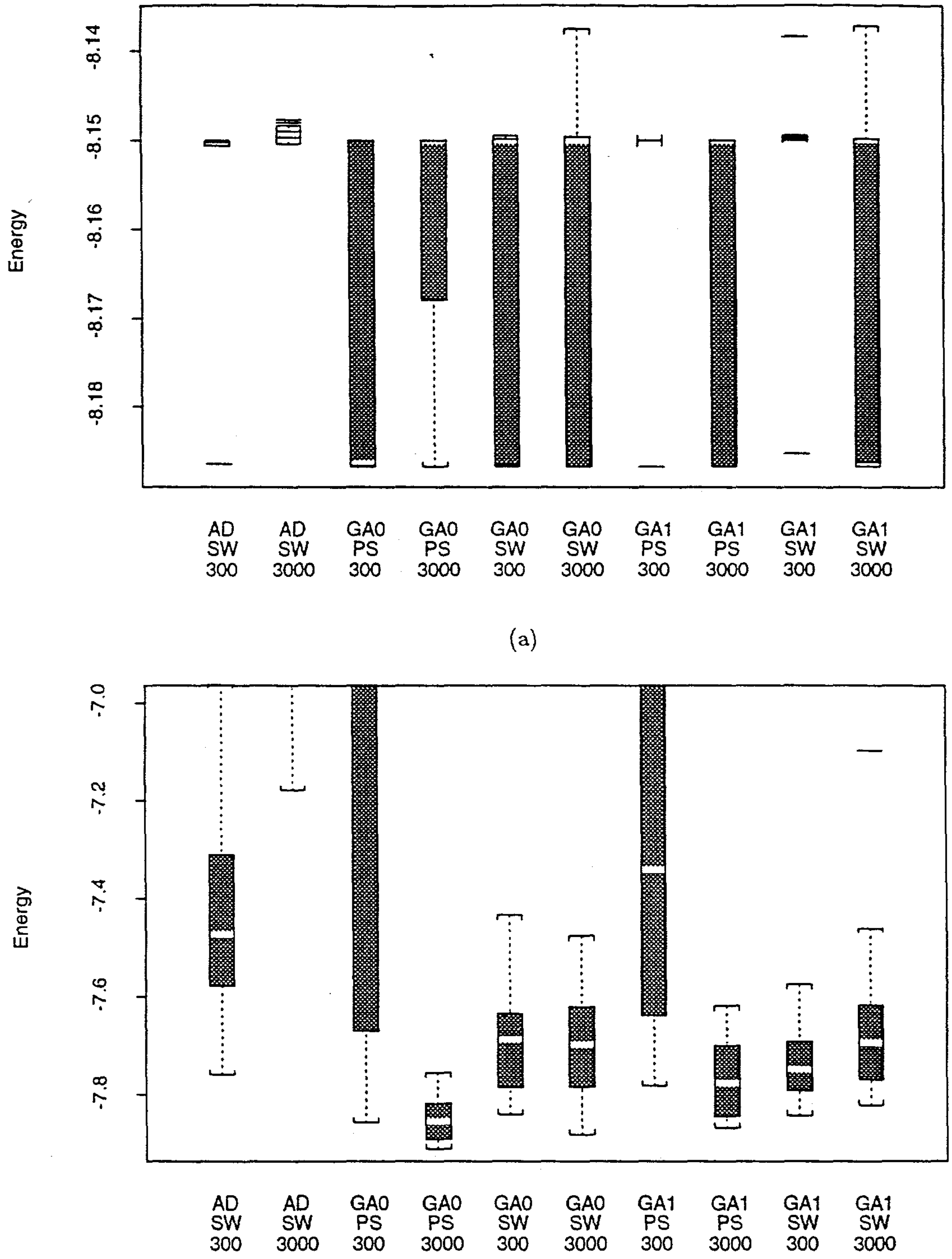


(a)



(b)

Figure 4: Final docking potentials for hybrid EAs: (a) 2cpp and (b) 2mcp.



(a)

(b)

Figure 5: Final docking potentials for hybrid EAs: (a) 3ptb and (b) 4hmg.

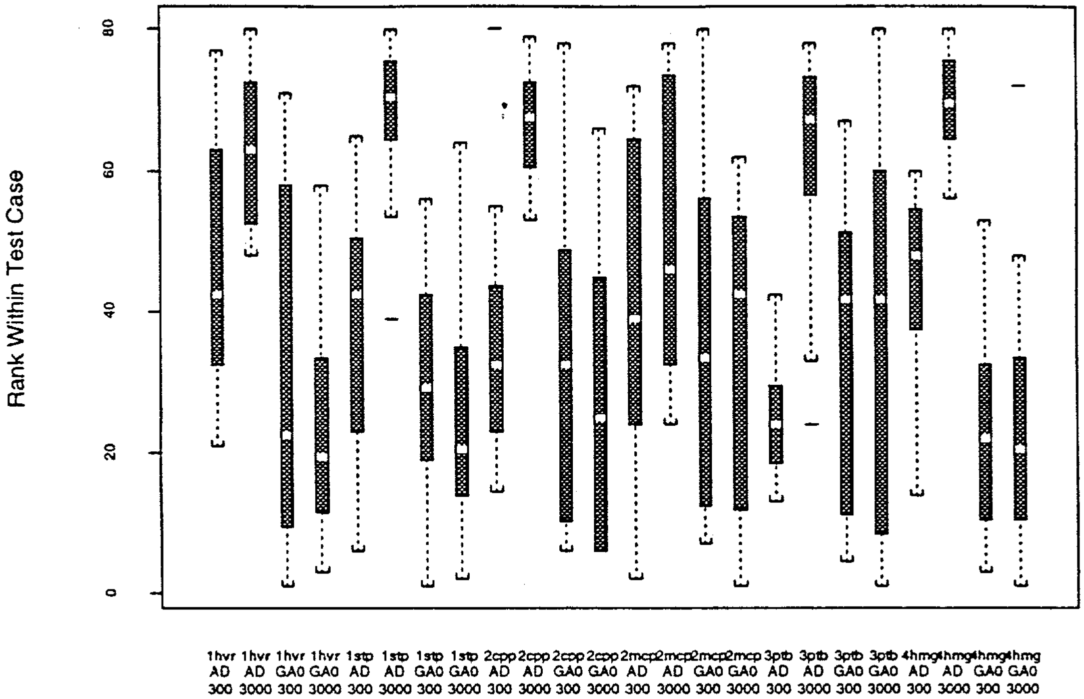


Figure 6: Comparison of rankings for Autodock hybrid EAs and SGOPT hybrid EAs using SW.

within each test case. This data has two distinct trends. First, the hybrid EAs from SGOPT find better solutions. Second, the hybrid EAs from Autodock find better solutions when the local search is short, while the EAs from SGOPT find better solutions when the local search is long. Our statistical analysis shows that the Autodock hybrid EAs are significantly different from almost all of the SGOPT hybrid EAs.

We believe that these differences may be explained by two factors. First, the initial step scale used by the Autodock hybrid EAs is smaller than the initial step scale of the SGOPT hybrid EAs. Although a smaller initial step scale focuses the local search about the starting point, in early iterations of the hybrid EA the local searches are not simply refining to a local minimum, but they are also helping guide the EAs search (through the Lamarckian return of the final point into the EA's population).

A second factor concerns Autodock's incorrect implementation of the lower bound on step scale. When the local search converges about a local minima, it will waste time refining the minima beyond the point where it is physically relevant. This effect will be particularly evident when the local search length is long, which accounts for the worse performance for the Autodock hybrid EAs with long local search.

6.2 SW vs PS

These experiments generally support the use of the more robust PS method over SW for local search. Here, robustness takes into account both the ability to find the lowest energy conformations as well as the consistency at which the hybrid EA can find low energy conformations. The statistical analysis shows many significant differences between hybrid EAs using PS and SW local search, particularly when long local searches were used.

Figure 7 summarizes the relative ranks of each hybrid EA with SW and PS local search, grouped

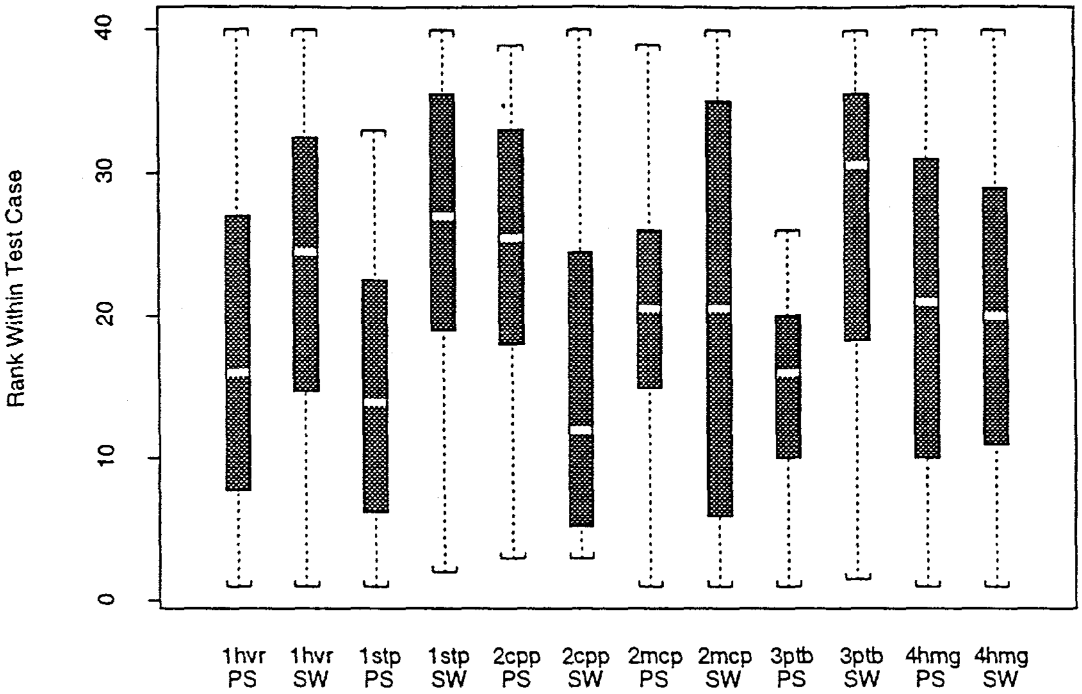


Figure 7: Comparison of rankings for SGOPT hybrid EAs using SW and PS local search.

by the test case. The trend is particularly strong for 1hvr, 1stp and 3ptb. For 2cpp the EAs using PS are consistently better than the EAs using SW, but the EAs using SW can find better solutions often enough to balance the overall ranking. For 4hmg, the EAs using PS are better when long local searches are used, but worse when short local searches are used, which again balances the ranking. For 2mcp, the hybrid EAs using PS are generally worse than the EAs using SW. However, if the termination threshold for PS is reduced to 0.0001 then the hybrids using PS are generally better, which again provides evidence that longer local searches with PS are better.

6.3 Length of Local Search

In the significance test, the hybrid EAs using SW do not consistently exhibit significant differences. However, the significance test shows a consistent significant difference between the hybrid EAs using short PS and long PS for test cases 1stp, 1hvr and 4hmg. These are the most difficult test problems, which suggests that hybrid EAs using long local searches will be most effective for nontrivial problems.

6.4 Step Length Initialization

The experimental results indicate that adaptive methods for initializing the step length of local search do not have a substantial impact on the performance of the hybrid EAs. In fact, there are many cases where initializing the local search length leads to worse performance. The adaptive hybrids have better performance with SW local search in several of the test cases, but in others they have worse performance. The adaptive hybrids with PS are generally indistinguishable from the nonadaptive hybrids. We also tried an alternative initialization strategy for local search, but it had similar results.

We investigated possible causes of this, and we discovered that the initialization methods often generate initial step lengths that are larger than the fixed value used by the GA0 hybrids. Also, we noted that the initial step lengths do not seem to converge to zero as we had anticipated. Instead, the variations in the population remain large enough to keep the adaptively determined initial step lengths rather large. This suggests that the dynamics of the EA have a significant impact on the utility of these adaptive mechanisms. For example, the linear rank selection used by Mühlenbein et al. [16] may be an important feature of the GA for the success of the adaptive methods.

Finally, it is possible that we were "lucky" and selected a good default value for the initial step lengths. In preliminary experiments, we confirmed that as the initial step lengths are raised the performance of the hybrid EAs using the fixed initial step length degrades. This was particularly true for the hybrid EAs using short LS, which is expected since they have fewer iterations to adapt their search scale.

7 Conclusions

Our experimental results demonstrate that the new hybrid EAs we have defined can perform significantly better than the methods currently available in Autodock. In particular, the hybrids using the PS method with long local search were significantly better than other hybrids in most cases. The dynamic initialization of the local search step was not a signficant factor in our experiments. However, it is clear that this is an important algorithmic factor that needs to be developed to make hybrid EAs with direct search methods more robust.

Table 2 summarizes the performance of the GA0-PS3000 methods for the six test cases. In five of the six test cases, the best value found by the hybrid EA is better then the best value found by all of the methods reported by Morris et al. [14]. Furthermore, the mean rmsd of the solutions found is lower for all test cases than the results in Morris et al. In all cases the hybrid EA found solutions below or about 1 angstrom, which is very reasonable. This confirms that these hybrid EAs not only perform a more effective search, but they are also finding better solutions to the docking problem.

	Energy		RMSD	
	Minimum	Mean	Minimum	Mean
1hvr	-21.42	-21.40	0.65	0.70
1stp	-10.17	-10.15	0.49	0.51
2cpp	-7.36	-7.36	0.90	0.91
2mcp	-5.55	-5.52	0.89	0.98
3ptb	-8.19	−8.16	0.29	0.29
4hmg	-7.91	-7.85	0.97	1.03

Table 2: Performance statistics for GA0-PS3000: final energy and rmsd from known crystal structure.

Our experiments in this paper have focused on the factors that affect the utility of local search in a hybrid EA. Consequently, we have used the same problem formulation and type of GA that we have applied in previous studies with Autodock [17, 14]. However, there are several ways that these methods could probably be improved. For continuous domains, the scale of mutation is often adapted [18] and recombination operators can be applied that do not impose a coordinate bias to the search [3]. Finally, the boundary constraints and the equality constraint on the quaternion's direction of rotation should be handled explicitly.

Acknowledgements

We thank Mark Land, Art Olson and David Goodsell for many useful discussions concerning drug docking and hybrid EAs. This work was performed in part at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

References

[1] D. E. Clark and D. R. Westhead. Evolutionary algorithms in computer-aided molecular design. *J Computer-Aided Molecular Design*, 10:337–358, 1996.

[2] L. Davis, editor. *Handbook of Genetic Algorithms.* Van Nostrand Reinhold, 1991.

[3] L. J. Eshelman and J. D. Schaffer. Real-coded genetic algorithms and interval schemata. In L. D. Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 187–202. Morgan-Kauffmann, San Mateo, CA, 1993.

[4] R. J. Freund and W. J. Wilson. *Statistical Methods.* Academic Press, 1997.

[5] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Publishing Co., Inc., 1989.

[6] D. S. Goodsell, G. M. Morris, and A. J. Olson. *J Mol Recog*, 9(1), 1996.

[7] D. S. Goodsell and A. J. Olson. Automated docking of substrates to protiens by simulated annealing. *Protiens: Structure, Function and Genetics*, 8:195–202, 1990.

[8] W. E. Hart. *Adaptive Global Optimization with Local Search.* PhD thesis, University of California, San Diego, May 1994. ftp://ftp.cs.sandia.gov/pub/papers/wehart/thesis.ps.gz.

[9] W. E. Hart. SGOPT: A library for stochastic global optimization. (in preparation), 1998.

[10] W. E. Hart, T. E. Kammeyer, and R. K. Belew. The role of development in genetic algorithms. In L. D. Whitley and M. D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 315–332, San Fransico, CA, 1995. Morgan Kaufmann Publishers, Inc.

[11] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *J. Stat. Phys.*, 34(5/6):975–987, 1984.

[12] I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, and T. E. Ferrin. A geometric approach to macromolecular-ligand interactions. *J. Mol. Biol.*, 161:269–288, 1982.

[13] M. Lewis and V. Torczon. Rank ordering and positive bases in pattern search algorithms. *Mathematical Programming*, 1998. (submitted).

[14] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, and A. J. Olson. Automated docking using a lamarkian genetic algorithm and an empirical binding free energy function. *J Comp Chem*, 19(14):1639–1662, 1998.

[15] G. M. Morris, D. S. Goodsell, R. Huey, and A. J. Olson. Distributed automated docking of flexible ligands to proteins: Parallel applications of Autodock 2.4. *J. Comp.-Aid. Mol. Des.*, 10:293–304, 1996.

[16] H. Mühlenbein, M. Schomisch, and J. Born. The parallel genetic algorithm as function optimizer. In R. K. Belew and L. B. Booker, editors, *Proc of the Fourth Intl Conf on Genetic Algorithms*, pages 271–278, San Mateo, CA, 1991. Morgan-Kaufmann.

[17] C. D. Rosin, S. Halliday, W. E. Hart, and R. K. Belew. A comparison of global and local search methods in drug docking. In T. Baeck, editor, *Proc 7th Intl Conf on Genetic Algorithms (ICGA97)*, pages 221–228, San Francisco, CA, 1997. Morgan Kaufmann.

[18] N. Saravanan, D. B. Fogel, and K. M. Nelson. A comparison of methods for self-adaptation in evolutionary algorithms. *BioSystems*, 36:157–166, 1995.

[19] F. Solis and R.-B. Wets. Minimization by random search techniques. *Mathematical Operations Research*, 6:19–30, 1981.

[20] W. Spendley, G. R. Hext, and F. R. Himsworth. Sequential application of simplex designs in optimisation and evolutionary operation. *Technometrics*, 4(4):441–461, Nov 1962.

[21] V. Torczon. On the convergence of pattern search methods. *SIAM J Optimization*, 7(1):1–25, Feb 1997.

[22] A. Törn and A. Žilinskas. *Global Optimization*, volume 350 of *Lecture Notes in Computer Science*. Springer-Verlag, 1989.

[23] P. van Laarhoven and E. Aarts. *Simulated Annealing: Theory and Applications*. Reidel, 1987.

[24] M. Vieth, J. D. Hirst, B. N. Dominy, H. Daigler, and C. L. Brooks III. Assessing search strategies for flexible docking. *J Comp Chem*, 19(14):1623–1631, 1998.