

2 Small-molecule Geometry Optimization and Conformational Search

Ron Wehrens

Abbreviations

EA	Evolutionary Algorithm
GA	Genetic Algorithm
ES	Evolution Strategy
SA	Simulated Annealing
NMR	Nuclear Magnetic Resonance
NOE(SY)	Nuclear Overhauser Effect (SpectroscopY)
RMS	Root-Mean-Square error

2.1 Introduction

The optimization of molecular geometry was one of the very first applications of *evolutionary algorithms* (EAs) in chemistry [1,2]. Once the possibilities of this class of optimization methods were realized, they quickly became very popular in a wide range of chemical application areas (several reviews have included applications of EAs in the optimization of chemical structures; see, e.g., [3–8]). Several reasons can be identified for this popularity. First of all, evolutionary methods have shown good results in problems where other methods have struggled. Second, their principles are easily understood and intuitively appealing. Furthermore, implementation is generally easy, given that several toolboxes are available in the public domain. In the extreme case, one only has to provide an evaluation function (see below) that assesses the quality of a trial solution. In structure optimization, such a function is often provided by molecular mechanics software, but other criteria may be used as well.

Several different forms of EAs are used in practice, but the majority of applications concern *genetic algorithms* (GAs) [9,10]. These are characterized by a population of trial solutions, each represented in a binary way, a selection operator that emphasises the best individuals in a population and crossover and mutation operators to generate the next generation. The German school of *evolution strategies* (ESs) [11] differs from GAs in that crossover is not applied, a real-valued representation rather than a binary one is used, and several search parameters such as the mutation rate are encoded on the trial solutions. In this way, the algorithm takes care of its own configuration. Boundaries between the two classes, however, are blurring; GAs often apply real-coded representations and many different, often problem-dependent, evolutionary operators, whereas ESs sometimes apply crossover-like operators or omit the inclusion of search parameters in the strings. To the knowledge of the author, no applications in molecular structure optimization are known of other forms of EA, such as *evolutionary programming* [12], *genetic programming* [13] or *classifier systems* [14], and these will not be treated further in this chapter.

For an algorithm to qualify as an EA, two criteria must be satisfied: a method should manipulate a population of trial solutions and some form of selection should be used. In practice, a third element is almost always present. This is that new trial solutions are created by combination of old solutions (the crossover operator, used in many different forms).

This chapter will not treat technical matters in great depth; it is assumed that readers are familiar with the basic concepts of EAs (see Chapter 1). Rather, attention will be focussed on the differences between EAs and other structure optimization methods, and strengths and weaknesses of each are identified. In the next section, some fundamental qualities of EAs will be discussed that are important in their application, especially in the field of structure optimization. Next, a short literature review will be given of the application of EAs in conformational search and geometry optimization, concentrating on the evaluation functions used, the representation of molecular structure and the types of molecule under study. Separate sections will be devoted to those applications where comparisons are made between EAs and other optimization methods.

2.2 Evolutionary Algorithms

As stated in the introduction, one of the characteristic features of EAs is that a population of trial solutions is maintained. This is a marked difference from other individual-based optimization methods such as simulated annealing (SA) [15], and has several important consequences. In many cases, population-based search behaviour will be markedly different from that of individual-based methods and, as such, the two classes complement each other. Although to date no single optimization method has consistently outperformed all other methods for a range of optimization problems (the ‘No Free Lunch’ theorem [16]), it is possible that for a very specific application there is a clear preference for either the population- or individual-based search. In this section, we highlight those aspects of population-based search that are fundamentally different from individual-based search and that therefore may determine which of these is most suitable for a specific application.

2.2.1 Diversity

Use of a population makes it possible to encourage a better coverage of the search space. Imagine a perfect single-solution optimization method that always finds the globally optimal molecular conformation. In Figure 2.1, this would mean that solution A would be found every time. However, the structures corresponding to some of the local optima B–E may also be populated at room temperature, especially if the energy wells are broad. With a single-solution method, only those conformations encountered during the journey from the starting point to the global optimum will be found. A way to find more would be to restart the search from another starting position, but there is no guarantee that another route will be followed. This kind of search behaviour is therefore also called ‘linear’.

With population-based search methods, there are ways around this problem. The

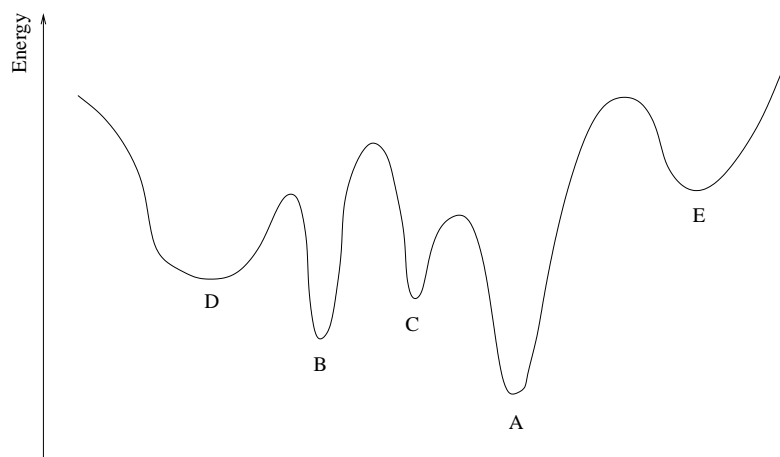


Figure 2.1: A one-dimensional energy surface. Not only the global optimum A may be important, but B-E may also be populated at room temperature.

ability to determine the similarity between several trial solutions opens up possible ways to promote diversity. Special operators such as forced mutations can be employed to ensure that the population does not converge completely to one single optimum. An even more rigorous approach is to maintain not one population but several subpopulations, which are allowed to evolve individually but for an occasional exchange of genetic material. Reports in the literature indicate that this approach not only leads to several good solutions (B-E in Figure 2.1) but also may speed up the route to the global optimum.

2.2.2 Creation of New Solutions

With individual-based search methods, one trial solution is the starting point for the generation of a new solution. The new solution may be generated by a random perturbation of the old solution, or by application of a modifying algorithm. The new solution may or may not replace the old solution as a starting point for the next iteration. This type of search behaviour is illustrated in the left plot in Figure 2.2; it follows a linear trajectory through search space. The consequence of such a strategy is that two consecutive solutions are probably very much alike, whereas two solutions that are far apart in the solution sequence are less likely to be similar. Unless special attention is paid to this problem, this behaviour may make it difficult to overcome large energy barriers (in the case of energy minimization of a molecular structure).

In the case of EAs, the generation of new structures, especially in the case where crossover is used, does not suffer from this problem. A trial solution in an offspring population may differ quite substantially from both its parents, as indicated in the right plot in Figure 2.2. This enables the algorithm to take large steps in the solution space, and the linear behaviour of the individual-based search is not observed. The other side of the coin is that this makes it difficult to obtain a precise estimate of the location of the optimum. This is also clearly visible in Figure 2.2: whereas the linear method goes

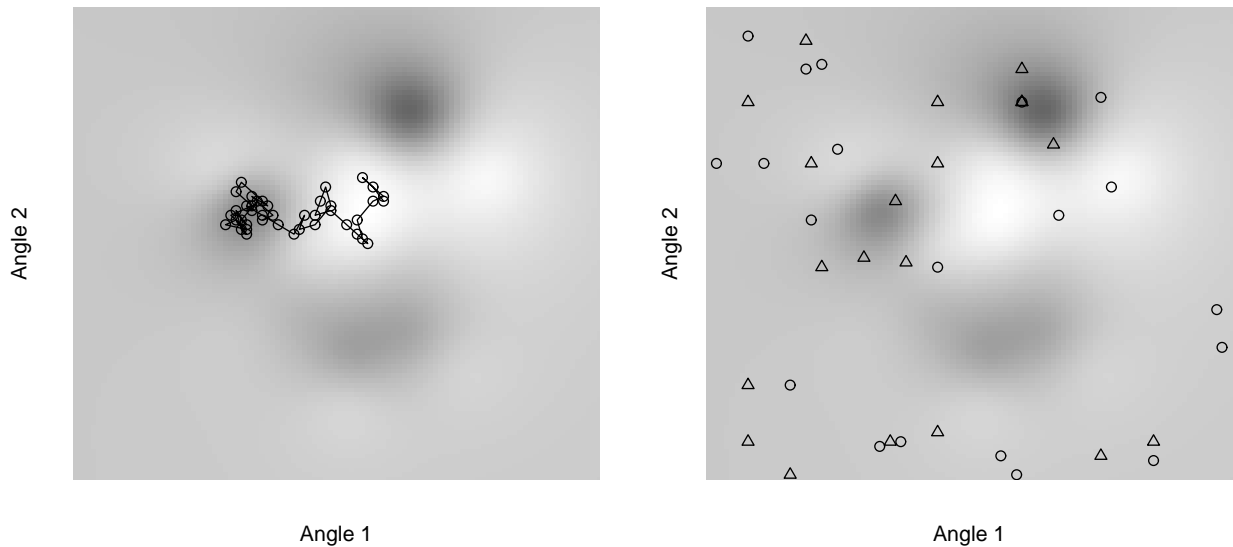


Figure 2.2: Linear versus non-linear search behaviour. The left plot shows a trajectory through search space of an individual-based method; the right plot shows four generations of a GA application in the same search space (in both plots, forty evaluations were performed). Circles indicate members of the zeroth (random) generation; triangles indicate members of the first generation. The darker the colour in the plot, the more negative the energy.

straight to one of the local optima, the GA seems to perform an almost random search. This example is of course very simplistic and EAs are much more suited for searching in high-dimensional spaces. To remedy the poor search precision, the final solution(s) of EAs are often further optimized by local optimization methods.

2.2.3 Constraint Satisfaction

In some optimization problems, including structure optimization, the overall quality of a trial solution may be quite good, but because some constraint is violated, the solution is not a valid one. EAs offer several flexible ways of dealing with this situation. One is to modify the solution so that the violation is removed. This so-called ‘repair’ strategy is also applicable in individual-based optimization methods, but has the disadvantage that it is slow, because of the large number of calculations needed to identify and remove the violation. In some cases, the optimization is also forced in the wrong direction because of a repair strategy. An example is depicted in Figure 2.3, where a grey area in a Ramachandran-like plot indicates a ‘no-go’ area, and the optimum is located at the other side. Individual-based methods employing repair strategies will not be able to cross the grey area and are forced to go around it.

Population-based methods employing the repair strategy are also described in liter-

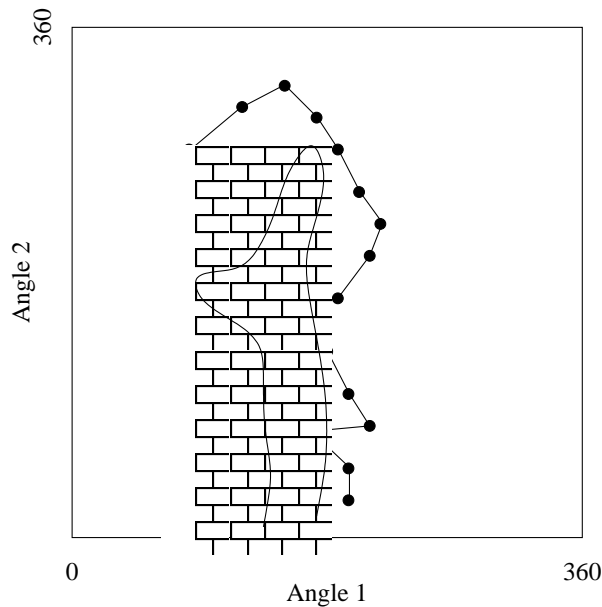


Figure 2.3: Hypothetical Ramachandran-type plot, indicating a forbidden combination of two torsion angles. Individual-based methods are forced to go around the ‘no-go’ area, whereas population-based methods can more or less ignore it. In the figure, three repair operations are necessary.

ature, but there are other strategies available. Because of the non-linear search characteristics mentioned in the previous paragraph, there is no need to go around the forbidden area. Solutions violating the constraints can be treated just like other solutions that are bad, for instance by penalizing them or removing them from the population altogether. An advantage of the penalization strategy is that the ‘genes’ of such a solution are still available for reproduction (although with a smaller probability), so that good elements of this solution may still propagate to the next generation. Both penalization and lethalization strategies are much faster than the repair method, and have been shown to give good results [17].

2.3 Technical Aspects of Method Comparisons

The comparison of several global optimization problems for a given application seems to be a simple task. One performs several optimizations with all the methods and finally picks the one that gives the best result. In the context of structure optimization, this may correspond to the structure with the lowest energy, as calculated by a force field method. However, there are several difficulties with this approach. First of all, one should be sure that each optimization method is properly tuned. Because of the large number of adjustable parameters in EA optimization (and in many other optimization methods as well), this is not a trivial task. Often, a one-at-a-time approach is used, starting from

standard settings, as published in the scientific literature. There is a very real danger that this will lead to suboptimally configured methods, and so more thorough methods utilising experimental design theory have been proposed [18,19]. A full optimization of the search settings may be costly in terms of computing power, and is therefore not often performed. In comparisons in the literature, one should always be aware of the fact that one of the methods employed may have received more attention (e.g., is a newly developed, and hence optimized, method), while others are applied without much tinkering. Comparisons with published results are only possible if exactly the same evaluation function is used in all methods.

Another point is that, especially with stochastic optimization methods such as SA and EAs, results of repeated runs may lead to quite different answers. This variability in the results must be taken into account when comparing optimization methods. One way to do this is to report several values for each optimization (consisting of a number of repeated runs), such as the best solution found, and the quality of the mean and worst final solutions.

These considerations are also important when fine-tuning a search algorithm. With EAs, this can be a difficult task, given the large freedom of the experimenter in choosing operators and search settings. As already stated, most measures of performance focus on the quality of the best solution found by the search method, or a mean value over a number of good solutions. Clearly, this is a one-sided view of the quality of an optimization method, and additional criteria have been suggested. These include measures of diversity during the optimization [20–22], the percentage of cases in which the global minimum was identified [23], the number of evaluations or the CPU time needed to find the optimum [18, 24–26] and the number of different structures found [21]. Exactly what is expected of the optimization method should be reflected in the criteria used to evaluate performance. In the context of structure optimization, a set of four quality criteria has been proposed recently [27]. These concentrate on coverage of the search space, coverage of the solution space and the reproducibility of these quantities in repeated runs. The application of these criteria in the fine-tuning of GA settings has also been described [19]. Because the criteria do not explicitly use values from the evaluation function, it is possible to fine-tune the evaluation function itself. This is important in cases where a weighted sum of several terms is used in evaluating trial solutions (see, e.g., [18,28]).

2.4 Traditional Methods for Structure Optimization

Traditional methods of structure optimization [29] fall in one of two categories, depending on the data that are available. If experimental distance constraints are available, for instance from NMR spectroscopy, distance geometry methods [30,31] can be used. The problem is then to convert these incomplete distance constraints to a complete definition of a molecular conformation in Cartesian coordinates. The distance constraints are formulated as pairs of upper and lower bounds; a constraint is said to be satisfied if the corresponding distance in a molecular structure is between these bounds.

The most common algorithm for distance geometry uses an eigenvalue decomposition to transform a distance matrix into Cartesian space, and a subsequent optimization step is performed to obtain the final coordinates. Often, conjugate gradient minimization [32] is used for this. The distance matrix that is transformed is a random sample of distances between the upper and lower bounds. After this so-called embedding step, a smoothing step may be performed (metrization) after each new distance is picked, to ensure that the triangle inequality holds. Alternatively, an exhaustive search method can be used to manipulate a structure in torsion angle space so that distance constraints are satisfied.

The second class of traditional structure optimization methods aims at finding molecular conformations with a minimal energy. Force fields [33, 34] are used to calculate the energy of a molecular conformation, and an optimization algorithm directs the search into a minimum. Several optimization methods are available: numerical methods [32], Monte Carlo methods and simulated annealing [15], random search and systematic search. To prevent the optimization method from becoming trapped in a local minimum, multiple runs are often performed.

Besides these general optimization methods, some chemistry-specific methods can be used to sample conformational space. In the popular molecular dynamics method [35], each atom in a trial structure is given an initial velocity, and Newtonian equations are used to calculate the structure at a specific instant in time. Although this is an appealing method conceptually, it has several disadvantages: it requires significant computer resources, and is easily trapped in a local optimum. Therefore, its primary use is to study the behaviour of conformations in the vicinity of an optimum.

Other examples of structure optimization methods are directed tweak [36] and direct search methods [37]. The latter is an intelligent adaptive grid search that may be very fast because no derivatives need to be calculated. A set of trial conformations is evaluated and the worst of these is replaced by a new conformation. The replacement is done by a series of geometric operations in multi-dimensional optimization space. The method is closely related to the simplex optimization method of Nelder and Mead [38].

2.5 Evolutionary Methods for Structure Optimization

EAs (and in particular GAs) nowadays are very popular methods for finding the optimal geometry of a molecular structure with respect either to agreement with experimental data, such as NMR-derived distance constraints, or internal energy. These two types of problem form the bulk of all EA applications in this area, and they will be treated in more detail in the next paragraphs. Several other applications also deserve a mention. Structure optimization with respect to similarity to an active site (e.g., [39, 40]) belongs to the domain of molecular docking and will be treated in Chapter 3. The reverse problem is tackled in [41], where a small set of diverse conformers is sought. The fitness of a solution is given by a measure of the difference with all other solutions in the population. The algorithm is initiated from a 3-D structure, and a population is generated by applying random mutations to this structure.

The optimization of clusters of atoms and molecules with evolutionary algorithms is also an active area of research (see, e.g., [23, 42, 43]). In [44], a nested GA is described that can calculate cluster geometries of flexible molecules, where the inner GA loop is used to optimize the structure of each molecule separately. These applications will not be treated further in this chapter, nor will structure optimization as used in the alignment of flexible molecules [45, 46]. These problems are very much related to the applications described in this chapter, however, since only the evaluation function is different. In almost all cases, torsion angles (internal coordinates) are used as the molecular representation, assuming standard values for bond angles and bond lengths. This has the advantage that only a few parameters are needed to describe relatively complex structures. However, one important disadvantage of this representation is the so-called ‘leverage’ effect: a small change in one torsion angle may lead to a drastic change in the overall conformation of the molecule. In general, such a representation is found to have difficulties. Note that this effect is independent of whether real or binary coding is used.

2.5.1 Satisfying Constraints from Experiments

One of the first applications of EAs in structure optimization was described by Lucasius et al. [2, 47]. They optimized the structure of a DNA dinucleotide using a standard GA. The method requires an experimental NOESY NMR spectrum, and the chemical shift information for each proton in the structure. Translation of the dihedral angles to a three-dimensional structure then makes it possible to calculate a theoretical NOE spectrum, which can be compared to the experimental one. The evaluation function in this case is a root-mean-square (RMS) difference between the intensities of crosspeaks in the theoretical and experimental spectra.

A different approach is taken in [48], where distance constraints derived from NMR spectroscopy are used in the evaluation function. A GA, called DG Ω , was used as a front-end for the distance geometry program DGII [49]. The upper and lower bounds for a set of 58 distance restraints were coded in the strings. Each member of the population was the starting point for a distance geometry calculation, and the fitness of this member was given by the number of distance constraint violations. It was shown that for cyclosporin A, a cyclic undecapeptide notorious for its difficult sampling properties, the GA led to a significant improvement in the sampling behaviour of the distance geometry algorithm.

In other applications, the agreement with experimentally obtained distance constraints is used as a fitness function for oligopeptides [50], RNA stem-loop structures [28] and DNA oligonucleotides [18, 47, 51]. Pearlman [52] uses a genetic algorithm to derive an ensemble of structures that best fits NMR data. The resulting weights indicate which conformers are important. In many of these applications, fitness penalties are given for structures with van der Waals overlap between atoms. Other extensions include the use of stereochemical constraints or constraints on the conformation of a substructure. The advantage of such an evaluation function is that it can be calculated quite quickly, much faster than a complete distance geometry optimization or the calculation of a theoretical NMR spectrum.

2.5.1.1 Comparisons with Other Methods

Several comparisons have been made between distance geometry programs and GA approaches. In [53], a GA for the optimization of torsion angles is compared to the DGII distance geometry package [49] for a modified thymine dimer. It was found that DGII was more successful in satisfying the distance constraints in the data. However, this came at the cost of a much larger variability in bond angles, which in some structures assumed rather extreme values. The GA identified a much more tight set of structures because only torsion angles could be modified during the search.

This comparison was extended to include the DG Ω program described in [48], and more or less similar results were obtained [54]. Again, the classical GA yielded the poorest sampling of the conformation space, albeit that all generated structures had a good covalent geometry. DG Ω performed slightly better than DGII, but required much more computing time.

2.5.2 Energy Minimization

An even more popular use of EAs in the optimization of molecular conformation is in the energy minimization of a molecular structure. Here, an appropriate force field calculation is used as evaluation function. In most cases, only the most important energy terms are taken into account to speed up calculations. Many examples can be found of the structural optimization of small organic molecules [20–22, 24, 55–58], small peptides and peptide analogues [59–62], and proteins [1, 25, 63–65].

Again, the most popular way to represent molecular structure is to use internal coordinates (torsion angles). Tufféry et al. [1, 25] describe an evolutionary algorithm that optimizes the structure of protein side chains by selecting rotamers from a predefined set, where the chance of selecting a specific rotamer is determined by its probability of occurrence, determined beforehand. In most cases, standard genetic operators are used, but the application of specialized operators sometimes leads to improvements. In [56], a graph crossover, where two subgraphs are combined (and repaired if necessary), is used, resulting in an increased robustness of the algorithm. Jin et al. compare three variants of their GAP program, differing in ways to enforce diversity during the optimization [22]. It was concluded that the different crossover operators did not influence the sampling characteristics very much, probably because the crossover operator is only of minor importance at the end of the search, when the diversity in the population has decreased.

Parallellization using island or migration models is reported to improve results, mainly by making it easier for the algorithm to maintain a diverse population [51, 56, 62, 63]. Niching, where the population is divided into subsets of more or less similar individuals, is also used for this purpose. A popular niching technique is sharing, where the fitness of individuals in the same niche is lowered when the niche is overpopulated. Measuring the degree of similarity in the population (also called convergence) is not always straightforward. Torsion angles of -179 and 179 degrees are of course quite close, so simply looking at differences or variances may lead to wrong conclusions. In [21], a simple criterion is defined

for the measure of similarity in a population defined by strings of torsion angles. Each angle is represented by a point on the unit circle, and if the angles are evenly distributed, the mean of all angles will lie at the origin. A larger distance from the mean to the origin indicates a smaller diversity. It should be noted, however, that even a small deviation in torsion angles can lead to large differences in overall three-dimensional structure, so that the measure is useful for signalling convergence in the population. Cluster analysis is sometimes used to determine the presence of similar conformations [1, 27, 57, 60].

Although in almost all applications the final solutions of an EA are further optimized by a local optimization method (typically steepest descent), it is also possible to perform a local optimization in each evaluation during the EA run [1, 24]. Although usually only a few steps are performed, run times will increase significantly, and for small molecules it usually is unnecessary.

2.5.2.1 Comparison with Other Methods

Many papers compare the performances of EAs with other search methods such as simulated annealing, direct search methods and random search (see, e.g., [26, 37, 66, 67]). The general conclusion is that EAs are consistently among the best performing general search algorithms, in many cases performing as well as or only slightly worse than optimization methods specifically designed for the problem at hand. As noted by several other authors, Judson states that GAs are particularly useful for quickly producing a family of low-energy conformers, but are less successful in fine-tuning these towards the exact global optimum [66]. The post-EA steepest descent optimization that is performed in most applications is the result of this realization. Because of the limited accuracy of energy calculations this disadvantage is not too important.

In the context of 3D database searching, Clark et al. found that GAs were better than distance geometry, systematic search and random search, and both GAs and directed tweak methods performed well enough to be useful in practical applications [67], a conclusion that is also reached in [26].

2.6 Discussion

The many successful applications of EAs, and in particular GAs, have made EAs the *de facto* reference method in the optimization of small- to medium-sized molecules. With larger structures, typically containing hundreds of torsion angles, results have been less encouraging, and it may be necessary to apply EAs to smaller substructures separately before analysing the overall geometry (see Chapter 11 for more details). Apart from the good results, in that usually very low energies are obtained or many of the experimental constraints are satisfied, EAs have the significant advantage that they provide a family of solutions. In some cases, it is shown that they are not always the most efficient search method in terms of number of evaluations, but with the dramatic increase in computing power of the last few years (and there is little reason to believe that this development is

slowing down), efficiency may not be the most important aspect anymore. Attention will probably shift to measures of completeness (are all relevant minima identified?), reliability and reproducibility.

As already stated, application of EAs is relatively straightforward, once an evaluation function is available. The most straightforward of these is a force field program, but more often custom-built evaluation functions are written. In most cases, the value that is returned is only required to give a relative quality measure of the trial solution so, for instance, large parts of expensive energy calculations can be omitted. Configuration of EAs may be a problem, and sometimes large performance drops can be observed because of an inadequate choice of operators or search settings. Although standard settings seem to be used widely, small differences in implementation may require an expensive meta-optimization. Table 2.1 summarizes some of the applications mentioned in the text.

Table 2.1: Summary of applications mentioned in the text.

Compound class	Evaluation	Representation	Reference
Org.	Energy	T	[20, 21, 24, 55, 56, 58]
Prot.	Energy	P	[1]
	Energy	T	[22, 57, 60, 62, 63, 65]
	Constraints	T	[50, 59, 61]
	Constraints	O	[48]
Nucl. Ac.	Constraints	T	[2, 28, 47, 51]
	Constraints	O	[18]

Compound classes: organic molecules (Org.), proteins and peptides (Prot.) and DNA and RNA (Nucl. Ac.)

Representation: T: torsion angles, P: predefined set of partial conformations, O: other.

The most critical choice for the success of an EA, however, is probably the representation. The most widely used representation, and also the most efficient one, consists of a series of torsion angles (see also Table 2.1). This leads to standard values for bond lengths and bond angles; usually, these are optimized in the subsequent local optimization. The biggest disadvantage of a representation by torsion angles is that a small change in one angle may lead to a large change in the value of the evaluation function. This leads to an EA ‘search landscape’ with very sharp peaks, which in test functions is observed to hamper the performance of the search method significantly. On the other hand, the deviation in one torsion angle may be compensated for by another angle, so that two different strings may code for almost the same three-dimensional structure. At the moment, however, in most applications there is no realistic alternative.

The flexible nature of EAs and many other global search techniques has lead to many hybrid methods, where elements of methods from different classes are combined. Examples in other fields include combinations of evolutionary methods with local optimization methods, simulated annealing and tabu search [68] (see Chapter 12 for a further discussion).

Two examples of methods having some characteristics of EAs can be found in [69, 70]. In [70], deep local minima are identified by combining torsion angles from a small set of other local minima. In [69], what is essentially a tree-search is performed, where the descendants of each node are generated by crossover like mechanisms. Again, torsion angles are taken from a pool of previously found local minima.

2.7 Conclusion

Evolutionary algorithms of many flavours have found wide application in the conformational search of small- to medium-sized molecules. Their success has been remarkable, especially since they are general problem solvers, not specifically designed for structure optimization problems. Moreover, the basic algorithm is very simple. Although there are several issues to be addressed, most notably the representation of molecular structure, EAs are likely to continue to play a major role in the future.

Bibliography

- [1] P. Tufféry, C. Etchebest, S. Hazout, R. Lavery, A New Approach to the Rapid Determination of Protein Side Chain Conformations, *J. Biomol. Struct. Dyn.* **1991**, 8, 1267–1289.
- [2] C.B. Lucasius, M.J.J. Blommers, L.M.C. Buydens, G. Kateman, A Genetic Algorithm for Conformational Analysis of DNA, In L. Davis, (Ed.), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, **1991**, pp. 251–281.
- [3] J. Devillers (Ed.), *Genetic Algorithms in Molecular Modeling*, Academic Press, New York, **1996**.
- [4] P. Willett, Genetic Algorithms in Molecular Recognition and Design, *TIBTECH* **1995**, 13, 516–521.
- [5] G. Bohm, New Approaches in Molecular Structure Prediction, *Biophys. Chem.* **1996**, 59, 1–32.
- [6] D.E. Clark, D.R. Westhead, Evolutionary Algorithms in Computer-Aided Molecular Design, *J. Comput.-Aided Mol. Des.* **1996**, 10, 337–358.
- [7] R.S. Judson, Genetic Algorithms and their Use in Chemistry, In K.B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 10*, VCH, New York, **1997**, pp. 1–73.
- [8] G. Jones, Genetic and Evolutionary Algorithms, In P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, P. R. Schreiner (Eds.), *Encyclopedia of Computational Chemistry*, John Wiley and Sons, Chichester, UK, **1998**, Volume 2, pp. 1127-1136.
- [9] J.H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, **1992**.
- [10] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, **1989**.
- [11] H.P. Schwefel, *Numerical Optimization of Computer Models*, Wiley, Chichester, UK, **1981**.

- [12] D.B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ, **1995**.
- [13] J.R. Koza, *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, **1992**.
- [14] L.B. Booker, D.E. Goldberg, J.H. Holland, Classifier Systems and Genetic Algorithms, *Artif. Intell.* **1989**, *40*, 235–282.
- [15] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by Simulated Annealing, *Science* **1983**, *220*, 671–680.
- [16] D.H. Wolpert, W.G. Macready, No Free Lunch Theorems for Optimization, *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82.
- [17] A.H.C. van Kampen, C.S. Strom, L.M.C. Buydens, Lethalization, Penalty and Repair Functions for Constraint Handling in Genetic Algorithm Methodology, *Chemom. Intell. Lab. Syst.* **1996**, *34*, 55.
- [18] A.H.C. van Kampen, L.M.C. Buydens, The Ineffectiveness of Recombination in a Genetic Algorithm for the Structure Elucidation of a Heptapeptide in Torsion Angle Space. A Comparison to Simulated Annealing, *Chemom. Intell. Lab. Syst.* **1997**, *36*, 141–152.
- [19] R. Wehrens, E. Pretsch, L.M.C. Buydens, The Quality of Optimization by Genetic Algorithms, *Anal. Chim. Acta* **1999**, *388*, 265–271.
- [20] D.B. McGarrah, R.S. Judson, Analysis of the Genetic Algorithm Method of Molecular Conformation Determination, *J. Comput. Chem.* **1993**, *14*, 1385–1395.
- [21] N. Nair, J.M. Goodman, Genetic Algorithms in Conformational Analysis, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 317–320.
- [22] A.Y. Jin, F. Y. Leung, D. F. Weaver, Three Variations of Genetic Algorithm for Searching Biomolecular Conformation Space: Comparison of GAP 1.0, 2.0, and 3.0, *J. Comput. Chem.* **1999**, *20*, 1329–1342.
- [23] J.A. Niesse, H.R. Mayne, Global Optimization of Atomic and Molecular Clusters using the Space-Fixed Modified Genetic Algorithm Method, *J. Comput. Chem.* **1997**, *18*, 1233–1244.
- [24] R.S. Judson, E.P. Jaeger, A.M. Treasurywala, M.L. Peterson, Conformational Searching Methods for Small Molecules. II. Genetic Algorithm Approach, *J. Comput. Chem.* **1993**, *14*, 1407–1414.
- [25] P. Tufféry, C. Etchebest, S. Hazout, R. Lavery, A Critical Comparison of Search Algorithms Applied to the Optimization of Protein Side-Chain Conformations, *J. Comput. Chem.* **1993**, *14*, 790–798.

- [26] J.C. Meza, R.S. Judson, T.R. Faulkner, A.M. Treasurywala, A Comparison of a Direct Search Method and a Genetic Algorithm for Conformational Searching, *J. Comput. Chem.* **1996**, *17*, 1142–1151.
- [27] R. Wehrens, E. Pretsch, L.M.C. Buydens, Quality Criteria of Genetic Algorithms for Structure Optimization, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 151–157.
- [28] H. Ogata, Y. Akiyama, M. Kanehisa, A Genetic Algorithm based Molecular Modeling Technique for RNA Stem-Loop Structures, *Nucl. Acids Res.* **1995**, *23*, 419–426.
- [29] A.R. Leach, A Survey of Methods for Searching the Conformational Space of Small and Medium-Sized Molecules, In K.B. Lipkowitz, D.B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 2*, VCH, New York, **1991**, pp. 1–55.
- [30] G.M. Crippen, *Distance Geometry and Conformational Calculations*, Research Studies Press, Chichester, UK, **1981**.
- [31] J.M. Blaney, J.S. Dixon, Distance Geometry in Molecular Modeling, In K.B. Lipkowitz, D.B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 5*, VCH, New York, **1994**, pp. 299–335.
- [32] T. Schlick, Geometry Optimization, In P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, P. R. Schreiner (Eds.), *Encyclopedia of Computational Chemistry*, John Wiley and Sons, Chichester, UK, **1998**, pp. 1136–1157.
- [33] J.P. Bowen, N.L. Allinger, Molecular Mechanics: the Art and Science of Parameterization, In K.B. Lipkowitz, D.B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 2*, VCH, New York, **1991**, pp. 81–97.
- [34] I. Pettersson, T. Liljefors, Molecular Mechanics Calculated Conformational Energies of Organic Molecules: a Comparison of Force Fields, In K.B. Lipkowitz, D.B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 9*, VCH, New York, **1996**, pp. 167–189.
- [35] W.F. van Gunsteren, H.J.C. Berendsen, Computer Simulation of Molecular Dynamics: Methodology, Applications, and Perspectives in Chemistry, *Angew. Chem., Int. Ed. Engl.* **1990**, *29*, 992.
- [36] T. Hurst, Flexible 3D Searching: the Directed Tweak Technique, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 190–196.
- [37] J.C. Meza, M.L. Martinez, Direct Search Methods for the Molecular Conformation Problem, *J. Comput. Chem.* **1994**, *15*, 627–632.
- [38] J.A. Nelder, R. Mead, A Simplex Method for Function Optimization, *Computer J.* **1965**, *7*, 308–313.

- [39] R.S. Judson, Y.T. Tan, E. Mori, C. Melius, E.P. Jaeger, A.M. Treasurywala, A. Mathiowetz, Docking Flexible Molecules: a Case Study of Three Proteins, *J. Comput. Chem.* **1995**, *16*, 1405–1419.
- [40] C.M. Oshiro, I.D. Kuntz, J.S. Dixon, Flexible Ligand Docking using a Genetic Algorithm, *J. Comput.-Aided Mol. Des.* **1995**, *9*, 113–130.
- [41] O. Mekenyan, D. Dimitrov, N. Nikolova, S. Karabunarliev, Conformational Coverage by a Genetic Algorithm, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, in press.
- [42] D.M. Deaven, N. Tit, J.R. Morris, K.M. Ho, Structural Optimization of Lennard-Jones Clusters by a Genetic Algorithm, *Chem. Phys. Lett.* **1996**, *256*, 195–200.
- [43] W.J. Pullan, Structure Prediction of Benzene Clusters using a Genetic Algorithm, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 1189–1193.
- [44] J. Mestres, G.E. Scuseria, Genetic Algorithms: a Robust Scheme for Geometry Optimizations and Global Minimum Structure Problems, *J. Comput. Chem.* **1995**, *16*, 729–742.
- [45] G. Jones, P. Willett, R.C. Glen, A Genetic Algorithm for Flexible Molecular Overlay and Pharmacophore Elucidation, *J. Comput.-Aided Mol. Des.* **1995**, *9*, 532–549.
- [46] D.A. Thorner, D.J. Wild, P. Willett, P.M. Wright, Similarity Searching in Files of Three-Dimensional Chemical Structures: Flexible Field-Based Searching of Molecular Electrostatic Potentials, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 900–908.
- [47] M.J.J. Blommers, C.B. Lucasius, G. Kateman, R. Kaptein, Conformational Analysis of a Dinucleotide Photodimer with the Aid of the Genetic Algorithm, *Biopolymers* **1992**, *32*, 45–52.
- [48] A.H.C. van Kampen, L.M.C. Buydens, C.B. Lucasius, M.J.J. Blommers, Optimization of Metric Matrix Embedding by Genetic Algorithms, *J. Biomol. NMR* **1996**, *7*, 214.
- [49] T.F. Havel, An Evaluation of Computational Strategies for Use in the Computational Determination of Protein Structure from Distance Constraints Obtained by NMR, *Prog. Biophys. Mol. Biol.* **1991**, *56*, 43–78.
- [50] P.N. Sanderson, R.C. Glen, A.W.R. Payne, B.D. Hudson, C. Heide, G.E. Tranter, P.M. Doyle, C.J. Harris, Characterization of the Solution Conformation of a Cyclic RGD Peptide Analogue by NMR Spectroscopy Allied with a Genetic Algorithm Approach and Constrained Molecular Dynamics, *Int. J. Peptide Protein Res.* **1994**, *43*, 588–596.
- [51] M.L.M. Beckers, E.P.P.A. Derks, W.J. Melssen, L.M.C. Buydens, Parallel Processing of Chemical Information in a Local Area. Part III. Using Genetic Algorithms for Conformational Analysis of Biomacromolecules, *Comput. Chem.* **1996**, *20*, 449–457.

- [52] D.A. Pearlman, FINGAR: a New Genetic Algorithm-Based Method for Fitting NMR Data, *J. Biomol. NMR* **1996**, *8*, 49–66.
- [53] M.L.M. Beckers, L. Buydens, J. Pikkemaat, C. Altona, Application of a Genetic Algorithm in the Conformational Analysis of Methylene-acetal-linked Thymine Dimers in DNA: Comparison with Distance Geometry Calculations, *J. Biomol. NMR* **1997**, *9*, 25–34.
- [54] A.H.C. van Kampen, M.L.M. Beckers, L.M.C. Buydens, A Comparative Study of the DG-OMEGA, DGII and Genetic Algorithm Torsion Angle Optimization Methods for the Structure Elucidation of a Methylene-acetal Linked Thymine Dinucleotide, *Comput. Chem.* **1997**, *21*, 281–297.
- [55] T. Brodmeier, E. Pretsch, Application of Genetic Algorithms in Molecular Modelling, *J. Comput. Chem.* **1994**, *15*, 588–595.
- [56] S. Beiersdörfer, J. Schmitt, M. Sauer, A. Schulz, S. Siebert, J. Hesser, R. Männer, J. Wolfrum, Finding the Conformation of Organic Molecules with Genetic Algorithms, In H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), *Lect. Notes in Comp. Sci. 1141*, Springer, Heidelberg, **1996**, pp. 972–981.
- [57] A.Y. Jin, F.Y. Leung, D.F. Weaver, Development of a Novel Genetic Algorithm Search Method (GAP1.0) for Exploring Peptide Conformational Space, *J. Comput. Chem.* **1997**, *18*, 1971–1984.
- [58] M. Keser, S.I. Stupp, A Genetic Algorithm for Conformational Search of Organic Molecules: Implications for Materials Chemistry, *Comput. Chem.* **1998**, *22*, 345–351.
- [59] T. Dandekar, P. Argos, Folding the Main Chain of Small Proteins with the Genetic Algorithm, *J. Mol. Biol.* **1994**, *236*, 844–861.
- [60] F. Herrmann, S. Suhai, Energy Minimization of Peptide Analogues using Genetic Algorithms, *J. Comput. Chem.* **1995**, *16*, 1434–1444.
- [61] T. Dandekar, P. Argos, Identifying the Tertiary Fold of Small Proteins with Different Topologies from Sequence and Secondary Structure using the Genetic Algorithm and Extended Criteria Specific for Strand Regions, *J. Mol. Biol.* **1996**, *256*, 645–660.
- [62] B.T. Luke, Applications of Distributed Computing to Conformational Searches, In D.G. Truhlar, W.J. Howe, A.J. Hopfinger, J. Blaney, R.A. Dammkoehler (Eds.), *Rational Drug Design*, Springer, New York, **1999**, pp. 191–206.
- [63] C.A. del Carpio, A Parallel Genetic Algorithm for Polypeptide Three Dimensional Structure Prediction. A Transputer Implementation., *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 258–269.

- [64] J. T. Pedersen, J. Moult, Genetic Algorithms for Protein Structure Prediction, *Curr. Opin. Struct. Biol.* **1996**, *6*, 227–231.
- [65] J. Wang, T. Hou, L. Chen, X. Xu, Conformational Analysis of Peptides using Monte Carlo Simulations Combined with the Genetic Algorithm, *Chemom. Intell. Lab. Syst.* **1999**, *45*, 347–351.
- [66] R.S. Judson, M.E. Colvin, J.C. Meza, A. Huffer, D. Gutierrez, Do Intelligent Configuration Search Techniques Outperform Random Search for Large Molecules?, *Int. J. Quantum Chem.* **1992**, *44*, 277–290.
- [67] D.E. Clark, G. Jones, P. Willett, Pharmacophoric Pattern Matching in Files of Three-dimensional Chemical Structures: Comparison of Conformational-Searching Algorithms for Flexible Searching, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 197–206.
- [68] F. Glover, M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Boston, **1997**.
- [69] J. Lee, H.A. Scheraga, S. Rackovsky, New Optimization Method for Conformational Energy Calculations on Polypeptides: Conformational Space Annealing, *J. Comput. Chem.* **1997**, *18*, 1222–1232.
- [70] M.S. Head, J.A. Given, M.K. Gilson, ‘Mining Minima’: Direct Computation of Conformational Free Energy, *J. Phys. Chem. A* **1997**, *101*, 1609–1618.