Comparing Evolutionary Programs and Evolutionary Pattern Search Algorithms: A Drug Docking Application

William E. Hart
Sandia National Laboratories
Applied Mathematics Department
P.O. Box 5800, MS 1110
Albuquerque, NM 87185-1110
wehart@cs.sandia.gov
(505) 844-2217

Abstract

Evolutionary programs (EPs) and evolutionary pattern search algorithms (EPSAs) are two general classes of evolutionary methods for optimizing on continuous domains. The relative performance of these methods has been evaluated on standard global optimization test functions, and these results suggest that EPSAs more robustly converge to near-optimal solutions than EPs. In this paper we evaluate the relative performance of EPSAs and EPs on a real-world application: flexible ligand binding in the Autodock docking software. We compare the performance of these methods on a suite of docking test problems. Our results confirm that EPSAs and EPs have comparable performance, and they suggest that EPSAs may be more robust on larger, more complex problems.

1 Introduction

Evolutionary programs (EPs) and evolutionary pattern search algorithms (EPSAs) are two classes of evolutionary algorithms (EAs) that have been specifically developed for solving problems of the form

$$\min_{x \in \mathbf{R}^n} f(x).$$

In particular, both of these classes of EAs include mechanisms for controlling the step length of the mutation operator. EPs employ a self-adaptive mechanism that dynamically adapts the step length along with the search parameters, using an indirect feedback from the evolutionary search to guide the selection of step lengths. EPSAs employ an explicit control mechanism of the step length parameter that uses statistics about the prior success of previous mutation steps about the best point in the population.

EPSAs are distinguished from EPs, however, by the convergence theory that guarantees that they weakly converge to a stationary point with probability one [6, 5, 8]. In particular, this is the only convergence theory for EAs that ensures a convergence to a stationary-point for nonconvex functions. This provides strong evidence that the step length control used by EPSAs will be robust. Further, termination rules have been developed for EPSAs that have been seen to robustly terminate near a stationary point [7].

We have recently evaluated the relative performance of EPs and EPSAs on a test suite of standard global optimization problems [9]. These results confirm that EPs and EPSAs perform a global search of the initial search domain. They also suggest that EPSAs will frequently converge to better solutions than EPs, even when EPs are allowed to run beyond the point when an EPSA terminates its search.

In this paper we consider the relative performance of EPs and EPSAs on a real-world application: molecular docking in Autodock. Computational methods for molecular docking are valuable tools for structure-based drug discovery. Autodock [4, 13] uses a physically detailed model that allows for a fixed receptor site and flexible ligand that is docked into the receptor. 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 [3] using a simulated annealing search method.

More recently, evolutionary algorithms (EAs) have been incorporated into Autodock and applied to standard test problems [10, 12, 14]. In this work, the EAs consistently perform better than simulated annealing. The molecular docking problem solved by Autodock is a challenging global optimization problem, and the EAs appear to perform a better global search across the range of positional, orientational and conformational parameters for flexible ligands.

Two forms of EAs have been used with Autodock: a genetic algorithm [2] and a hybrid EA that uses local search. The hybrid EA applies local search in each iteration to refine points. Rosin et al. [14] and Morris et al. [12] show that this local refinement can significantly improve the performance of the EA. However, the genetic algorithm used in these studies does not adaptively modified the mutation operator's step length. EAs that adapt the step length generally outperform EAs that do not, so these results do not provide a complete justification for the use of an EA hybrid.

In this paper we apply a standard formulation of EP and the formulation of an EPSA recommended by Hart and Hunter [9] to standard Autodock test cases. Our results confirm that EPSAs and EPs have comparable performance, and they suggest that EPSAs may be more robust on larger, more complex problems. Further, our results highlight how the EP's ability to dynamically rescale the search in different directions provides an important search capability that is missing in our formulation of EPSAs. We compare these results to the performance of the hybrid EAs described by Hart et al. [10], and note that the hybrid EAs have much better performance for this application.

2 Autodock

Autodock docks small flexible substrate molecules to large rigid macromolecules like proteins [13]. 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. [12] 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. [18], 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 Background

3.1 Evolutionary Programs

Evolutionary programs (EPs) are a standard paradigm for applying evolutionary methods to continuous optimization problems [1]. EPs are similar to evolutionary strategies [16] in many respects. These EAs generally do not rely on recombination to perform a global search of the search domain. Mutation is typically performed by adding normally distributed random variables to each dimension of an individual, and the standard deviation of these normal deviates is usually modified by a self-adaptive mechanism. This mechanism can be viewed as a separate encoding of the mutation standard deviation along with the search parameters.

Figure 1 shows pseudo-code for a canonical EP that uses self-adaptation. $N(0,1)$ is a normally distributed variable with standard deviation 1, and $N(0,\sigma)$ is a vector of normally distributed random variables with standard deviation σ_i. The function **selection** selects individuals from the previous population (possibly creating a multiset) that are used to perform additional search, and the function **compose** forms the next population using the newly generated points and the previous population. This code uses a log-normal update to σ_i, which Saravanan, Fogel and Nelson [15] confirm is generally preferable to the additive update that has been proposed for EPs. This update uses the constants $\tau = \left(\sqrt{2\sqrt{n}}\right)^{-1}$ and $\tau' = \left(\sqrt{2n}\right)^{-1}$ [16]. The stopping rules used for EPs typically rely on measures of the rate of improvement or population statistics that evaluate whether the population has converged to a single point [1].

3.2 Evolutionary Pattern Search Algorithms (EPSA)

Figure 2 shows pseudo-code for a class of simple EPSAs, which share many of the common features of standard EAs like EP and ES. Our analysis of EPSAs [5, 6, 8] defines requirements on EPSAs that can be used to guarantee a stationary-point convergence. Specifically, we have shown that for a continuously differentiable function the sequence of best solutions in each generation, $\{x_k^*\}$, has the property that

$$P\left(\liminf_{k\to\infty} \| \bigtriangledown f(x_k^*)\| = 0\right) = 1,$$

where $\bigtriangledown f(x)$ is the gradient of f at x. Although this only guarantees convergence to a stationary point or local optimum, experience with direct search methods suggests that EPSAs can be successfully applied to a wide range of global optimization problems (e.g. see [11]). Our previous empirical evaluation of EPSAs [9, 7] indicates that they can perform a nonlocal optimization of the search domain.

To provide this convergence guarantee, mild conditions are placed upon the **selection** and **compose** functions to ensure that (a) the best point in the population has a nonzero chance of being selected in each generation and (b) the best point between the newly generated points and the current population is included in the subsequent populations. The **crossover** function is also restricted to generate a point such that $\text{crossover}(x,y) \in \{x_1,y_1\} \times \{x_2,y_2\} \times \ldots \times \{x_n,y_n\}$, which is consistent with most standard crossover operators (e.g. two-point crossover). The call to $\text{uint}(j)$ uniformly generates an integer from 1 to j. The expansion of steps is controlled by an expansion factor, **exp-factor**, which is greater than or equal to one.

EPSAs differ from self-adaptive EAs like the EP in Figure 1 in that the step length parameter is controlled explicitly. Further, EPSAs use a single step length parameter, while EPs have separate step length parameters for each dimension. The EPSA step length parameter may be expanded if an improving step is generated from a mutation step off of the current best point. Also, the step length may be contracted if all mutation steps about the current best point have worse fitness than the current best point.

The convergence theory requires that the set of mutation offsets in an EPSA, S, form a positive basis of the search domain. The *positive span* of a set of vectors $\{a_1,\ldots,a_r\}$ is the cone $\{a \in \mathbf{R}^n \mid a = c_1a_1 + \ldots + c_ra_r, c_i \geq 0 \quad \forall i\}$. The set $\{a_1,\ldots,a_r\}$ is *positive independent* if none of the a_i's is a positive combination of the others. A *positive basis* is a positive

Given initial step length vectors $\{\sigma_1^0, \ldots, \sigma_N^0\}$
Select an initial population $X_0 = \{x_1^0, \ldots, x_N^0\}$, $x_i^0 \in \mathbf{R}^n$
$x_0^* = \arg\min\{f(x_1^0), \ldots, f(x_N^0)\}$
Repeat $t = 0, 1, \ldots$
    $\hat{X} = \mathrm{selection}(X_t)$
    For $i = 1 : N$
        $\nu = N(0,1)$
        For $j = 1 : n$
            $\sigma_i^{t+1}(j) = \sigma_i^t(j) * \exp(\tau'\nu + \tau N(0,1))$
            $\hat{x}_i(j) = \hat{x}_i(j) + N(0, \sigma_i^{t+1}(j))$
    $X_{t+1} = \mathrm{compose}(X_t, \hat{X})$
    $x_{t+1}^* = \arg\min\{f(x_t^*), f(x_1^{t+1}), \ldots, f(x_N^{t+1})\}$
Until some stopping criterion is satisfied

Figure 1: **A canonical EP or ES using self-adaptation.**

independent set whose positive span is $\mathbf{R}^n$. A positive basis has at least $n+1$ vectors and at most $2n$ vectors. Figure 3 illustrates two sets of mutation offsets. Figure 3a depicts the *coordinate mutation offsets*. This set of offsets uses $S = S_{\text{coord}} = \{e_1, -e_1, \ldots, e_n, -e_n\}$. Hence, the mutation steps are parallel to coordinate axes. S_{coord} contains $2n$ mutation offsets, so it forms a maximal positive basis. Figure 3b depicts a set of mutation offsets that form a minimal positive basis. The $n+1$ mutation offsets are defined by vectors from the centroid of a regular simplex to each of its corners. This set of mutation offsets consists of axes that are separated by an angle of 120 degrees. The regular simplex is an equilateral triangle in two dimensions, a tetrahedron in three dimensions, and so on. In n dimensions *regular simplex mutation offsets* can be derived using the method defined by Spendley, Hext, and Himsworth [17].

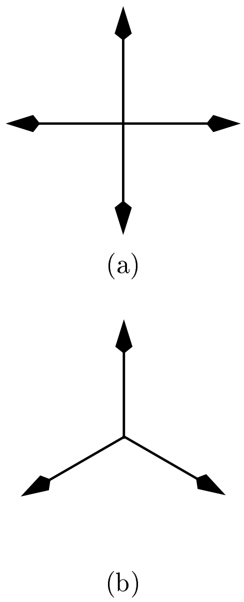

Figure 3: Illustrations of sets of mutation offsets: (a) coordinate mutation offsets and (b) regular simplex mutation offsets.

4 Experimental Comparison

4.1 Methods

We used the EPSA that was recommended by our preliminary experiments with standard global optimization test problems [9]. This EPSA uses the regular simplex mutation offsets and no expansion of the step length is allowed. EPSA was used without crossover and with a two-point crossover operator applied to 50% of the population. For both EP and EPSA tests, we used a population size of 50. The mutation operator is always applied in both EAs. Linear rank selection was used in the EPSA, and the standard competitive selection mechanism was used in the EP.

The docking problems have different ranges in each dimension, but we normalized the search by rescaling each dimension to the range $[0.0, 1.0]$. In this range, the initial step length for EPSAs was set to 0.1. We setup the EP step length parameters to correspond to the step lengths for the EPSA. Specifically, we selected the initial vectors σ_j^0 so that the expected distance of mutation was approximately equal to 0.1 by setting $\sigma_j^0 = 0.1/\sqrt{n}$ (see Hart and Hunter [9] for further details).

The EPSA was terminated when the mutation step length fell below a threshold of 10^{-8}. Because of the stochastic nature of step length updates in the EP,

(1) Given $\Delta_0 \in \mathbf{Q}^{>0}$
(2) Given $S = \{s_1, \ldots, s_m\}$, where $s_i \in \mathbf{Z}^n$ and S forms a positive basis.
(3) Let $\eta = \{0\}^m$
(4) Select an initial population $X_0 = \{x_1^0, \ldots, x_N^0\}$, $x_i^0 \in \mathbf{Q}^n$
(5) $x_0^* = \arg\min\{f(x_1^0), \ldots, f(x_N^0)\}$
(6) Repeat $t = 0, 1, \ldots$
(7) $\quad \overline{X} = \text{selection}(X_t)$
(8) $\quad$ For $i = 1 : N$
(9) $\qquad$ If $(\text{unif}() < \chi)$ then
(10) $\qquad\quad \hat{x}_i = \text{crossover}(\overline{x}_{\text{uint}(N)}, \overline{x}_{\text{uint}(N)})$
(11) $\qquad$ Else
(12) $\qquad\quad \hat{x}_i = \overline{x}_{\text{uint}(N)}$
(13.a) For $i = 1 : N$
(13.b) $\qquad$ If $(\text{unif}() < \mu)$ then
(13.c) $\qquad\quad j = \text{uint}(m)$
(13.d) $\qquad\quad$ If $(\hat{x}_i == x_{t-1}^*)$ $\eta_j = 1$
(13.e) $\qquad\quad \hat{x}_i = \hat{x}_i + \Delta_t \cdot s_j$
(14) $\quad X_{t+1} = \text{compose}(X_t, \hat{X})$
(15) $\quad x_{t+1}^* = \arg\min\{f(x_1^{t+1}), \ldots, f(x_N^{t+1})\}$
(16) $\quad$ If $(f(x_{t+1}^*) < f(x_t^*))$
(17) $\qquad$ If $(\exists s \in S \text{ s.t. } x_{t+1}^* = x_t^* + s)$ $\Delta_{t+1} = \Delta_t * \text{ exp} - \text{factor}$
(18) $\qquad \eta = \{0\}^m$
(19) $\quad$ ElseIf $(|\eta| == m)$
(20) $\qquad \eta = \{0\}^m$
(21) $\qquad \Delta_{t+1} = \Delta_t / 2$
(22) $\quad$ Else
(23) $\qquad \Delta_{t+1} = \Delta_t$
(24) Until $(\Delta_{t+1} < \Delta_{lb})$

Figure 2: **A simple EPSA using multinomial mutation.**

we simply bounded the σ_i values below by $10^{-8}/\sqrt{n}$, which keeps the step length above 10^{-8}. This makes the comparison between the EPSA and the EP fair by not allowing the EP to shrink its step length below the step length of the EPSA. The EP was terminated after 1.5 million function evaluations (which enables a comparison with previous work [14, 12]).

4.2 Experiments

A test suite of six docking problems was used in 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 [13]. In each experiment, 30 trials were done with different random seeds.

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 $[-1,1]^3 \times [-\pi, \pi]$, and each torsion angle lies within $[-\pi, \pi]$; the points in the initial population have each parameter generated randomly in its range.

The parameter bounds were used to create the initial populations of the EAs, but the evolutionary search was not constrained to these regions. A quadratic penalty term is used within Autodock to penalize atoms that lie outside of the coordinate search domain. We renormalized the quarternion parameters to form a unit vector, and all angular coordinates were renormalized to lie within $[-\pi, \pi]$.

5 Results

Figures 4 shows boxplots of the relative rank of the final docking energies for EPSA and EP. Boxplots are a

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 problems.

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.

For each test case, the trials for the hybrid EAs are ranked, and the boxplots show the distribution of ranks within each test case. For EP, we present the final results after 1.5 million function evaluations. We also considered the performance of the EP after the median number of function evaluations required by the EPSA. These results were not substantially different from the the final EP results, which suggests that the EP converges to its final solution about as quickly as the EPSA.

The EPSAs using crossover were generally better than the EPSAs without crossover, which is consistent with our previous results [9]. The median number of function evaluations used by the EPSAs is given in Table 2. While the EPSAs with crossover found better solutions, they took about twice as long to terminate.

	Crossover Rate	
	0.0	0.5
1hvr	141807	246888
1stp	94179	179366
2cpp	40010	69557
2mcp	76294	142126
3ptb	31141	59218
4hmg	152710	260804

Table 2: Median number of function evaluations for the EPSA with different crossover rates.

Neither the EP nor the EPSA have a strict advantage in performance. However, there is a distinction between the performance on the test cases 2cpp and 3ptb and the other four test cases. The EP does somewhat better on 2cpp and 3ptb and the EPSA does somewhat better on the other test cases. The difference between these is that 2cpp and 3ptb are the smallest problems, and they have no torsion angles in the ligand.

Upon inspection, we realized that a plausible explanation for the different relative performance on 2cpp and 3ptb is that the EP can rescale the search for the coordinate position. These problems may be more sensitive to the coordinate position since in the other problems the torsion angle parameters can be refined to reduce the energetic penalty of a poor choice of coordinate position. To evaluate this hypothesis, we took the Autodock clustering information from our experiments with 2cpp and 3ptb, and took the solutions from the lowest-energy cluster. For each problem, we took the range of coordinate positions in these solutions and reran the experiments to force the EP and EPSA to have coordinate positions that lie within this range. The roughly corresponds to fixing the coordinate positions in these experiments; since we do not precisely know the value at the global optima, we used a focused range of values.

For 2cpp and 3ptb, both EPSA and EP have virtually identical performance in these constrained experiments. In all cases both methods found solutions within 0.03 of the best known value. These results support our hypothesis, which suggests that a limitation of EPSAs is their inability to rescale the search in different search directions.

6 Discussion

These results provide the first validation of the utility of EPSAs on a real-world application. The EPSA and EP used in our experiments had comparable performance. Further, the comparison between the EPSA and EP recommends the EPSA for its robustness in

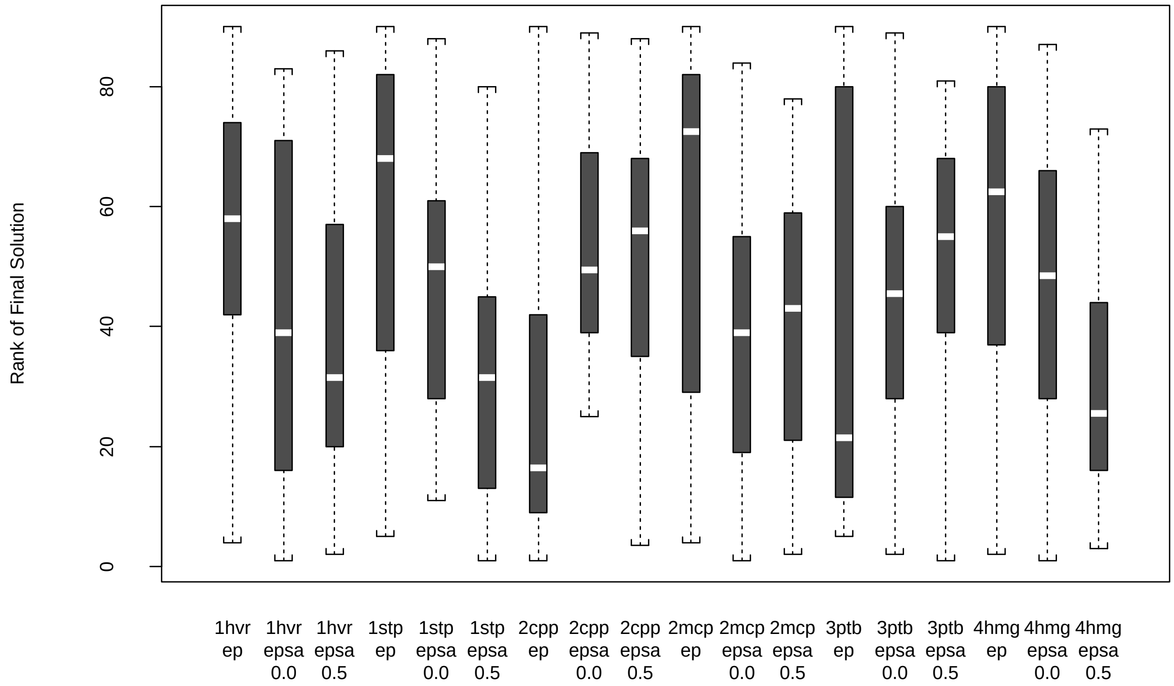

Figure 4: Relative ranks of final docking potentials.

larger, more complex problems. We did not evaluate the gradient of the final point that the EPSA terminated with because the docking potential in Autodock is not everywhere differentiable. However, a more careful analysis is needed to evaluate whether or not the point found by the EPSA is locally optimal.

Although our experiments accounted for the initial step lengths in the EP and EPSAs, there remain a number of important factors that need to be further evaluated. For example, the EP and EPSA used different mechanisms for selecting individuals and composing the next generation of points. These differences could have a substantial impact on the performance of these EAs, but a complete experimental comparison is beyond the scope of this work. Also, we were not able to compare the performance of evolutionary strategies that use the $1/5$ rule to adapt a global step length parameter. Since this rule is similar to the adaptation mechanism used in EPSAs, comparing these two methods could provide insight into the convergence properties of ESs.

Table 3 compares the performance of the EPSA and EP against the hybrid EA used by Hart et al. [10]. It is clear that the hybrid EA finds better results than the EAs in all cases. This confirms that the local search used in these hybrid EAs is a critical part of their success, and not simply that the local search performs a localized step length adaptation.

Acknowledgements

We thank Art Olson, David Goodsell, Chris Rosin and Rik Belew for many useful discussions concerning drug docking. We also thank Keith Hunter for his discussions concerning EPSAs. Finally, we thank Kumar Chellapilla for the use of his EP code in this work. This work was performed 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. B. Fogel. *Evolutionary Computation.* IEEE Press, Piscataway, NJ, 1995.

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

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

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

	Hybrid EA		EPSA 0.0		EPSA 0.5		EP	
	Minimum	Mean	Minimum	Mean	Minimum	Mean	Minimum	Mean
1hvr	-21.42	-21.40	-21.39	-6.33	-21.23	-10.55	-21.09	3.77
1stp	-10.17	-10.15	-8.89	-5.65	-10.06	-6.09	-9.74	-5.17
2cpp	-7.36	-7.36	-7.35	-7.19	-7.36	-7.22	-7.36	-7.26
2mcp	-5.55	-5.52	-5.34	-3.41	-5.32	-3.43	-5.00	-2.64
3ptb	-8.19	-8.16	-8.19	-7.93	-8.19	-8.08	-8.17	-7.09
4hmg	-7.91	-7.85	-7.00	-3.54	-6.97	-4.55	-7.00	-2.66

Table 3: Best and mean final energies for the EPSA, the EP and the hybrid EA reported by Hart et al. [10].

[5] W. E. Hart. A generalized stationary point convergence theory for evolutionary algorithms. In T. Baeck, editor, *Proc 7th Intl Conf on Genetic Algorithms (ICGA97)*, pages 127–134, San Francisco, CA, 1997. Morgan Kaufmann.

[6] W. E. Hart. A stationary point convergence theory for evolutionary algorithms. In R. K. Belew and M. D. Vose, editors, *Foundations of Genetic Algorithms 4*, pages 325–342, San Fransico, CA, 1997. Morgan Kaufmann Publishers, Inc.

[7] W. E. Hart. On the application of evolutionary pattern search algorithms. In *Proc Evolutionary Programming VII*, pages 303–312, New York, 1998. Springer.

[8] W. E. Hart. A convergence analysis of unconstrained and bound constrained evolutionary pattern search algorithms. 1999. (submitted).

[9] W. E. Hart and K. Hunter. A performance analysis of evolutionary pattern search with generalized mutation steps. In *Proc Conf Evolutionary Computation*, 1999. (to appear).

[10] W. E. Hart, C. R. Rosin, R. K. Belew, and G. M. Morris. Improved evolutionary hybrids for flexible ligand docking in autodock. In *Proc Intl Conf on Optimization in Computational Chemistry and Molecular Biology*, 1999. (to appear).

[11] J. Meza and M. L. Martinez. Direct search methods for the molecular conformation problem. *Journal of Computational Chemistry*, 15(6):627–632, 1994.

[12] 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.

[13] 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.

[14] 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.

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

[16] H.-P. Schwefel. *Evolution and Optimum Seeking.* John Wiley & Sons, New York, 1995.

[17] 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.

[18] P. F. W. Stouten, C. Frömmel, H. Nakamura, and C. Sander. *Mol. Simul.*, 10, 1993.