

Global Optimization Methods

based on Tabu Search

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Julius-Maximilians-Universität Würzburg

vorgelegt von

Svetlana Stepanenko

aus Ivanovo

Würzburg 2008

Eingereicht am: 30.10.2008

bei der Fakultät für Chemie und Pharmazie

1. Gutachter: Prof. Dr. Bernd Engels

2. Gutachter: PD Dr. Reinhold Fink

der Dissertation

1. Prüfer: Prof. Dr. Bernd Engels

2. Prüfer: PD Dr. Reinhold Fink

3. Prüfer: Prof. Dr. Christoph Sotriffer

des Öffentlichen Promotionskolloquiums

Tag des Öffentlichen Promotionskolloquiums: 05.12.2008

Doktorurkunde ausgehändigt am: _____

*dedicated to the memory of my father,
Nikolay Valentinovich Kuznetsov*

Die vorliegende Arbeit wurde unter Anleitung von Prof. Dr. Bernd Engels von Mai 2004 bis Oktober 2008 am Institut für Organische Chemie der Julius-Maximilians-Universität Würzburg angefertigt.

Teilergebnisse dieser Arbeit waren Gegenstand von Publikationen^{333,338} sowie von Postern und Kurzvorträgen.

Acknowledgements

The preparation of this thesis would not have been succeeded the way it did without the support of many people contributed directly or indirectly. First of all, I extend my gratitude and appreciation to my supervisor, Prof. Dr. Bernd Engels, whose suggestions, help, valuable insights, motivation, and encouragement helped me in all the time of research and writing of this thesis, whose comments and criticism ensure the quality of it. My sincerest thanks also go to Priv.-Doz. Dr. Reinhold Fink, who kindly accepted the task of reviewing the present thesis, for his inspiring conversations, for the time he invested in helping me with all sorts of theoretical chemistry problems. I thank Prof. Dr. K. Baumann for fruitful discussions. My additional thanks go to all members of the group (past and current) for all their help, support, interest and helpful hints, for a friendly and humorous climate.

Ongoing financial support by the Deutsche Forschungsgesellschaft in the framework of the SPP 1178 and the SFB 630 is gratefully acknowledged.

Finally, I would like to express my dearest thanks to my parents, Nikolay Kuznetsov and Nadezhda Kuznetsova, who supported me for better for worse, for guidance in sending me down the right path in my education and in life, for their continuous love, encouragement, and understanding during my life. I wish to thank my family, Oleg and Vladimir Stepanenko for their insight, support, and patience, which allowed me to concentrate and finish my work on this thesis. Thank you for always standing by my side in all situations and believing in me when I didn't believe in myself.

“Die Phantasie arbeitet in einem schöpferischen Mathematiker nicht weniger als in einem erfinderischen Dichter.“

Jean-Baptist le Rond D'Alembert (1717 - 1783)

“Es ist nicht genug, zu wissen, man muß auch anwenden; es ist nicht genug, zu wollen, man muß auch tun.“

Johann Wolfgang von Goethe (1749-1832)

Table of Contents

Chapter 1	Introduction	3
1.1	Global Optimization (GO) Problem	4
1.2	Heuristics.....	6
1.2.1	Metaheuristic Features	7
1.2.2	Population metaheuristic	10
1.2.2.1	Swarm methods	10
1.2.2.1.1	Ant Colony Optimization (ACO).....	11
1.2.2.1.2	Particle Swarm Optimization (PSO)	14
1.2.2.2	Genetic Algorithms (GAs)	18
1.2.2.3	Scatter Search (SS).....	22
1.2.3	Simulated Annealing (SA)	25
1.2.4	Variable Neighbourhood Search (VNS)	30
1.2.5	Tabu Search (TS)	37
1.3	Conclusions	46
Chapter 2	TS based methods.....	48
2.1	Gradient Tabu Search.....	48
2.1.1	Algorithm description	48
2.1.1.1	Modest Ascent and Local Search Strategy.....	50
2.1.1.2	Diversification Strategy.....	55
2.1.2	TS Memory Elements.....	56
2.1.3	Implementation and Experiments.....	59
2.1.3.1	Parameters	60
2.1.3.2	Numerical Results	63
2.1.4	Conclusions	70
2.2	TSPA and GOTS	70
2.2.1	Description of algorithms.....	71
2.2.2	Investigation of the efficiency of the new approaches	75
2.2.2.1	Parameters	76
2.2.2.2	Tests of the efficiency	80
2.2.3	Conclusions	91
2.3	Conclusions	92

Chapter 3	Application and Discussion	94
3.1	Introduction	94
3.1.1	Conformational search techniques	95
3.2	GOTS application.....	106
3.2.1	Variable choice and numbering rules.....	106
3.2.2	Adaptation of the GOTS to the conformational search.....	109
3.2.3	Comparing the influence of different parameters	115
3.3	Experimental results.....	117
3.3.1	Conformational studies of amino acids.....	121
3.3.2	Conformational studies of acetylcholine.....	126
3.3.3	Conformational studies of ACE and HIV-1 protease inhibitors.	128
3.4	Conclusions	133
Chapter 4	Summary.....	134
Chapter 5	Zusammenfassung.....	139
Appendix A	Unconstrained Test Problems.....	145
Appendix B	Notation	148
References	152

Chapter 1 Introduction

One of the foundations of our world is the search for an optimal state. Since human beings exist, we exert for improvement. We have a strong wish to reach the maximum success with minimal efforts. Therefore, optimization is one of the oldest skills which even extends into daily life¹ and many problems in science, engineering, business and economics, such as acoustics equipment design, cancer therapy planning, chemical process modelling, data analysis, classification and visualization, economic and financial forecasting, environmental risk assessment and management, industrial product design, laser equipment design, model fitting to data, optimization in numerical mathematics, optimal operation of “closed” engineering or other systems, packing and other object arrangement problems, portfolio management, potential energy models in computational physics and chemistry, process control, robot design and manipulations, systems of nonlinear equations and inequalities, and waste water treatment systems management, travelling salesman problems (TSP)², the global optimization of Artificial Neural Networks (ANNs)³ and telecommunication networks⁴ and some applications appeared also in questions of the chemical industry^{5,6}, can be formulated as computing globally optimal solutions^{7,8,9,10,11}. In the case of a complex nonlinear system the associated decision model will in general have an enormous amount of local optima whose number is normally unknown. Typically, most of the local solutions are also unacceptable as compared with the global one. Therefore, general local optimization strategies are not applicable to the problems. Instead, a global search approach is required and one needs appropriate global optimization (GO) ideas and techniques.

The objective of GO is to find the best solution of a created mathematical model which corresponds to the global minimum (or maximum) of a suitable objective function within a given collection of feasible constraints. The objective function is a characteristic property of the system, such as profit, utility, damage, risk or error. Constrains may be given by existence conditions of the physical, technical, economic or some other system.

Due to the complexity of many optimization problems, particularly of high dimensional ones encountered in most practical settings, exact algorithms generally perform very poorly. Actually, metaheuristics^{9,12,13,14,15} is conspicuously preferable in practical applications and it

was shown to obtain highly accurate solutions in many cases. The power of metaheuristics is robustness and success for a wide range of problems. It is generally the method of choice if:

- Calculation of the objective function is very expensive or time consuming.
- The exact gradient of the objective function cannot be computed, or its numerical approximation is very expensive or time consuming.
- The values of the objective function may contain noise.
- The search space is very large.

If obtaining any feasible solution is not sufficient and the quality of solutions is critical, it is very important to investigate effective procedures to obtain the best possible solutions within a given time. In this thesis recent developments of intelligent search methods like Tabu Search and its application to problems arising in the chemical area are discussed and a new approach applicable to chemical problem is developed.

The purpose of this chapter is to introduce basic heuristic concepts of approaches that generate feasible solutions and to show how they can be applied to combinatorial optimization problems. In Chapter 2 new nonlinear global optimization routines based on the Tabu Search strategy are described. They try to determine the global minimum of a function by the *steepest descent - mildest ascent* strategy. The new algorithms are explained and their efficiencies are compared with other approaches by determining the global minima of several well-known test functions with varying dimensionality. This includes an investigation about the influence of user-defined parameters. The efficiency of the new approaches is also studied by comparisons with other approaches for the test cases. In Chapter 3 one of our methods is adapted to conformational search problems. It is tested by locating the global minimum energy conformation of amino acids, two angiotensin converting enzyme (ACE) inhibitors¹⁶, 2-acetoxy-N,N,N-trimethylethanaminium, and HIV-1 protease inhibitor¹⁷. The last chapter summarized this work.

1.1 Global Optimization (GO) Problem

GO aims to find optimal configurations for a given problem. Thus, it is first necessary to define what an optimum is¹⁸. In the case of an objective function F , an optimum is either a maximum or a minimum. To formulate the problem of the global optimization it is required that F is a function of the variable vector x_i , the component-wise bounds x_l and x_r related to the decision variable vector x_i are finite, and the feasible set D is nonempty. If this is fulfilled, the extreme value theorem¹⁹ guarantees that the global optimization model is a well-posed problem. Formally, a general optimization problem can be written as

$$\begin{aligned} & \min F(x_i) \text{ or } \max F(x_i) \\ & \text{subject to } x_i \in D \end{aligned} \quad (1)$$

where $F(x_i)$ is the continuous objective function with $D = \{x_i : \text{lower bound} \leq x_i \leq \text{upper bound}\}$. Depending on the problem x_i are binary, integer, or continuous variables. A well-known example from computational chemistry is the conformational search for a large and very flexible molecule. This task comprises all typical ingredients of an optimization problem. It possesses a large number of possible solutions with similar quality and the handling of the problem necessitates the scan over a large space. Finally, as in many optimization problems of econometrics, it remains uncertain if the optimal solution was really found.

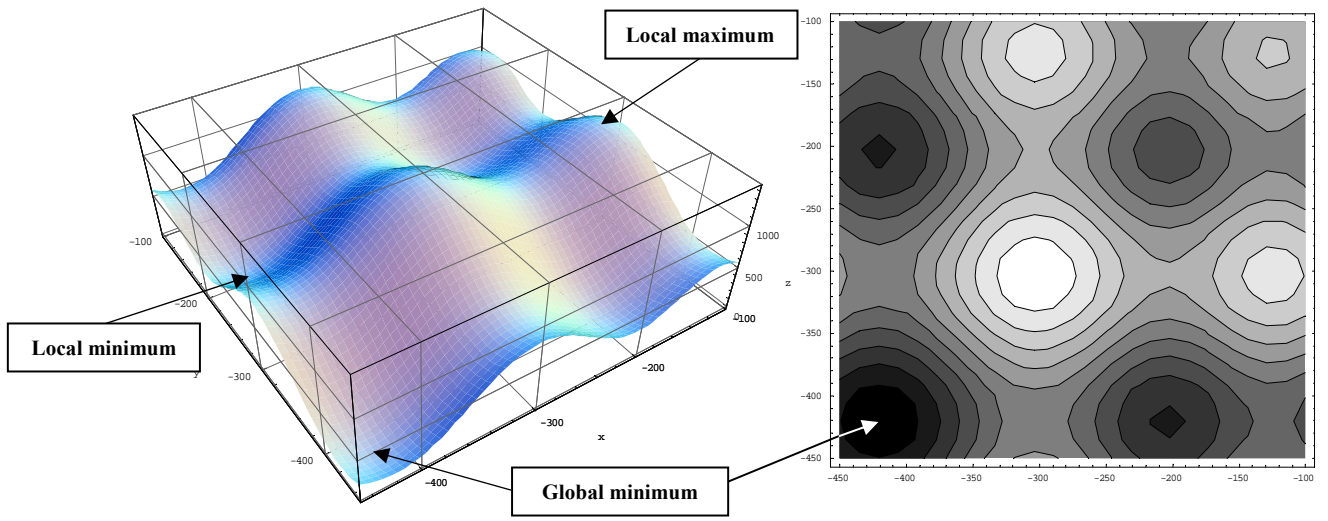


Figure 1-1. Three-dimensional and contour plot of the Schwefel function.

Figure 1-1 shows the Schwefel function defined over a two-dimensional search space $x_i = (x_1, x_2)$. Local and global optima are discriminated. A global one is an optimum of the whole feasible set D while a local optimum is only an optimum of one of its subsets. The definitions of local and global optima²⁰ are:

Local maximum definition: A local maximum $x_l^* \in D$ of an objective function $F(x_i)$ is a solution with $F(x_l^*) \geq F(x_i)$ for all x_i neighbouring x_l^* .

If $D \subseteq \mathbb{R}$, the definition can be written:

$$x_l^* : \exists \varepsilon > 0 : F(x_l^*) \geq F(x_i) \quad \forall x_i \in D \mid |x_i - x_l^*| < \varepsilon$$

Local minimum definition: A local minimum $x_l^* \in D$ of an objective function $F(x_i)$ is a solution with $F(x_l^*) \leq F(x_i)$ for all x_i neighbouring x_l^* .

If $D \subseteq \mathbb{R}$, the definition can be written:

$$x_l^* : \exists \varepsilon > 0 : F(x_l^*) \leq F(x_i) \quad \forall x_i \in D \quad |x_i - x_l^*| < \varepsilon$$

Global maximum definition: A global maximum $x_l^* \in D$ of an objective function $F(x_i)$ is a solution with $F(x_l^*) \geq F(x_i) \quad \forall x_i \in D$.

Global minimum definition: A global minimum $x_l^* \in D$ of an objective function $F(x_i)$ is a solution with $F(x_l^*) \leq F(x_i) \quad \forall x_i \in D$.

Depending on the context the term *optimum* can replace either one of the terms maximum or minimum. Some optimization problems search for a global maximum value while others search for a global minimum value. To find the global optimum and not just a local one, it is necessary to search globally in the search space. But many nonlinear optimization problems are non-convex and are likely to have many local optima. If traditional local search methods are used to solve such a problem, the outcome depends on the starting point of the search, i.e. as a rule only the next local optimal solution is found which can be of rather bad quality. In order to find the global optimum, a global search strategy is needed. The most important global optimization heuristic strategies are described below, together with short descriptions and references. Most global optimization software implementations are based on one of these approaches, possibly combining ideas from several strategies.

1.2 Heuristics

The word heuristic stems from Latin *heuristicus* or Greek *heuriskein* that means "to find, to discover". In computer science, a heuristic is a technique designed to solve a problem. The technique usually produces a good solution and helps to solve a simpler problem that contains or is in close agreement with the solution of the more complex problem. Heuristics increases computational performance or simplicity, at the expense of precision. There are a great variety of definitions in the literature^{21,22,23,24}. Heuristics is particularly used to find a solution that is usually reasonably close to the best possible answer. More accurate methods, especially when they are applied to complex problems, tend to show slow convergence that goes along with a high computational cost. The main reason for this slow convergence is that these methods explore the global search space by creating random movements without using much local information about promising search directions or already visited areas. In contrast, local search methods converge much faster as they use local information to determine the most promising search direction. However, they are easily entrapped in local optima.

The basic concept of the heuristic search as a support in solving of problems was introduced by Polya²⁵. A classic example of a heuristics is hill climbing. Hill climbing is a local search optimization technique. It can be applied to a large variety of optimization

problems. The algorithm starts with a random (potentially bad) solution and applies a local search to find an improved solution. If such a solution is found, the search moves to it and the local search starts again. The method stops when the solution is not improved upon already obtained solutions after ordinary local search. Ideally, at that point a solution closed to the optimal solution is found, but it is not guaranteed that hill climbing will ever come close to the optimal solution. The main disadvantage of a hill-climbing method is its incapacity to escape from a local optimum. A way to improve the performance of this simple heuristic procedure considerably is metaheuristic.

1.2.1 Metaheuristic Features

In the beginning, heuristics was typically applied to the solution of complex combinatorial optimization problems. The term “metaheuristics” for solving such tasks was introduced by Glover²⁶. It has been widely applied in the literature, both in comparative studies as well as in research papers. Since the introduction of metaheuristics by Glover a lot of books and monographs were published on this subject^{9,12,14,15,24,27,28,29,30,31,32}. Metaheuristics includes all heuristic methods which allow finding a good quality solution and also local optimization techniques. The picture has changed drastically. Modern powerful computers and parallel platforms allow successful applications of metaheuristics to real-time problems within acceptable time^{33,34}. However, metaheuristics gives no guarantee of obtaining the global solutions either.

The use of the metaheuristic search strategies makes it possible to escape from a local optimum during the iterative procedure. Metaheuristics has been developed to solve complex optimization problems in various fields. The efficiency of the procedures is due to the use of information about the environment. The method is a result of adjusting metaheuristic strategies to specific optimization problems. It is important to note that metaheuristic and “local optimality” approaches have essential differences. Typical heuristic concepts are "rules of thumb", educated guesses, intuitive opinions or simply common sense. They ignore whether the solution to the problem can be proven to be correct. Such iterative heuristics belong to local search methods such as descent or ascent methods. Therefore, the development of methods which allow moving aside from this classical design and broadening the application area is of great importance.

During the last few years, metaheuristics developed substantially. Now it is based on “intelligent” search methods that can be classified with respect to the three basic ideas:

- Use of adaptive memory and responsive exploration
- Neighbourhood search

- Number of current solutions carried from one state to the next.

These options help to classify metaheuristic methods in the form $X/Y/Z$, which was proposed by Glover and Laguna³⁵. The possible values of X , Y , Z elements are noted below:

X	A	method employs adaptive memory
	M	method is “memoryless”
Y	N	method uses some systematic neighbourhood search
	S	method relies on random sampling
Z	I	method with point-to-point strategy
	P	population-based approach

This simple classification scheme gives us a possibility to categorize various metaheuristics methods. Such a classification must stay ambiguous because of large quantity of method modifications; some of them are closed to “standard concepts”, others are more or less expanded developments of them. A metaheuristic method may be classified in two or more $X/Y/Z$ forms. Examples are shown Table 1.

Table 1. Metaheuristics classification.

Metaheuristic method	Classification 1	Classification 2
Genetic algorithms	M/S/P	M/N/P
Simulated annealing	M/S/I	M/N/I
Tabu Search	A/N/I	A/N/P
Scatter search	M/N/P	A/N/P

Surely, metaheuristics can include other strategies to find the global optimum. A metaheuristic method may modify the search strategy on the basis of the change of the objective function value during the search procedure. The same modification may be carried out during a neighbourhood search by excluding some members and introducing others. It can be illustrated at the example of the strategic oscillation approach of the Tabu Search. The scope of activity of the standard neighbourhood strategy that includes moves only among feasible solutions is extended in order to involve unfeasible solutions as well. Accordingly, the search overcomes the feasibility boundary in order to proceed into the unfeasible region.

Another classification of the metaheuristics, which differentiates between population-based strategies and single-solution metaheuristics, is also often found in the literature. In the latter methods, a search step requires only the information of a single preceding solution which the next iteration starts. On the other hand, population-based strategies invoke a collection of solutions at each stage. Such procedures are usually characterized by the class of *evolutionary methods*. Well-known examples are genetic algorithms, the scatter search, and the path relinking method³⁵ (a useful combination of intensification and diversification

strategies) based on strategies for “combining” solutions. It is also a conspicuous subclass of the metaheuristic methods which use multiple heuristics to generate new population members instead of relying on a single rule.

It is also possible to classify metaheuristic methods based on the use of adaptive memory. Patterns, whose present state depends on the sequence of previously visited solutions, and therefore includes a covert form using “memory” term, reside in practically all heuristics except those that use complete randomization. All previous choices are “remembered” and are inherited by the current one. The term “memory” has represented primarily by Tabu Search and its variations are sometimes called the “adaptive memory programming”³⁶, but a number of other metaheuristics use mechanisms that can also be considered as memories. In recent years, other approaches have made attempts to unify various aspects of such memory structures and strategies, however, typically in only rather simple form. Developments, which produce hybrids of the tabu search with other approaches at a more advanced level, have become an important way to embed an adaptive memory into other methods, and have established an active area of research. In genetic algorithms and the scatter search the memory is employed to store a population of the solutions, where the mode of combination more clearly lends itself to transmit features of selected past solutions to current solutions. However, such an implicit memory is not an intelligent memory construction. It lacks any mechanism to save the solutions prior to the last ones as well as methods to compare and improve the current solutions with the preceding generations.

Neural network approaches introduce another memory based distinction. Artificial neural networks have roots in our understanding of the human brain. Initial concepts were based on attempts to mimic the brain's way of processing information. Subsequent efforts gave rise to various models of the biological neural network structures and learning algorithms. Such methods accentuate an associative form of memory. Neural network approaches implicitly involve a form of optimization, and they have been applied to several optimization problems in recent years. In spite of a variable success of the performance, neural networks are often regarded as being appropriate to be included within the metaheuristic classification. Many research problems have been solved using neural network techniques (in particular the Hopfield network³⁷) including graph partitioning^{38,39}, knapsack⁴⁰, and constraint satisfaction problems⁴¹ as well as linear and nonlinear⁴² programming. Looi⁴³ mentioned that “although there is a large collection of operations research based approach and other methods for solving all of these problems, comparisons between neural network methods with existing methods have been lacking”. Some researchers

made up these deficiencies by successfully combining neural networks with Simulated Annealing, Genetic Algorithms, and most recently Tabu Search.

Metaheuristics is often considered to be a set of intelligent components, but it is important to note that the intelligence depends more on the underlying design than the particular property (or behaviour) of the method itself. For this reason, it is not necessary for a procedure to qualify as intelligent in a rigorous sense in order to grant its membership in the category of metaheuristics. Figure 1-2 shows a graphical representation of various metaheuristics whose principles are presented below.

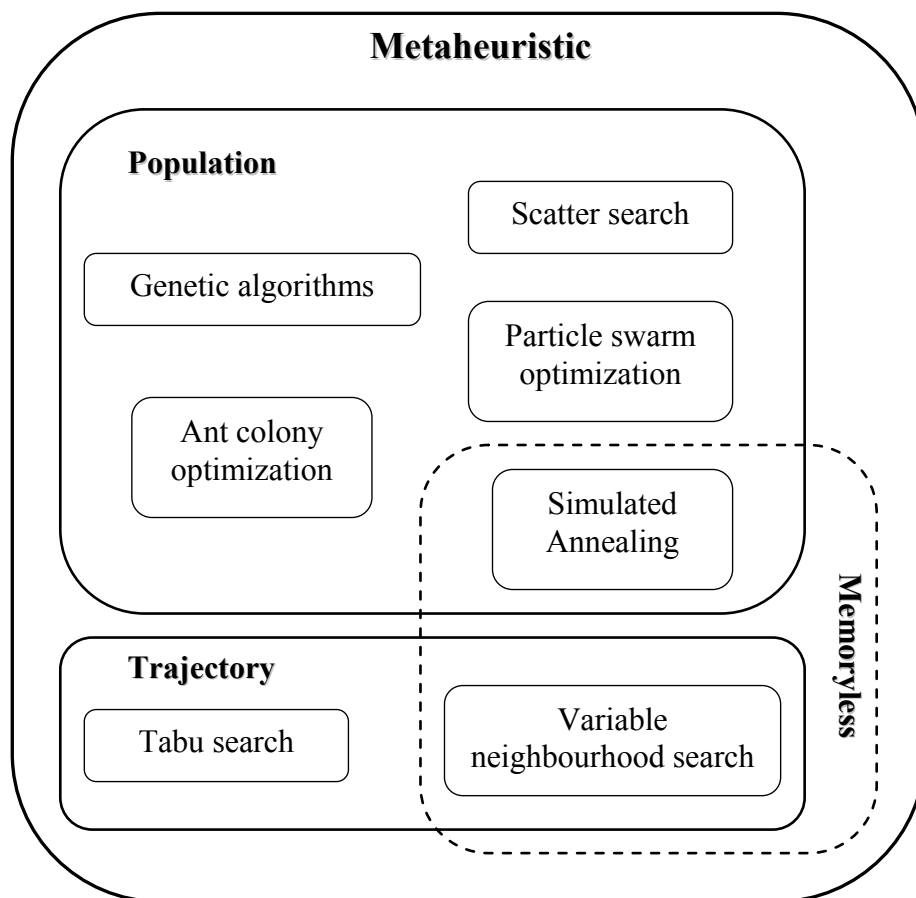


Figure 1-2. Metaheuristic methods.

1.2.2 Population metaheuristic

This class of metaheuristics uses populations of solutions. It provides better results than single-solution metaheuristics. New solutions are obtained by an implicit or explicit combination of different solutions.

1.2.2.1 Swarm methods

There are two popular swarm inspired methods in the area of computational intelligence: Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). ACO

was inspired by the behaviour of ants and has many successful applications in discrete optimization problems. The particle swarm concept originates from a simulation of simplified social systems and can be used as an optimizer.

1.2.2.1.1 Ant Colony Optimization (ACO)

The ACO algorithm introduced by Marco Dorigo in his PhD thesis⁴⁴ and originally developed by him, Maniezzo and Colormi in the early 1990s^{45,46}, is a probabilistic technique for solving combinatorial optimization problems that can be realized by finding good paths through graphs. The behaviour of ants⁴⁷ in finding paths from the colony to food sources gave rise to this idea. Even if a single ant has only restricted abilities, the behaviour of a whole ant colony is highly structured as the consequence of coordinated interactions. The way of ant's communication is a chemical compound, known as pheromone. A moving ant lays various quantities of these compounds on the ground, thus it marks the path its moves with a pheromone trace. Usually a single ant begins to move randomly, but by detecting a pheromone trail, the ant will follow it with higher probability intensifying it with its own pheromone. Hence the probability that ants will follow a path correlates with the number of ants that did so before. This is a form of *autocatalytic* behaviour — or *allelomimesis*. Pheromones also vanish with time if they are not refreshed. If all of the food is taken away from a particular place, ants will stop putting pheromones onto the respective track since they cannot find any food at this location anymore. The process is thus characterized by a positive feedback loop.

In Figure 1-3, an example is presented that this mechanism can lead to the shortest paths (as the pheromone tends to accumulate faster on these paths). It starts with a given path from the ant hill to food source. If this path is cut off by an obstacle, the ants have to pass the obstacle along the right or left path. Each ant makes a choice on the basis of some heuristic evaluations and the intensity of the pheromone trails left by previous ants. The path with a good heuristic evaluation and a high level of pheromone is more likely to be selected, so it gives the following ant a stronger stimulus and thus a higher probability to turn right. The first ant forced to decide which path must to be taken has the same possibility to turn right or left (since there is no pheromone trail on the alternative paths). If the right path is shorter than left one, the ant that took it will be faster than the ant following the left (long) path. Due to the shorter distance they can move more often. The next ants will find a stronger trail on the right path and it will become preferred (in probability) to the left path. The probability with which an ant decides to move along the path to follow is more and more prejudiced towards the shorter one because the intensity of pheromones increases faster on the shorter path. The final

result is that all ants will quick-witted in choosing of the shorter right path. However, it is significant that the decision is never deterministic, thus there remains always a possibility to explore alternative routes.

Computational models have been developed, to simulate this mechanism^{48,49}: the problems are visualized as (directed) graphs containing various junctions. In a preparation step, a few ant individuals perform randomized walks through the graphs. Pheromones are laid out in accordance with solution profit, so the probabilities to walk along given paths increase. In the following steps, the ants move again through the graph. They cannot only move along the already visited paths, but also can choose other routes, because randomization is used to allow the construction of a variety of different solutions. Again pheromones are laid out in accordance with solution profit. When all ants have constructed a complete solution, the procedure is restarted with the updated pheromone level. This is repeated for the number of allowed iterations or until the solution does not improve after a number of iterations. Thus, a minimal level of individual complexity can explain a complex collective behaviour. With a more complex prescription for the single steps it is possible to change the algorithm such that it escapes from local optima and copes better with environmental changes. However, a lowest limit is required to establish the desired behaviour.

ACO is now developed to a powerful, many-sided optimization tool with a lot of publications and numerous applications in diverse areas of operations research, management science and technology. A number of refinements have been integrated into this general iterative scheme. The approach of Dorigo and Di Caro²⁹ is an alternative to the Tabu Search, Genetic and Evolutionary Algorithms, Simulated Annealing, the Iterated Local Search, the Variable Neighbourhood Search, and some other search methods. Dorigo and Di Caro reviewed successful implementations of the ACO method which are applied to a number of important and difficult combinatorial optimization problems, such as:

- Quadratic assignment^{32,50,51}
- Travelling salesman^{52,53,54,55,56,57}
- Scheduling⁵⁸
- Connection-oriented network routing^{59,60,61}
- Connection-less network routing^{62,63,64}
- Vehicle routing^{29,32,65,66}
- Sequential ordering⁶⁷
- Graph colouring⁶⁸
- Shortest common supersequence⁶⁹
- Conformational analysis of flexible drug-like molecules⁷⁰.

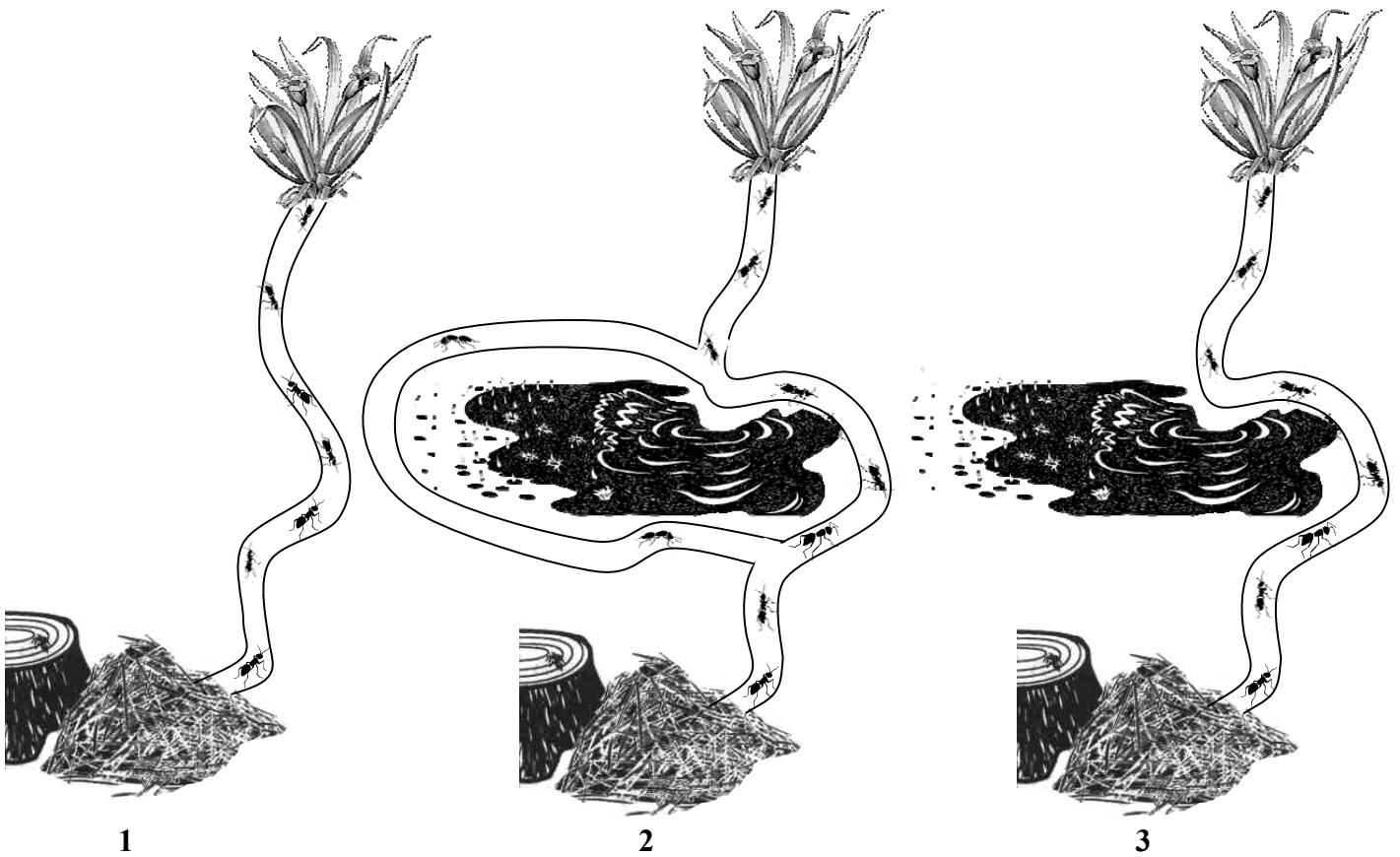


Figure 1-3. Allelomimesis example.

- 1) Some ants are walking on a path from the ant hill to food source
- 2) An obstacle suddenly appears and the ants must get around it
- 3) At steady-state the ants choose the shorter path

A large amount of experimental work on ACO was carried out in the last years. In their recent survey Dorigo and Blum used the ACO theory⁷¹. Meanwhile, the results of different ACO implementations have converged if applied to standard optimization problems^{72,73,74,75,76}, and also first investigations modelling the dynamics and the finite-time dynamics of ant colony optimization are carried out^{77,78}. Gutjahr reported an article⁷⁹ that addresses the important question how the (expected) runtime needed to obtain a solution of a given quality scales with the size of the problem. The first results with increased complexity of the iteration steps were given for two ACO algorithms. One of them is GBAS with lower pheromone bounds (GBAS/lb) which also forms the backbone of the Simulation based Ant Colony Optimization (S-ACO) algorithm^{80,81} for stochastic combinatorial optimization, which has already proven successful in applications^{82,83}.

Besides ant activity ACO can include pheromone trail evaporation and daemon procedures. These techniques are useful to avoid a too strong convergence of the algorithm towards a sub-optimal region. Pheromone trail evaporation decreases automatically the pheromone intensity as a function of time. Useful form of forgetting is implemented in order to favour exploration of new areas of the search space. Individual daemon actions are used to

centralize actions, which can be performed by single ants. Examples of such actions are local procedures or the collections of global information that can be used to analyse the usefulness of laying additional pheromone to bias the search process from non-local perspective. The daemon can monitor the path which was found by each ant and add extra pheromone to the shortest path. Such pheromone updates are called offline pheromone updates.

Korošec and Šilc⁸⁴ introduced that even real vector optimizations can be treated as graph problems. As shown by the simulation results, the approach can be extended to a broader class of problems. The general idea of this model is that of an autocatalytic process pressed by a "*greedy force*". The greedy force alone is incapable to find anything but a suboptimal tour. The autocatalytic process alone leads to convergence to a suboptimal path with exponential speed. In their combination greedy force steers the autocatalytic processes towards the best available local optimum and lets it converge quickly to very good, often optimal solutions.

1.2.2.1.2 Particle Swarm Optimization (PSO)

PSO is a stochastic, population-based evolutionary optimization algorithm. It was developed in 1995 by Kennedy and Eberhart^{85,86}. It uses the so called swarm intelligence that is based on social-psychological principles and comprehension biological social system. By analogy with a swarm, each swarm member (particle) takes advantage of own memory and has a degree of freedom, or randomness, in its movement as well as knowledge gained by the whole swarm to find the best available food source^{87,88,89,90}. After the first algorithm introduction, it has been studied by a number of different authors^{91,92,93,94} who mostly concentrated on multimodal mathematics.

A problem of a food search can be solved by optimizing a fitness function. The definition of the communication structure (social network) is realized by assigning the neighbours for each swarm. Each single solution is a "swarm member" called particle in the search space. All particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct their motion in the multidimensional search space. Any single particle iteratively evaluates the fitness of the solutions it meets and keeps track of the coordinates in the search space associated with the best solution (fitness) it has achieved so far as well as the fitness values. Each particle remembers the information about its best solution, its position in the search space and both are available to its neighbours. Thus, there is a local best solution given by the best solution of all particles in the neighbourhood. When all particles in the swarm are considered as neighbours, this best solution is the optimum presently available.

The swarm is typically simulated by particles in multidimensional space that have a position $x(p) \in \tilde{X} \subseteq \mathbb{R}^n$ and a velocity $v(p) \subseteq \mathbb{R}^n$. The position $x(p)$ of a particle p indicates a possible solution for the problem whereas its velocity $v(p)$ determines the direction of the subsequent search. Both the position and the velocity are real vectors. This ensured that PSO is especially suitable for a numerical optimization. The particles move through the solution vector hyperspace (i.e. \mathbb{R}^n) and know:

- their own best solution
- the presently best available solution,

where "best" simply means the solution with the optimal fitness function value. In the beginning, the positions and velocities of all individuals are randomly initialized. The swarm members swap for information about good positions with neighbours and update first their own velocity and then the position based on this information. In order to carry out appropriate changes of its position and velocity, each particle p has a memory holding the following information:

- $best(p)$ – “Particle best” position which is the best solution the particle has seen itself
- $best(N(p))$ – “Local (neighbourhood) best” position that the particle obtains by communicating with a subset of the swarm
- $best(X_{pop})$ – The optimal (“global best”) position ever obtained by any individual in the population that is known to all and immediately updated when a new best position is found by any particle in the swarm.

In the original PSO two different kinds of neighbourhoods were defined. To adjust the velocity of the particle p the PSO algorithm use either the $best(N(p))$ or the $best(X_{pop})$ positions.

- In the “global best” swarm, all the particles are neighbours of each other; thus, the position of the best overall particle in the swarm is used in the social term of the velocity update equation. Exploitation only the “global best” position leads to rapid convergence of the algorithm as all the particles are attracted simultaneously to the best part of the search space but decreases the probability to find the global optimum when the global optimum is not close to the “particle best”. It may even become impossible for the swarm to explore other areas. It means that the swarm is trapped in local optima.

- In the “local best” swarm, only a certain number of particles can affect the velocity of a given particle. With recourse of neighbourhood information the global optimum is found more likely for the prize of slower convergence.

The i_{th} particle velocity update equations for the cases of global and local PSO are given by

$$v(p)_i = v(p)_i + (c_i * random_n()) * (best(p)_i - x(p)_i) + (d_i * random_n()) * (best(X_{pop})_i - x(p)_i) \quad (2)$$

$$v(p)_i = v(p)_i + (c_i * random_n()) * (best(p)_i - x(p)_i) + (d_i * random_n()) * (best(N(p))_i - x(p)_i) \quad (3)$$

In both cases the i_{th} particle positions are updated by

$$x(p)_i = x(p)_i + v(p)_i \quad (4)$$

The learning rate vectors c and d have further influence of the convergence speed and the ability of the swarm to find the optimum. Moreover, the values of all dimensions of $x(p)$ are normally kept within the bounds of the search space. If the inertial coefficient of the velocity is small, all particles could slow down until they approach zero velocity at the “global best”. The fitness of the “global best” solution improves with each swarm iteration. It could also happen that all particles, which are to be influenced by the “global best” swarm, move in close proximity to the “global best” in the search space without exploring the rest of search space. In this case the fitness never improves in spite of number of made PSO iterations. The way to avoid this situation is to reinitialize the particle positions at a certain interval or after the detection convergence. Algorithm illustrated the native form of the Particle Swarm Optimization can easily be generalized for multi-objective optimization and for returning sets of optimal solutions.

Both “global best” and “local best” can be seen as "social" neighbourhoods, as the relations among particles does not depend on their positions in the search space, but on "external" relationships that are not dependent on the problem that is being solved. Kennedy and Mendes^{95,96} investigated the alternative "social" neighbourhood topologies. The following additional neighbourhood topologies were tested:

- Random
- “von Neumann”, a two dimensional grid with neighbours to the north, east, west, and south
- Pyramid, a three-dimensional triangular grid
- Star, all the particles connected to a central particle

- Heterogeneous, particles are grouped in several cliques
- Hypercube
- Ring

Illustrations of some of the topologies are presented on Figure 1-4.

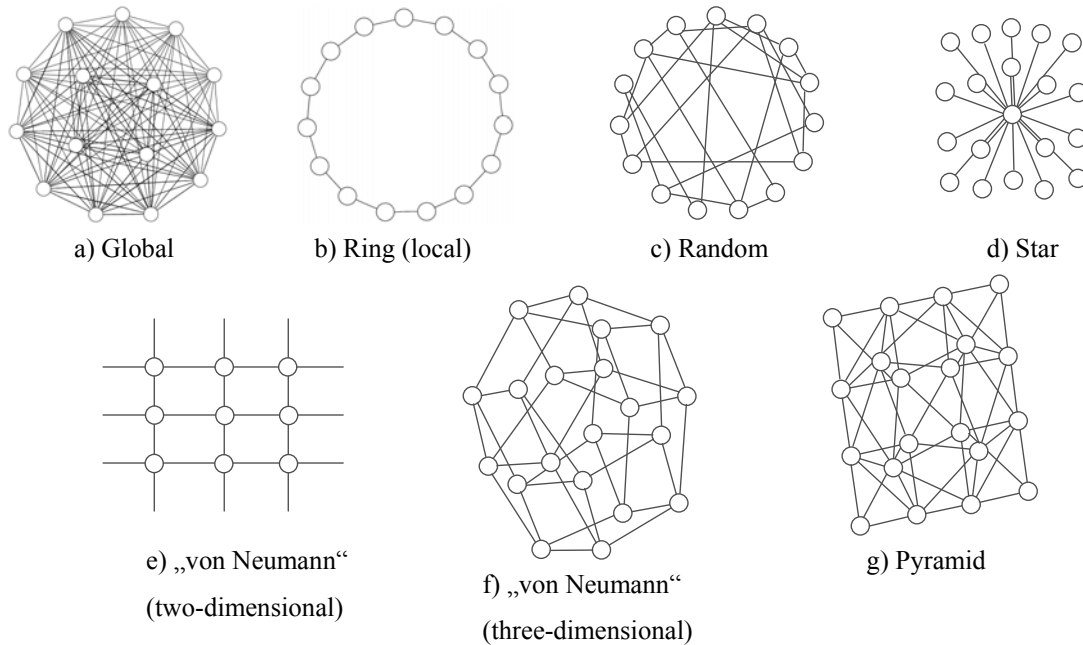


Figure 1-4. Neighbourhood topologies.

There are two factors offered by Watts^{97,98}, which can be used to characterize the different neighbourhoods:

- The degree of connectivity that measures the number of neighbours of a particle
- The amount of clustering that measures the number of neighbours of a particle that are also neighbours of each other

Despite the fact that the results depend on the selected variable, the “von Neumann” and Pyramid neighbourhoods prove to be the best while the star and “global best” are the worst. The concept of a "dynamic" neighbourhood was also explored⁹⁹. In the work of Li¹⁰⁰, a dynamic neighbourhood topology was used for a multimodal function optimization.

Recently, PSO has been successfully applied in many areas. Few applications of the algorithm to structural and multidisciplinary optimization are known. Fourie and Groenwold suggested exploitation of particle swarm in size and shape optimization¹⁰¹ and an application to topology optimization¹⁰². Some example areas of application of particle swarm optimization are:

- training of artificial neural networks^{86,103}
- training of hidden Markov models¹⁰⁴

- global optimization of mathematical functions^{86,105}
- in antenna or filter design¹⁰⁶
- water resource and quality management¹⁰⁷
- quantitative structure-activity relationship (QSAR) modelling in chemistry^{108,109}

The PSO algorithm was successfully applied to a continuous and integer/discrete structural optimization problem. It is inherently a continuous algorithm although it requires much higher computational cost than gradient-based optimization methods. The results show that the PSO algorithm is more suited for integer/discrete and discontinuous problems where use of a gradient-based optimizer may not be appropriate. PSO has many similarities with evolutionary computation techniques such as Genetic Algorithms (GAs). Both algorithms start a population of random solutions and search for optima by updating generations and using random techniques. They don't give guaranteed success. However, unlike GA, PSO has no genetic operators like crossover and mutation. The advantages are that PSO is easy to implement and possessing only a few fitting parameters.

1.2.2.2 Genetic Algorithms (GAs)

The idea of applying computer-aided simulations of biological principle of natural evolution to study artificial systems was first introduced in the 1950s by biologists like Barricelli. Some pioneering works, which traced back to the 1960s made by Bremermann¹¹⁰ and Bledsoe^{111,112}, use evolutionary approaches based on binary genomes to function optimization and balance weights for neural networks. Later on, Bagley¹¹³, Cavicchio¹¹⁴, and Frantz¹¹⁵ made a further important research on these genomes that crowns with the main presentation of the GAs of Holland in 1975¹¹⁶. Works related to the subject are from De Jong¹¹⁷, Baker¹¹⁸, Goldberg²⁷, Mühlenbein and Schlierkamp-Voosen¹¹⁹, Chipperfield et al.¹²⁰, Reeves²⁴, Michalewicz³⁰, and Berthiau and Siarry¹²¹.

GAs, evolution strategies, evolutionary programming, and genetic programming are grouped under the term evolutionary algorithms or evolutionary computation. They have been successfully applied to numerous problems from different domains, including optimization, automatic programming, machine learning, economics, ecology, and population genetics, studies of evolution and learning, and social systems. Although GA is a subclass of evolutionary algorithms, it was restrictedly applied until multipurpose presentation of Goldberg²⁷ in search, optimization, design, and machine learning areas. Nowadays, GAs are considered to be the most widely known and applicable type of metaheuristics^{30,122,123,124} with many applications of genetic algorithms in science, economy, research, and development¹²⁵.

A GA is a search technique to find exact or approximate solutions of various combinatorial optimization problems. GAs use techniques inspired by natural processes based on the Darwinian principle of natural selection such as “mutation”, “selection”, and “crossover” (also called recombination). Through the “selection” process, only the best solutions are allowed to become “parents” and to generate “offspring”. This probabilistically biases the algorithm towards the best elements in the population. “Crossover” is the mating process. With a given probability, it takes two selected individuals, called “parents” and combines their most desirable features by exchanging parts of their genomes solutions to create one or two new individuals, called “offspring”. In the simplest form, substrings are exchanged after a randomly selected “crossover” point. This operator tends to enable the evolutionary process to move toward “promising” regions of the search space. “Mutation” is performed for a few offspring: for such offspring, one variable is altered by a small perturbation, for instance the change of one bit in the binary coding case. It is introduced to prevent premature convergence to local optima by randomly sampling new points in the search space. “Mutation” entails flipping bits at random, with some small probability.

GAs are implemented as computer simulation of an optimization problem with a population of abstract representations, called chromosomes, genotypes or genomes, and of candidate solutions, called individuals, creatures, or phenotypes. For the definition of a typical genetic algorithm, one needs the followings:

1. genetic representation of the solution space
2. fitness function to evaluate the solution space.

Bit arrays are a standard solution representation, but also arrays of other types enabling the crossover operation can be used. A fitness function is a particular type of objective function defined over the genetic representation. It quantifies the optimality of a solution so that a particular solution may be ranked against all the other ones. The fitness function is always problem dependent. Optimal solutions or at least more promising solutions are allowed to breed and to mix for producing a new generation that will hopefully be even better.

The standard GA starts with an initial population that is randomly generated. Every evolutionary step is called a generation. The predefined quality criterion, the fitness or fitness function is then evaluated for each individual. To create a new population (the next generation), individuals are selected according to their fitness. Many selection procedures are currently in use, one of the simplest being Holland's original fitness-proportionate selection, where individuals are selected proportional to their relative fitness. This ensures that the expected number of times an individual is chosen is approximately proportional to its relative performance in the population. Thus, high-fitness individuals possess a better chance of

“reproducing”, while low-fitness ones are more likely to disappear. Parents are combined, by means of the recombination operator, to produce offspring. Before replacing the old population, the members of the new population receive the small random perturbations by means of the mutation operator. The offsprings mute with a given probability, and the fitness and the objective function value of the resulting offspring are computed. Then the next generation is created. The cycle is iterated until some optimization criteria are reached or when a maximum number of generations has been produced. Figure 1-5 sketches out the structure of a simple genetic algorithm.

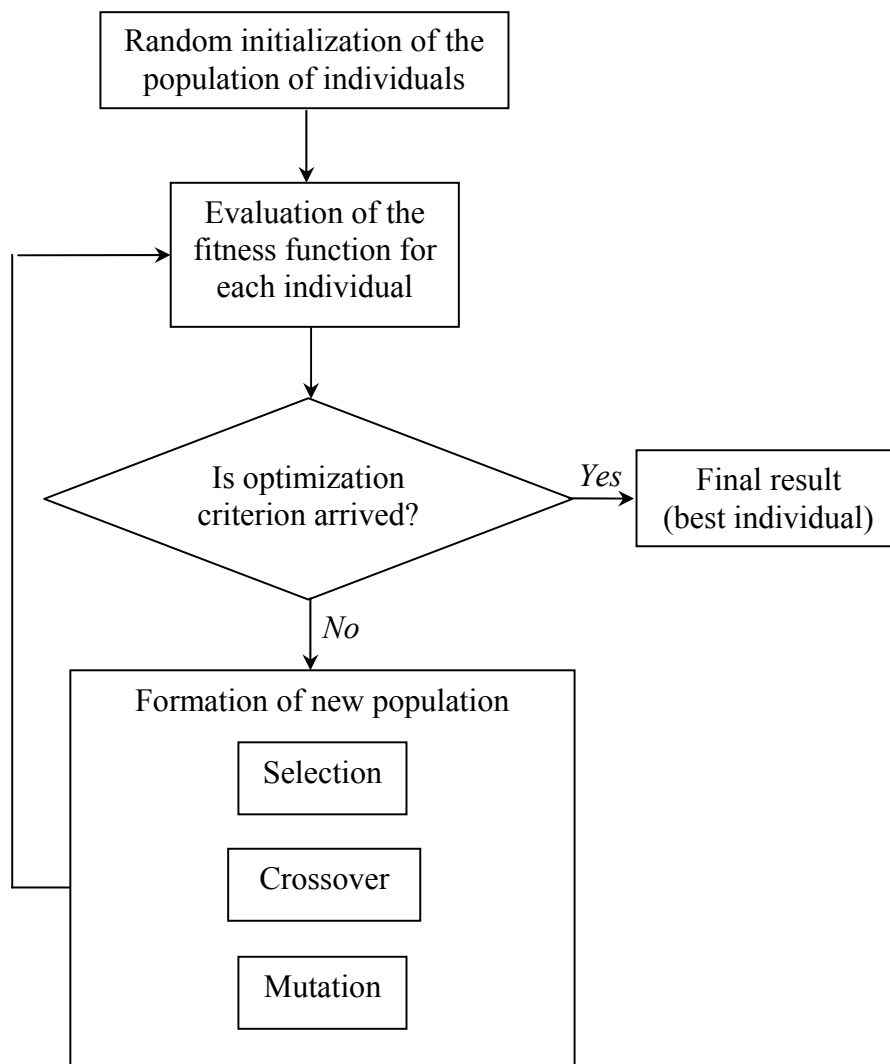


Figure 1-5. Simple genetic algorithm.

There are a large number of different types of genetic algorithms. The Continuous Genetic Algorithm (CGA)¹²⁶, which will be of some importance later, was proposed by Chelouah and Siarry as an adaptation of GA to the global optimization problems of continuous multim minima functions. It uses a type of real coding, which is as close as possible to Holland’s approach using binary coding. The main contribution of the CGA is the

introduction of two concepts also widely used in Tabu Search: diversification and intensification. In the diversification phase, CGA starts with a large population, and a high mutation probability, to cover the whole search space homogeneously, and detect a promising area. The intensification phase is then performed inside a promising area, after having reduced the search domain, the population size and the mutation probability. The “selection”, the “crossover”, and the “mutation” are performed using the decimal code. There are also combinations between GAs and derivative based methods¹²⁷ like the quasi-Newton method to solve difficult unconstrained optimization problems.

Genetic algorithms find application in bioinformatics, phylogenetics, computer science, engineering, economics, chemistry, manufacturing, mathematics, physics, and other fields. Some example areas of application of genetic algorithms are:

- scheduling applications, including job-shop scheduling^{128,129,130,131}
- chemistry and chemical manufacturing^{132,133,134}
- medicine^{135,136,137,138}
- data mining and data analysis^{139,140,141,142,143}
- geometry^{144,145,146,147,148}
- finance and trade¹⁴⁹
- optimizing distributed protocols^{150,151}
- building phylogenetic trees¹⁵²
- chemical kinetics (gas and solid phases)
- design of water distribution systems
- distributed computer network topologies
- game theory equilibrium resolution
- gene expression profiling analysis¹⁵³
- linguistic analysis¹⁵⁴
- marketing mix analysis
- mobile communications infrastructure optimization
- molecular structure optimization^{155,156}
- multiple sequence alignment¹⁵⁷
- operon prediction¹⁵⁸
- protein folding and protein/ligand docking¹⁵⁹
- RNA structure prediction¹⁶⁰
- timetabling problems¹⁶¹
- training artificial neural networks

- finding hardware bugs.^{162,163}

Genetic algorithms are a very effective way of quickly finding a reasonable solution to a complex problem. They don't give instantaneous and exact effects, but do an excellent job of searching through a large and complex search space. GAs are most effective in the search space with little knowledge about a problem to be solved, but can also produce solutions that only work within the test environment.

1.2.2.3 Scatter Search (SS)

Scatter search^{164,165,166,167} is a population-based metaheuristic that constructs new solutions by intelligently combining and improving previous solutions, called reference set. The update is made with regard to the results of the improvements. The methodology is very flexible, since each of the scatter search elements can be implemented in a variety of ways and degrees of sophistication.

SS algorithms apply "crossover" or combination mechanisms that allow the sharing of information between solutions to create new solutions¹⁶⁸ similar to the GAs. The main idea is a combination of the characteristics of two parent vectors to create several offspring, with the assumption that good solutions may generate better ones. The main search operator of the GA^{169,170} and SS¹⁷¹ approaches is handling of available information to influence future searches. Unlike the large population in GAs, the reference set of solutions in SS is rather small. In SS, the selection of the parents is not random as in traditional GAs, but the two or more elements of the reference set are chosen in systematic way. Due to this selection process that considers at least all pairs of solutions in the reference set, the reference set can be kept small. The methods applied in the SS to improve the solutions range from simple local searches to very specialized searches. For example, simple local searches are based on the selection of the most improving move or of the first improving move during the selection. Tabu Search³⁵, a Variable Neighbourhood Search¹⁷² or various sophisticated hybrid heuristic searches¹⁵ are also applied as variable neighbourhoods, intermediate memory, or hashing scanning methods of the neighbourhood. From this standpoint, SS procedures can be considered as memetic algorithms (MAs)^{29,173}, which combine local search heuristics with crossover operators.

From the implementation standpoint the SS method can be divided into the following subroutines based on the well-known templates^{165,174}:

- *The Diversification Generation Method* is used to generate a collection of diverse trial solutions from one or more arbitrary seed solutions as an input.

- *The Improvement Method* transforms a trial solution into one or more enhanced solutions. Neither the input nor the output solutions are required to be feasible, though the output solutions will usually be expected to be so. If no improvement occurs for the given trial solution, the resulting solution is considered to be the same as the one submitted for improvement.
- *The Reference Set Update Method* creates and maintains a set of reference solutions consisting of the best solutions found according to the criteria of providing efficient accessing by other parts of the method. The goal is to ensure diversity while keeping high-quality solutions.
- *The Subset Generation Method* generates subsets of the reference set as a basis for creating combined solutions.
- *The Solution Combination Method* uses weighted structured combinations to transform each subset of solutions produced by the Subset Generation Method into one or more combined solutions

The reference set is a collection of high quality solutions and diverse solutions. Both are required by the Solution Combination Method. Its goal is to produce weighted centres of selected subregions and to project these centres into regions of the solution space. This space shall be explored by auxiliary heuristic procedures. The reference set is generated from diverse solutions that improves during the search and will provide the information for the search process. The reference set update is based on comparisons between new and already visited solutions. To build the large set of diverse solutions the Diversification Generation Method is used. The size of the set of diverse solutions is typically smaller than the size of the reference set. The initial reference set is built according to the Reference Set Update Method.

The following simple mechanism demonstrates the initialization of the reference set and its updating procedure during the optimization search. The initialization of the reference set starts with the selection of the best solutions from the set of diverse solutions built by Diversification Generation Method. These best solutions are added to the reference set and deleted from the set of diverse solutions. For each improved solution of the updated diverse set the minimum of the Euclidean distances to the solutions of the reference set is computed and the solution with the maximum of these distances is added to the reference set. The process is repeated until the set of the diverse solution becomes empty. Solutions in the reference set are ordered according to their quality. The best solution is the first one in the list.

The simple Subset Generation Method creates all pairs of reference set solutions and puts them in a list to apply the Solution Combination Method. This method tries to combine good characteristics of the selected solutions intelligently to get new high quality solutions

after the improvement. At each iteration the number of subsets is generated depends on the number of new added solutions of the reference set. These trial solutions are subjected to the Improvement Method. The Improvement Method is the range of approaches from the simplest local searches to very specialized procedures. The reference set is updated according to the quality and the dispersion of the improved found solutions. The process is iterated with the new reference set until a stop condition is met. Finally, the set of disperse and high quality solutions in the reference set is provided by the method. A general template for the SS algorithm can be organized in two phases outlined as follows on Figure 1-6.

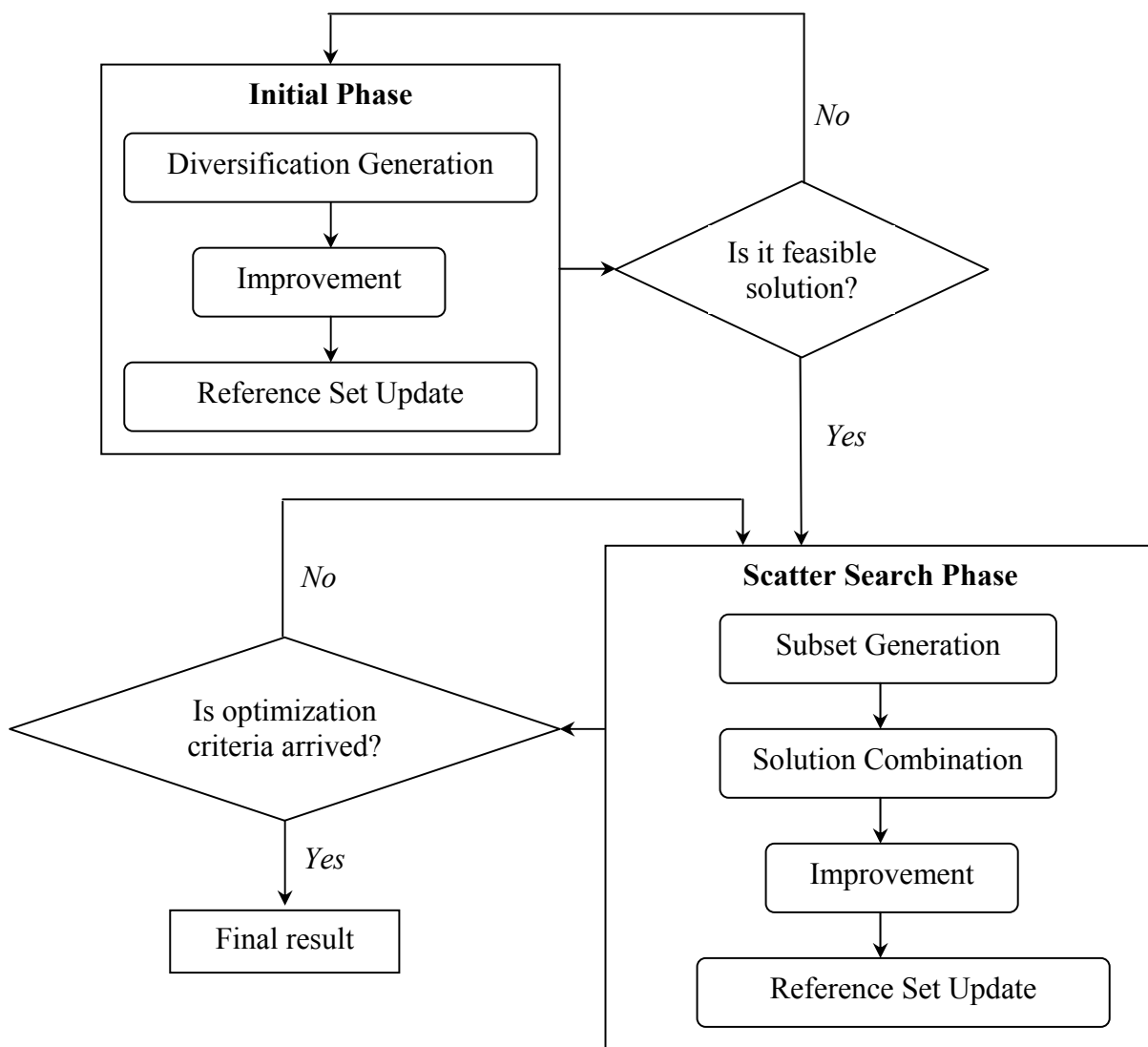


Figure 1-6. Scatter Search algorithm.

Recent studies confirm the practical advantages of this approach for solving a various optimization problems. SS differs from other evolutionary procedures, such as GAs, by providing combined principles for the solutions based on generalised path constructions in the Euclidean space. Furthermore, the search for the local optimum in SS is guided with less

space of randomization. Additional advantages of this approach are provided by intensification and diversification mechanisms. It uses adaptive memory elements that link SS to Tabu Search. The success of SS and related strategies such as continuous version of SS¹⁷⁵, which works directly with vectors of real components, is applied in a variety of application areas, such as:

- dense wavelength division multiplexing¹⁷⁶
- graph colouring¹⁷⁷
- data mining¹⁷⁸
- linear ordering problem²⁹
- network design problem¹⁷⁹
- history-matching problem¹⁸⁰
- chemical and bio-process optimization¹⁸¹
- 3D image registration problem¹⁸²
- neural network training¹⁸³
- multi-objective routing problem¹⁸⁴
- commercial implementation¹⁸⁵
- classical vehicle routing¹⁸⁶

SS is an evolutionary metaheuristic usually presented as a non-nature inspired one. However, most of the implementation that can be found in the literature sources have nature-inspired elements¹⁸⁷ in its components or are very similar to those used in standard nature-inspired metaheuristics.

1.2.3 Simulated Annealing (SA)

Simulated annealing (SA) is a generic probabilistic technique for locating good approximations to the global optimum in a large search space. It is based on the principles of thermodynamics. Name and idea of the method come from the annealing process in metallurgy, a technique involving heating, maintaining a suitable temperature and controlled cooling of a material. The goal is to induce softness, to relieve internal stresses, and to refine the structure so that the working properties of the material are improved. Annealing induces the diffusion of atoms within a solid material, so that the material progresses towards its equilibrium state. Heat increases the rate of diffusion by providing the energy needed to break bonds so that the atoms can move from their initial positions (a local minimum of the internal energy) and can wander randomly through states of higher energy. The slow cooling opens the chances to arrive at some more stable (global or local minimal potential energy) equilibrium. This general principle is applicable to both discrete and continuous global

optimization problems under mild structural requirements^{14,28,188}. It forms the basis of an optimization technique for combinatorial and other problems.

The original ideas of SA come from a paper published by Metropolis et al. in 1953¹⁸⁹ that introduced an efficient algorithm to simulate the equilibrium of a collection of atoms at a given temperature. The algorithm simulates the cooling process by gradually lowering the temperature of the system until it converges to a steady, “frozen” state. The adaptation of the Metropolis-Hastings algorithm in optimization as Simulated Annealing was first proposed by Kirkpatrick, Gelatt and Vecchi (1983)¹⁹⁰, and by Cerny (1985)¹⁹¹.

By analogy with this physical process, in a combinatorial optimization context a solution corresponds to a state of the physical system and the solution value corresponds to the energy of the system. At each iteration the current solution is replaced by a randomly selected trial solution. The probability to take the new or the old solution depends on the difference between the corresponding function values and on a main control parameter in the cooling schedule called temperature T . This means that the new solution is accepted according to the so-called Metropolis criterion. The difference between the function value of the resulting solution $F(pos_{i+1})$ and the function value of the current solution $F(pos_i)$ is defined as:

$$\Delta F = F(pos_{i+1}) - F(pos_i) \quad (5)$$

If ΔF is negative (i.e. the function value of the resulting solution is better than the function value of the current solution), resulting solution is accepted and becomes the new solution in the chain. If this difference is positive (i.e. the old solution is better than the new one), the resulting solution is only accepted on the basis of a comparison of some probability BF (Eq. 6) with a random generated number (RGN) between 0 and 1. BF is related to the magnitude of the cost increase and a parameter T . If $BF \geq RGN$, this new conformation is accepted; otherwise, it is rejected.

$$BF = \exp\left(-\frac{\Delta F}{kT}\right) \quad (6)$$

where ΔF is the increase in F and T is the main control parameter.

The main control parameter T is progressively lowered during the cooling process according to the given schedule and a certain number of iterations is performed at each temperature level. In general, a move is the most probable to be accepted if the temperature is high and the cost increase is low. The dependency is such that the current solution changes almost randomly when T is large. If T goes to zero only improving moves are accepted,

“downhill”, and the method stops at a local optimum, The allowance for "uphill" moves, which avoids to become trapped in local minima, is the major advantage over other methods, such as Tabu Search, Simulated Annealing. The illustration of the main principle of SA is represented in Figure 1-7. The new solution with better function value is always accepted, but the new state with worse function value is only accepted with a certain probability. The probability of accepting a solution is high at the high temperature and decreases at the drop in temperature.

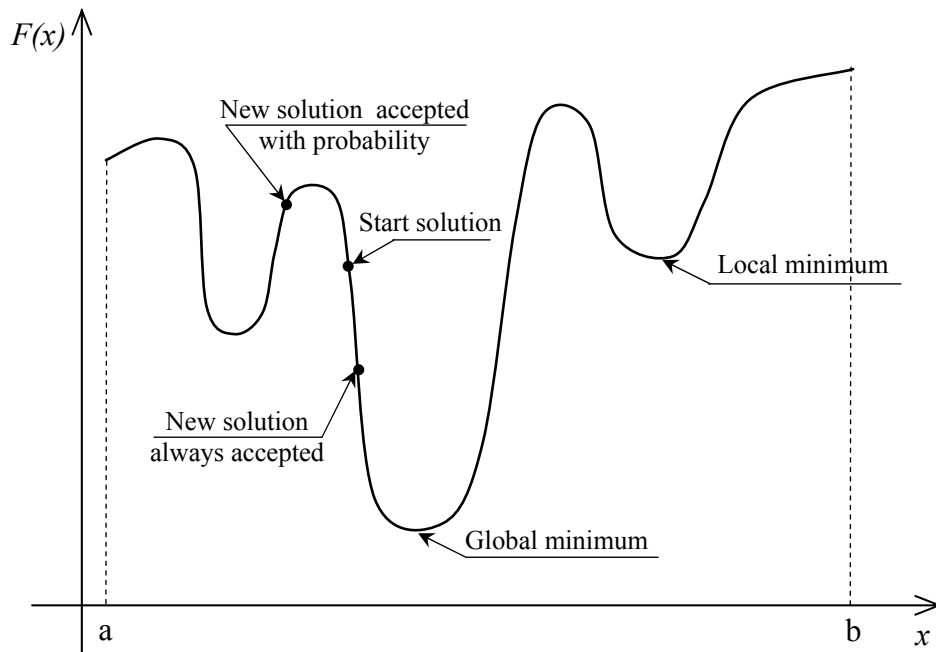


Figure 1-7. Selection of new states in simulated annealing.

The implementation of the basic SA algorithm is straightforward. The following Figure 1-8 shows its structure. It has been proved that with proper cooling schedule, SA asymptotically converges to a global optimum. However, this requires infinite number of iterations. Finite-time implementations do not provide such a guarantee.

Since the presentation of Kirkpatrick et al., a lot of studies that exploit SA have appeared. The theory as well as the applications of SA has been widely studied^{12,15,192,193}. The subsequent developments were carried out by Aarts, Korst, and van Laarhoven²⁸; Aarts and Ten Eikelder⁹; Henderson, Jacobson, and Johnson¹⁵ and have been focused on the following topics:

- convergence based on more general forms of acceptance rule than the Metropolis one
- deterministic variants^{194,195,196}
- different forms of static and dynamic annealing schedules^{197,198}

- parallel annealing^{199,200,201,202}
- thermostatical persistency²⁰³
- hybridizations with other metaheuristics^{204,205,206,207}

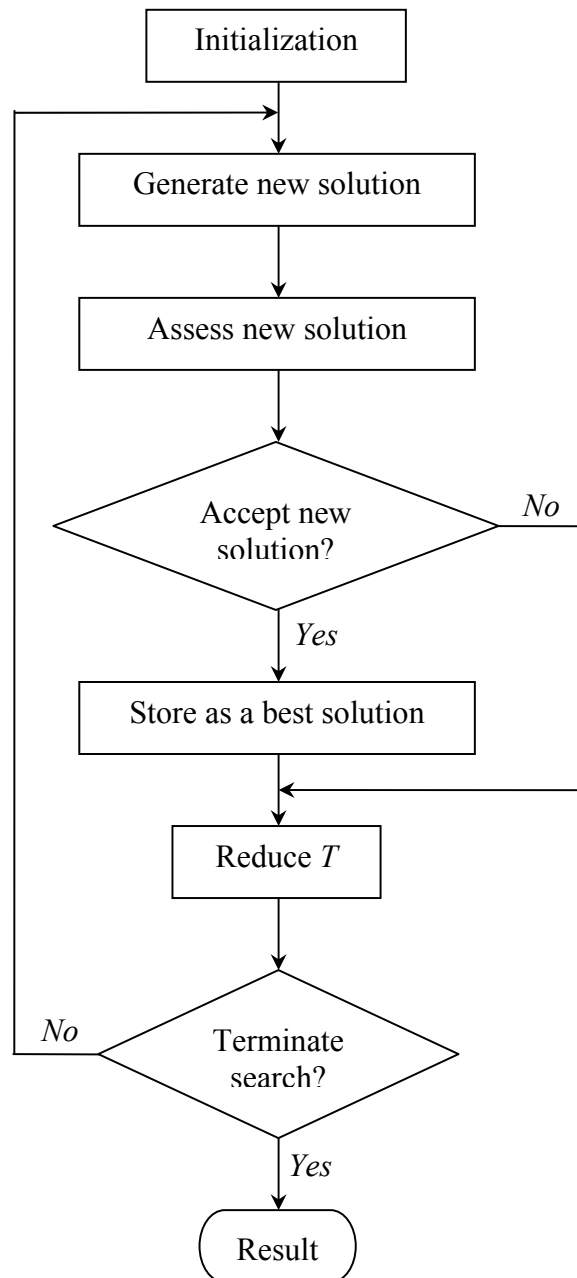


Figure 1-8. The flowchart of the SA algorithm.

SA becomes an algorithm of choice in such application areas as:

- financial instruments and trading^{208,209}
- complex portfolio selection problem²¹⁰
- econometric problems^{211,212}

- job shop scheduling problem (JSS)²¹³
- travelling salesman problem (TSP)^{214,215,216}
- vehicle routing problems¹⁹⁶
- global minimization of the Lennard-Jones energy function^{217,218}
- molecular conformation problem for peptide²¹⁹
- determination of silicon clusters²²⁰
- prediction of crystal structures^{221,222}
- design of the composite structures for aircraft²²³
- plastic design of circular steel plates²²⁴
- water distribution in irrigation canals²²⁵
- path planning^{216,226,227}
- paper cutting waste optimization²²⁸
- seismic waveform inversion²²⁹ and etc.

The ability of easily escaping of stagnations in local minima by accepting "uphill" moves through a probabilistic procedure especially in the earlier stages of the search is one of the most powerful features of SA. It is a robust and general technique and its main advantages are its flexibility and its ability to approach global optimality. It is able to handle more classes of constraints, highly nonlinear models, and chaotic or noisy data. This makes it, as neural nets and GAs, highly promising for portfolio optimization and asset allocation problems. However, it appears less appropriate for modelling financial time series. Due to the fact that the algorithm does not rely on any restrictive properties of the model, it is quite versatile. The possibility of "tuning" SA methods is also very significant, especially for any reasonably difficult nonlinear or stochastic system. This optimization algorithm can be tuned to enhance the performance. The ability to tune SA for more than one problem should be considered as an important feature of an algorithm.

A disadvantage is that the SA methods are computation-intensive. Since SA is metaheuristic, rather delicate work is needed to account for different classes of constraints and to fine-tune the parameters of the algorithm to adjust them to the actual problem. There is a clear tradeoff between the quality of the solutions and the time required to compute them. The precision of the numbers used in the implementation of SA have also a strong influence on the result quality. The main disadvantages that have been noticed on SA are its general slow convergence and its wandering around the optimal solution in the case where high accuracy is needed.

For some problems^{211,212}, SA algorithm performs better, e.g. more reliably in finding more optima, than other numerical techniques such as GAs. The technique appears well suited to asset allocation problems with constraints. It is often used when the search space is discrete. SA does not use gradient information and makes relatively few assumptions about the problem being solved.

1.2.4 Variable Neighbourhood Search (VNS)

VNS is relatively recent metaheuristic designed for solving optimization problems. It was proposed by Mladenović and Hansen^{9,15,230} and is based on the principle to explore neighbourhoods of growing size iteratively.

VNS is based on the following three rules:

1. The global optimum solution is a local optimum in all possible neighbourhood structures.
2. The local optimum solution with respect to a single neighbourhood structure does not imply that it is a local optimum in any other neighbourhood structure.
3. For many problems local optima of one or more neighbourhood structures are relatively close to each other.

The last item is an empirical observation. It suggests that a local optimum often provides information about the global one. Therefore an intensive study of the neighbourhood of this local optimum enables the finding of a better.

To describe the VNS algorithm a set of neighbourhood structures is denoted by N_i with $i = 1, \dots, i_{max}$ and the set of solutions in the i^{th} neighbourhood of the solution x is denoted by $N_i(x)$ (see Figure 1-9).

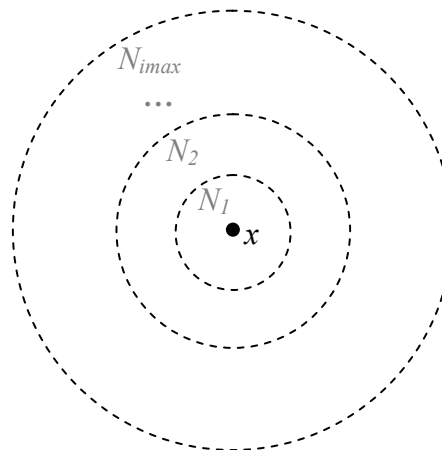


Figure 1-9. Set of neighbourhood structures.

Neighbourhoods N_k may be determined by one or more metric (or quasimetric) functions that define a distance between the elements of a set in the solution space. In order to solve general optimization problem (Eq. 1) using several neighbourhoods, facts 1 to 3 can be implemented in the following ways:

- a) deterministic approach
- b) stochastic approach
- c) combination of deterministic and stochastic approaches.

(a) Variable Neighbourhood Descent (VND) is a popular deterministic approach where the best neighbour of the current solution automatically becomes the new current solution if an improvement is obtained. The search is then restarted from this point. If the current solution is better than all its neighbours the next neighbourhood is considered. The search stops when all neighbourhoods are visited and no improvement is possible which means that the solution is a local optimum for all neighbourhood structures. The steps of the resulting basic VND scheme are given below:

Initialization:

Select the set of neighbourhood structures N_i , for $i = 1, \dots, i_{max}$, that shall be used in the descent;

Find an initial solution x ;

WHILE (no improvement is obtained){

Set $i=1$;

WHILE ($i < i_{max}$){

Exploration of neighbourhood: Find the best neighbour x' of x ($x' \in N'_i(x)$);

Moving: IF ($F(x')$ is better than $F(x)$){

Set $x = x'$ and $i = 1$;

}ELSE Set $i = i + 1$;

}

}

Most local search heuristics use in their descents strategies $i_{max} \leq 2$ neighbourhoods. Since the final solution should be a optimum with respect to all i_{max} neighbourhoods, the opportunities to reach a global optimum are larger than by using a single neighbourhood. A VND heuristic can easily be constructed for a given problem by combining it with available heuristics from the literature. The order of its applying and its ranking by non-decreasing size of the neighbourhoods has a significant influence. In addition to the sequential order of neighbourhood structures in VND above, one can go further and split neighbourhoods used in

a heuristic into a set of nested ones of increasing size (nested strategy). Such an approach is applied in Ref. 231, 232, 233.

To increase the efficiency of this scheme, one should consider the complexity of the different moves, the order of their application, the sufficiency of the considered moves to ensure a thorough exploration of the region containing x , and desired solution precision. As a consequence, if selecting and ranking moves involve a lot of elementary changes, the resulting heuristic may be very slow and often take more time than the exact algorithm on small or medium size problems. A frequent implementation involves ranking moves in the order of the complexity of their application and a return to the easiest type each time a new descent is found and explored. If some neighbourhoods are much easier to explore than others, the algorithm returns to the first neighbourhood as soon as an improved local optimum is found, instead of repeating all steps in sequence. The sufficiency of the considered moves is a crucial question: for some problems elementary moves are not sufficient to leave a narrow valley. Hence such heuristics can achieve poor results since they become trapped in local minima. In the beginning, one will strive to obtain the best possible solution within the allocated computing time by the exploitation of only the VND method. Afterwards, it is preferred to get good solutions fairly quickly by the VND and to improve it later by the faster stochastic search in VNS.

(b) Local optima of many combinatorial and global optimization problems tend to be close with each other and situated in one or sometimes several parts of D . Reached local optimum contains implicit information in its neighbourhood about better ones and even perhaps global optimum. Hence it is reasonable to explore the vicinity of the discovered optima.

The Reduced Variable Neighbourhood Search (RVNS)²³⁴ is the method of choice for these aims. The random solutions are selected from $N_i(x)$ without being followed by the search for the best neighbour. The neighbourhoods $N_i(x)$ with $i = 1, \dots, i_{max}$ are nested in most cases, i.e. each neighbour contains information about the previous one. It was observed that $i_{max} = 2$ is the best value. A solution is randomly chosen in the neighbourhood. If this solution is better than the previous one (i.e., if $F(x') < F(x)$ for minimization), the search is transferred to the new position ($x = x'$). Otherwise, the search is proceeded in the next neighbourhood. After visiting of all neighbourhoods the search continues again in the first one, until a stopping condition is met. Usually, a maximum number of iterations or a maximal computing time between two improvements is set as stopping criteria. Owing to the nested form the size of successive neighbourhoods increases. Therefore, close neighbourhoods of x are explored

more thoroughly than farther ones, even if no further improvements were observed within the first. The steps of RVNS are following:

Initialization:

```

    Select the set of neighbourhood structures  $N_i$ , for  $i = 1, \dots, i_{max}$ , that will be used in
    the search;
    Find an initial solution  $x$ ;
    Choose a stopping condition;
WHILE (stopping condition is not met){
    Set  $i = 1$ ;
    WHILE ( $i < i_{max}$ ){
        Shaking: Generate a random point  $x'$  from the  $i^{th}$  neighbourhood of  $x$ 
            ( $x' \in N_i(x)$ );
        Moving: IF ( $F(x')$  is better than  $F(x)$ ){
            Move to  $x'$  ( $x = x'$ ):
            Continue the search with  $N_1$  ( $i = 1$ ):
        } ELSE Set  $i = i + 1$ .
    }
}

```

RVNS is useful for cases where the local search is expensive. It is akin to a Monte-Carlo method, but more systematic.

(c) In the previous two ways, directions, which use variable neighbourhoods to reach a local optimum and to find promising regions for near-optimal solutions, were examined. Combination of the tools for both tasks leads to the General Variable Neighbourhood Search (GVNS) scheme.

Merging the local search with systematic changes of the neighbourhoods around the current local optimum leads to the Basic VNS method²³⁰. It is based on the idea to change the neighbourhood structure systematically within a local search heuristics, rather than staying in a single neighbourhood structure. In this basic scheme, a set of neighbourhood structures are selected. They are often nested, i.e. they define the neighbourhoods around the current solution $x \in D$ of the solution space. A solution x' is randomly generated in the first neighbourhood of the current solution. A local search is performed yielding the local optimum x'' of this neighbourhood. If x'' is worse or equal to the best previous solution, the procedure is iterated using the next neighbourhood. The search is recentered around x'' and restarted using this neighbourhood as the first one. If no better solution has been found or if every neighbourhood structure has been explored the search begins again at the first neighbourhood

$N_I(x)$ until a stopping condition is met. The stopping criteria may be the allowed maximum CPU time, maximum number of iterations, or maximum number of iterations between two improvements.

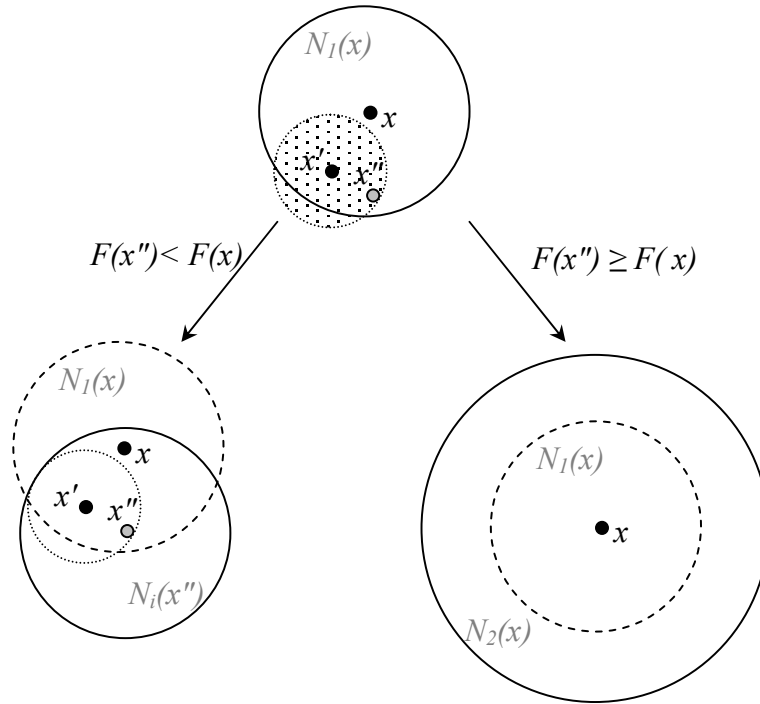


Figure 1-10. Variable neighbourhood search for minimization.

The illustration of the basic VNS is presented in Figure 1-10. The steps of this approach are following:

Initialization:

Select the set of neighbourhood structures N_i , for $i = 1, \dots, i_{max}$, that will be used in the search;

Find an initial solution x ;

Choose a stopping condition;

WHILE (stopping condition is met){

Set $i=1$;

WHILE ($i < i_{max}$) {

Shaking: Generate a random point x' from the i^{th} neighbourhood of x
 $(x' \in N_i(x))$;

Local search: Apply some local search method with x' as initial solution;
denote with x'' the so obtained local optimum;

Moving: IF (this local optimum is better than the incumbent){
Move there ($x=x''$);
Continue the search with N_1 ($i=1$);


```

    }ELSE Set  $i=i+1$ 
  }
}

```

The GVNS scheme is obtained if the VND is used for the local search of x'' and if the initial solution is improved by RVNS. This scheme has the following steps:

Initialization:

Select the set of neighbourhood structures N_i , with $i = 1, \dots, i_{max}$, that will be used in the shaking phase, and the set of neighbourhood structures N_l with $l = 1, \dots, l_{max}$ that will be used in the local search;

Find an initial solution x and improve it by using RVNS;

Choose a stopping condition;

WHILE (stopping condition is not met){

Set $i=1$;

WHILE ($i < i_{max}$){

Shaking: Generate a point x' at random from the i^{th} neighbourhood $N_i(x)$ of x ;

Local search by VND:

Set $l=1$;

WHILE ($l = l_{max}$){

Exploration of neighbourhood: Find the best neighbour x'' of x' in $N_l(x')$;

Moving: IF ($F(x'')$ better than $F(x')$){

Set $x' = x''$ and $l=1$;

}ELSE Set $l=l+1$;

}

Moving: IF (this local optimum is better than the incumbent){

Move there($x=x''$);

Continue the search with N_1 ($i=1$);

}ELSE Set $i=i+1$;

}

}

The VNS heuristic is able to find the best valleys rather quickly. To avoid trapping in a valley, the set of the neighbourhoods around all feasible solutions x should contain the whole feasible set D :

$$D \subseteq N_1(x) \cup N_2(x) \cup \dots \cup N_{i_{max}}(x), \quad \forall x \in D.$$

These neighbourhood sets may cover D without its partitioning. It is easy to implement, e.g. by nested neighbourhoods (i.e., each one contains the previous):

$$N_1(x) \subset N_2(x) \subset \dots \subset N_{i_{\max}}(x), \quad D \subset N_{i_{\max}}(x), \quad \forall x \in D.$$

It is also possible to explore D completely by investigating the small neighbourhoods around solution along some path, but such approach is not guaranteed for more efficient search. Nested neighbourhoods are easily obtained for many combinatorial problems. For this purpose it is necessary to define a first neighbourhood $N_1(x)$ by a type of move and then to iterate it i times to get neighbourhoods $N_i(x)$ for $i = 2, \dots, i_{\max}$. The sizes of the subsequent neighbourhoods increase. Since the investigation of the whole search space one goes many times through the whole sequence of neighbourhoods. In this case the first neighbourhoods will be explored more thoroughly than the last ones. This meets the requirements of third rule, i.e. local optima with respect to one or more neighbourhood structures are relatively close to each other. Moves to the feasible set D may be constrained, particularly if this set is disconnected. Introducing some or all constraints in the objective function with Lagrangian multipliers allows moving to infeasible solutions²³⁵.

Three improvements of the basic VNS for solving large instances are now considered:

- the Variable Neighbourhood Decomposition Search (VNDS) method²³⁴ extends the basic VNS into a two-level VNS scheme based upon decomposition of the problem;
- the Skewed Variable Neighbourhood Search (SVNS) method²³⁶ addresses the problem of exploring valleys far from the previously found best solution;
- the Parallel Variable Neighbourhood Search (PVNS) method^{15,237,238} is a way for parallelizing VNS

As the change of the neighbourhood during the search for good solutions to (Eq. 1) is a simple but very powerful tool, several authors have added such a feature to other metaheuristics rather than VNS:

- Tabu search^{239,240,241,242}
- GRASP^{243,244,245}
- Constraint programming^{246,247}.

VNS has been successfully applied in many areas; some of them are listed below:

- Vehicle routing problem²⁴²
- Arc routing problem²⁴⁸
- Travelling salesman problem²⁴⁹
- Linear ordering problem²⁵⁰
- Graph coloring²⁵¹
- Protein side chain placement problem²⁵²

- Molecular Distance Geometry Problem²⁵³
- Fuzzy clustering problem^{254,255}
- Finding the three-dimensional structure of a molecule²⁵⁶
- p-Median problem²⁵⁷
- Simple plant location problem²⁵⁸
- One- and multi-dimensional knapsack problem^{259,260}
- Oil pipeline design problem²⁶¹
- Maximum clique problem²⁶²
- Degree-constrained minimum spanning tree problem²⁶³
- Max-cut problem²⁶⁴
- Cable layout problem²⁶⁵
- k-Cardinality tree problem^{266,267}
- Generalized minimum spanning trees problem²⁶⁸
- Nurse rostering problems²⁶⁹
- Resource-constrained scheduling problem²⁷⁰
- Generalized minimum edge biconnected network problem^{271,272}
- Pooling problem²⁷³.

VNS metaheuristic is based on a simple and clear principle, which is easy to understand, and most important, easy to use. This approach provides optimal or near-optimal solutions for solving the problems of several benchmarks within reasonable computing times. Furthermore, the performance of the VNS appears to be robust. It has a few parameters or sometimes none. Using VNS one could obtain good or better results than most other metaheuristics on many problems. In view of the aforesaid VNS fulfils the requirements of desirable properties of metaheuristics and may play an important role in the design of new heuristics for new types of applications.

1.2.5 Tabu Search (TS)

Let's dwell up on a description of the TS method that forms the basis of our new approaches and which developed to one of the most powerful methods for solving combinatorial optimization problems.

TS is a metaheuristic algorithm that belongs to the class of local search techniques with the possibility to accept worse-cost local solution in order to escape from local optimum. It was first proposed in its present form by Glover in 1986²⁷⁴. Seminal related ideas were also developed by Hansen²⁷⁵ which he called *steepest ascent/mildest descent* method. It is significant that in Glovers view TS was a metaheuristic, i.e., a general strategy for guiding

and controlling interior heuristics specifically adjusted to the problems. Additional efforts of formalization and further developments for combinatorial optimization problems were reported later^{35,276,277,278}. A didactic presentation of TS and some of its applications have been collected in a book of Glover, Taillard, Laguna, and de Werra²⁷⁹.

The basic principle of TS is to guide a local heuristic search procedure for exploration of the solution space beyond local optimality. TS works on a single solution candidate. The local search procedure uses an operation called *move* to create a neighbouring individual of this candidate. The definition of the moves is highly problem-specific. The combination of the current solution with the possible moves defines the neighbourhood of the current solution. Its size simpliciter depends on the number and the kinds of defined moves. Consequently, if every neighbour has to be evaluated at each iteration, the algorithm may become quite slow. Therefore, usually only a subset of the neighbourhood is considered. Random selection of a fixed number of neighbours for consideration is a commonly employed method.

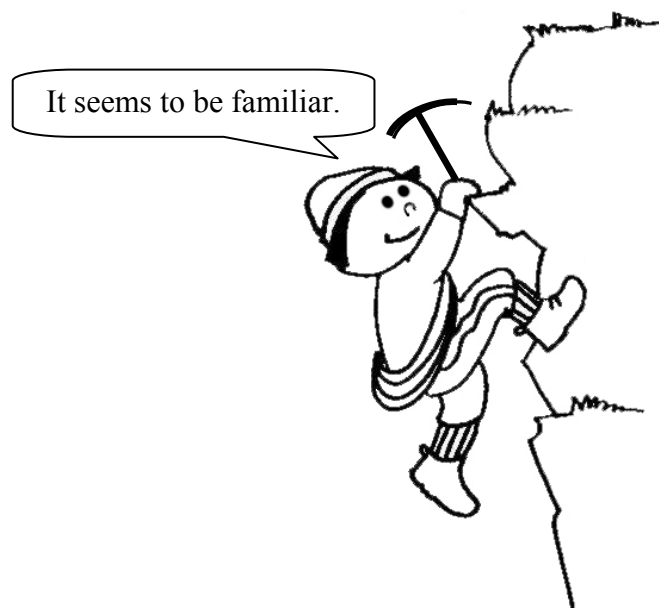


Figure 1-11. Adaptive memory and responsive exploration.

The main feature of TS is its use of an adaptive memory, which creates a more flexible search behaviour, and responsive exploration. Mountain climbing can be used as an illustrative example of both features. An adaptive memory term can be explained by the example of an alpinist who selectively remembers the path elements during the climb of the mountain in memory. Using this knowledge to make a strategic choice along the way represents the responsive exploration. It is illustrated in Figure 1-11. The adaptive memory feature of TS allows the implementation of procedures that are capable of searching the solution space more economically and effectively. Using the mountaineer analogy, one must

analyze current alternatives using previously obtained attainments during ascents in similar regions. The responsive exploration in TS, whether in a deterministic or probabilistic implementation, rests on the assumption that a bad strategic choice yields more information than a good random choice. It integrates the basic principle of intelligent search i.e., using features of previous good solutions to explore new promising regions.

Simple TS combines a local search procedure with anti-cycling memory-based rules to prevent the search from getting trapped in local minima, so that a global optimization can be conducted. It restricts the return to recently visited solutions by constructing a list of them called Tabu List (TL). The last visited solution is placed on a TL, and the reverse move is forbidden during a certain number of iterations. Certainly, after a local optimum is found, the next neighbour is reached by an uphill move. In most cases, during the next iterations, the search will try to go downhill again, but this reverse of an already accepted move is tabu, so that the algorithm has to continue to go uphill along the modest ascent direction, i.e. it will leave the local optimum. An important parameter is the TL size, which determinates the number of moves that are forbidden, i.e. how long the visited solution will remain prohibited. TL is usually managed with respect to FIFO (First In First Out) principle. The longer TL is, the smaller the chances that the algorithm gets back to already visited solutions are, i.e. the smaller the chance that it is trapped in local minima is. But with the increasing size of TL the search gets more and more limited since feasible solutions could be missed because the moves leading to them are tabu for a too long time.

Specific properties of TL or attributes of the tabu solutions can be made more effective for some application domains. However, it can also put new problems and can lead to the complication of the algorithms, because it may prohibit a new promising solution even when there is no danger of cycling or may lead to an overall stagnation of the searching process. Therefore, to avoid such problems aspiration criteria can be defined which revoke TL, so that otherwise excluded solutions are included in the allowed set. The simplest and most commonly used aspiration criterion includes solutions, if their objective values are better than the current best-known solution. More complicated and therefore rarely used aspiration criteria are the neglect of all tabus if cycling cannot occur. So, the update of the memory-based TL can be modified and controlled by the following concepts:

- Tabu tenure: the length of time during which a certain move is classified as tabu. It can be kept constant or varied dynamically throughout the search.
- Aspiration criteria: to accept improving solution the existing tabu rules can be overridden.

In each iteration TS generates the neighbourhood of the current solution. After that the selection of the subset is carried out by the remove of the tabu solutions from the neighbourhood. Each solution in this new neighbourhood subset is evaluated with regards to the objective function. Certainly, a generation process that avoids generating a trial solution that is already recently visited is preferred. The best solution in the neighbourhood of the current solution is selected as the new current solution and becomes the “initial” solution for the next iteration. This will also include uphill movements which are necessary to escape local minimum and to avoid getting trapped in an optimum. Such uphill movement go along the modest ascent direction. In Figure 1-12 an algorithm scheme of a TS algorithm is defined.

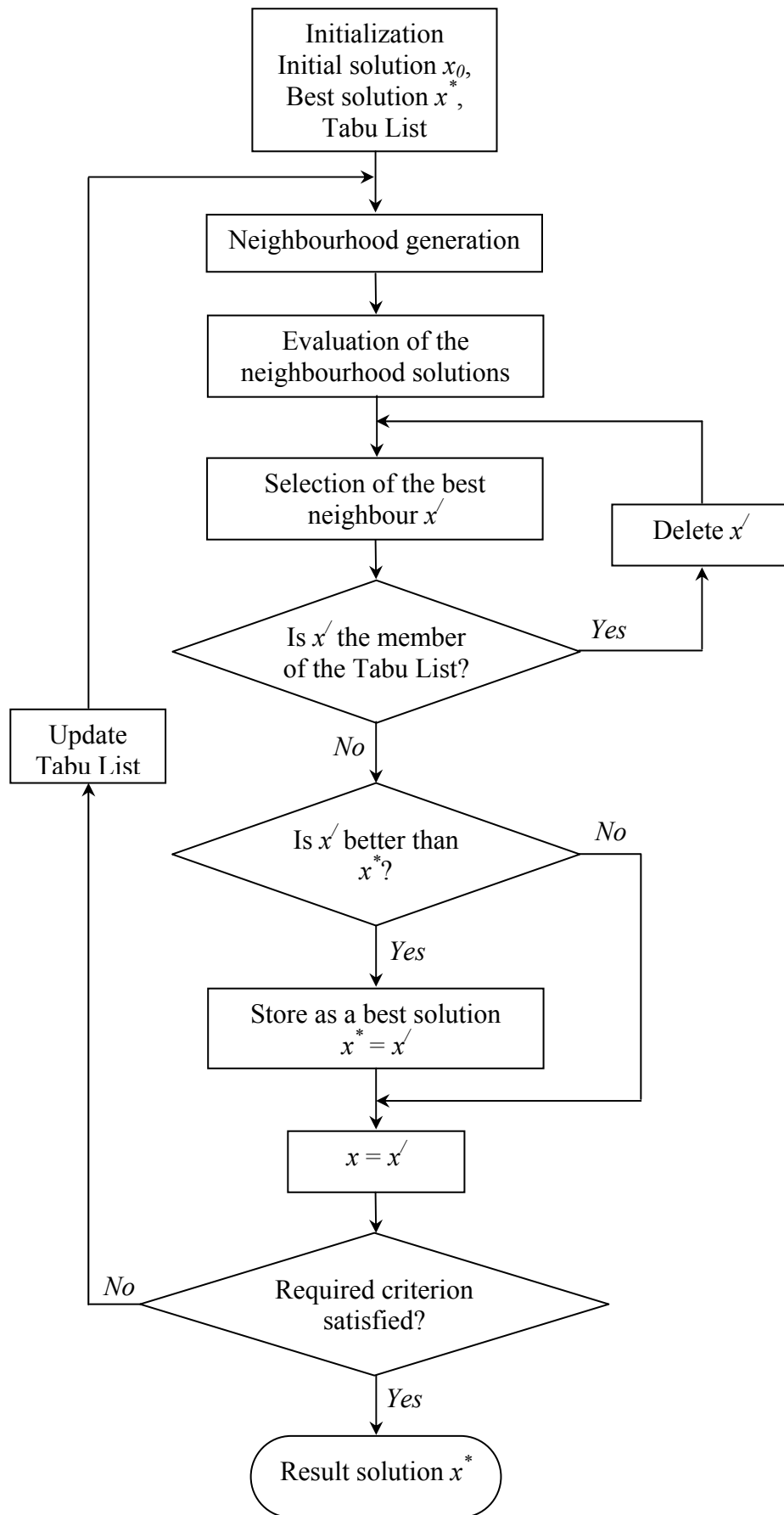


Figure 1-12. Flowchart of Tabu Search.

Two very important components of TS are intensification and diversification strategies. The basic purpose of the diversification search is to explore unvisited regions of the search space that have not been investigated previously, whereas intensification search intensifies the search within a limited space, e.g. the promising region. This main difference between intensification and diversification is illustrated in Figure 1-13.

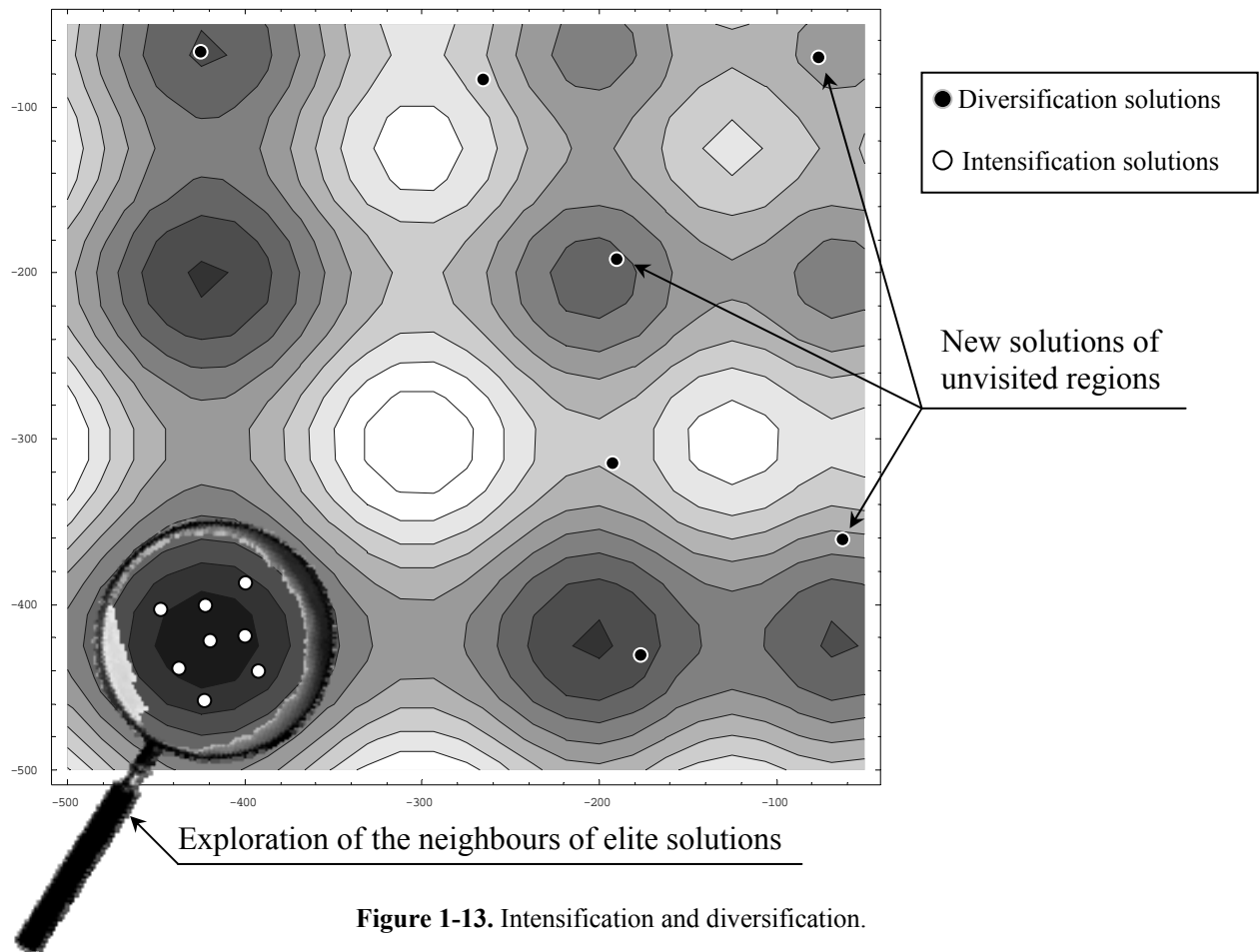


Figure 1-13. Intensification and diversification.

Typically, TS stops after a fixed number of iterations or a maximum number of consecutive iterations without any improvement of the incumbent solution. During these iterations TS keeps sufficient information about the search in memory. Such a simple TS structure is called short-term memory TS. The short-term memory used in the search intensification holds only the recent history of all moves or subset of them. Its basic idea is to determine the attributes of performed moves. This information is used to encourage move combinations and solution features, which were found good in previous stages of the search. It is also possible to return to previous promising regions to search them more thoroughly. The intensification search focuses on investigation of the neighbourhood of the best solutions found during the search which are called elite solutions. In intensification stage the term neighbour has a broader meaning as in the usual context of neighbourhood search. During the intensification search neighbours are also produced by the combination of the components of

good solutions or by modified evaluation strategies that favour the introduction of such components into a current solution.

Long-term memory has been suggested to achieve a better performance and to keep a more important search information such as the recency, the quality, and the frequency⁹. Specifically, long-term memory in high-level TS saves attributes of special characters like elite and frequently visited solutions. Accordingly, the second highly important component of the TS, the diversification strategy, may be based on some long term memory, such as frequency-based memory. This frequency-based memory keeps the total number of iterations that various “solution components” have been presented in the current solution or have been involved in the selected moves. By means of this algorithmic mechanism, one makes the attempt of forcing the search into previously unexplored areas of the search space and generates solutions that differ in various significant ways from those seen before.

There are some major diversification strategies. The simplest form of diversification is certainly the classical random restart diversification, in which the search is periodically stopped (according to some predetermined criterion) and then restarted from a randomly generated solution. More interesting but also more complicated approaches select restart solutions by using historical information collected in a long-term memory. Examples may be rarely used components in the current solution or the incumbent solution. Such information can be used in different ways, for instance, to avoid highly frequent moves or, conversely, to encourage the use of moves having very low frequencies and finally to design restart mechanisms within TS. This principle can also be exploited in a continuous diversification which integrates diversification considerations directly into the regular searching process. This avoids interruptions and the loss of information accompanied with the restart scheme. This is achieved by the modified evaluation of the possible moves which added a small term related to the component frequencies²⁸⁰ to the objective function. In this way, neighbourhood search can be directed into yet unexplored regions where TL operation is restarted.

The TS ideas of intensification and diversification are beginning to find their way into other metaheuristics. Since the appearance of the simple TS scheme described above, many new developments and refinements have been proposed over the last few years. They are briefly mentioned in the followings:

- intensification and diversification techniques: In addition to previously described methods the following forms of long-term memories^{35,280} are often implemented:
 - adaptive memory procedure referred to as “probabilistic diversification and intensification”, which was first proposed by Rochat and Taillard²⁸¹ for the vehicle routing problem. Adaptive memory involves an attribute-based focus which depends

intimately on the elements of recency, frequency, influence, and logic. Its main idea is that attributes of high quality solutions are used to construct other high quality solutions, as it was done in GAs. The method therefore stores a continually updated pool of previously generated elite solutions, which can be extracted from the pool and used to restart the search. The part of the pool that represents better solutions has a larger probability of being selected. Usually, different fragments of elite solutions are taken and combined to generate a new starting solution. If the components are taken from elite solutions, which are situated in a common area of the search space, this is the case of intensification, otherwise it is a question of diversification. Adaptive memories provide a generic paradigm for guiding local searches and can be coupled with different types of metaheuristics.

- strategic oscillation or tabu tunneling³⁵ is the method to achieve an effective cooperation between intensification and diversification. It manages moves in relation to a critical level called the oscillation boundary which often represents a point where the manner would normally stop. The boundary is identified at various stage of the construction or it represents a chosen interval of functional values. The search is allowed to go for a specified depth beyond the boundary before turning around. When the boundary is crossed again from the opposite direction, the search goes beyond it for a predefined depth and after that turning around again. The process of repeatedly approaching and crossing the critical level from different directions creates an oscillatory behaviour. It is possible to control this behaviour by varying the amplitude of the oscillation, by generating modified evaluations and rules of movement to explore a region of the search space depending on the type of the region and the search direction.
- path relinking^{35,282} generates new solutions by exploring connecting trajectories of elite solutions. It starts from one of these solutions, called initiating solution, and generates a path in the neighbourhood space leading to another solution, called the guiding solution. This can be done by selecting moves that introduce attributes contained in the guiding solution. This technique can be used in both diversification and intensification strategies, depending on the path generation mechanism and the choice of the initiating and guiding solutions.
- reactive TS²⁸³: (Battiti and Tecchiolli) It provides a technique which dynamically fits the search parameters based on the search history. The continuous reactive TS is a generalization of the reactive TS developed by the same authors. The continuous reactive

TS method tries to locate most promising boxes. In a second step, refined solutions are generated within these boxes.

- hybrid TS: It was developed by Al-Sultan and Al-Fawzan²⁸⁴. It represents a hybrid method that combines TS with the local search method of Hooke and Jeeves²⁸⁵.
- probabilistic TS³⁵: It provides a selection scheme of the moves which is based on elite solution recovery. It also introduces randomization into the search method. This can be attained by converting the evaluation and the references to tabu status into probabilities of selection, strongly made over to favour higher evaluations. Such randomization can also be implemented in the design of a candidate list strategy. It only considers a subset of neighbourhood solutions, because a complete neighbourhood evaluation is often too expensive.
- directed TS²⁸⁶: It uses direct-search-based strategies. These strategies are based on the well-known Nelder-Mead method²⁸⁷ and a new pattern search procedure called adaptive pattern search. The role of direct search methods is to make the search robust especially in the neighbourhood of a local optimum. Instead of using completely blind random searches for the generation of the neighbourhood, appropriate direct search strategies are exploited. Moreover, a TL conception with new anticycling rules called tabu regions (TR) and semi-TR were introduced.
- unified TS²⁸⁸: It aims to produce simpler and more flexible TS codes. The major benefits of the approach are its speed, its simplicity, and flexibility coupled with a dynamic adjustment of only a few parameters. A similar approach is the granular TS²⁸⁹, which is based on the use of drastically restricted neighbourhoods. They exclude moves with elements that are not likely to belong to good feasible solutions. Such restricted neighbourhoods are called granular, and may be seen as an efficient implementation of candidate-list strategies proposed for TS algorithms.
- TS for continuous multim minima problems²⁹⁰: Cvijovic and Klinowski^{291,292} generalized TS to functions with continuous variables but this approach also allows the optimization of integer values. Franzé and Speciale²⁹³ introduced an algorithm which explores a grid of points with a dynamically defined distance. Chelouah and Siarry²⁹⁴ presented a new algorithm called enhanced continuous TS. It results from an adaptation of combinatorial TS, which aims to follow Glover's basic approach as close as possible. To cover a wide domain of possible solutions, this algorithm firstly performs the diversification. It locates the most promising areas, by fitting the size of the neighbourhood structure to the objective function and its definition domain. When the most promising areas are located,

the algorithm continues the search by intensification within one promising area of the solution space.

There are some examples of TS application areas:

- general manufacturing problems²⁹⁵
- travelling salesman problem (TSP)^{296,297}
- quadratic assignment problem²⁹⁸
- general combinatorial problems^{24,299}
- optimization of artificial neural networks³⁰⁰
- resonance assignment in NMR (Nuclear Magnetic Resonance) spectroscopy³⁰¹
- test design in education³⁰²
- vehicle routing problems³⁰³
- chemical process optimization³⁰⁴
- identification of the good variable subsets within the framework of QSAR techniques³⁰⁵.

The above examples have shown that TS is a powerful algorithmic approach. It has been applied with great success to many difficult problems and has now become an established optimization technique which can compete with almost all known techniques. It can even exceed many classical procedures due to its flexibility. A great advantage of TS is that, as in the case of all approaches based on a Local Search, it can quite easily deal with really complicated constraints that are typical for real-life problems. Local search algorithms move from solution to solution in the search space until a solution considered as optimal is found or a time bound is elapsed. This ensured that it is a really expedient, but not a universal approach. Significant problem knowledge is absolutely required to perform the most basic steps of the development of any TS procedure, viz. the choice of the search space of an effective neighbourhood structure and the moves between the solutions. The fine tuning of an apparently large collection of parameters is also a problem. Consequently, this choice must be sufficient. To be successful, all metaheuristics need to explore the search space “in depth” and “in width”. Depth is usually not a problem for TS, which commonly achieves sufficiently acceptable good solutions in an early stage of the search. In this respect TS is quite aggressive and effective, but width can be a hard problem. This disadvantage can only be removed by an effective diversification scheme.

1.3 Conclusions

The optimization aim is to find a discrete mathematical object that maximizes or minimizes an objective function specified by the user of the metaheuristic. Such discrete

mathematical objects called states and the search space are problem-specific. User usually provides the function to be optimized, called objective function, as a procedure that evaluates the function on a given state.

Many computer scientists point out that no metaheuristic is better than another when its performance is averaged over all possible discrete functions (No-free-lunch theorem³⁰⁶). Therefore, at best, a specific metaheuristic can be highly efficient only for classes of objective functions. However, all these imperfections are characteristic also for other methods of the global optimization as well.

Metaheuristics are best and generally used for non-differentiable objective function where one has no possibility to exploit analytical tools as Hessians and derivatives, or if the objective function has no analytically closed form (for instance, if it is only the output data of another algorithm process). But even for all other cases metaheuristics achieve very fast and very accurate solutions if techniques from classical optimization such as local search and gradients are implemented in the metaheuristic approaches.

Although many criticisms exist, metaheuristics are powerful algorithmic approaches which have been applied with great success to many difficult combinatorial optimization problems. As previously observed, many well-known metaheuristics seem to converge towards a combined structure. New opportunities for combining the merits of these methods appear in this unifying process leading to even more powerful search approaches for difficult combinatorial optimization problems.

Based on an analysis of foregoing metaheuristic algorithms which evaluated their strengths, their weaknesses, and their appropriateness for the solving of set of tasks the TS method was chosen to create improved approaches. It forces also the fact that in contrast to most of all metaheuristic optimization techniques TS are extremely effective for the standards of continuous nonlinear optimization that confirms by last method developments. Detailed descriptions of the new approaches are introduced in the next chapters.

Chapter 2 TS based methods

2.1 Gradient Tabu Search

In this work a modification of the TS method was developed, which for clarity will be called Gradient Tabu Search (GTS). Its main goal is to enhance the efficiency of the TS by adapting strategies developed to find minimum energy paths of the potential energy surfaces. It exploits the analytical first and second derivatives. The GTS uses gradients for a fast location of the next local minimum while second derivatives are used in the mildest ascent part. For an efficient implementation novel concepts had to be introduced for storage of previous moves. The GTS can be classified as a multi-start method. In contrast to least square fit procedures, which are widely used for such problems, the GTS allows to find solutions which do not represent the next local minimum.

2.1.1 Algorithm description

Three basic search procedures are used in the GTS method: the Local, the Modest Ascent, and the Diversification searches. In the first search procedure one of the Local Search methods or combinations of them are used to find the nearest local minimum. After determining a local minimum, the Modest Ascent Strategy is employed to escape from this local minimum and to move to the next one along the shortest way. Moreover, Tabu List, Tabu Region, and Tabu Direction restriction rules are applied to avoid return to recently visited solutions or to get caught in a local optimum. The Diversification Search (DS) is needed in order to jump to other areas of the solution space that may be more promising for the seeking global minimum.

Figure 2-1 shows the flowchart of the GTS. After the initialization one starts with a search for improved minima. This part consists of local searches for the next minimum using gradients and mildest ascent searches to escape to the next valley. Weighted Hessians are used for the mildest ascent search. In between an update of the solution vector and the TL is performed. If the solution does not improve after a number of iterations or if all neighbourhood solutions are already set tabu, the search for improved minima is aborted and the DS is performed to obtain new starting points. A pseudo-code of the GTS algorithm is given in Figure 2-2.

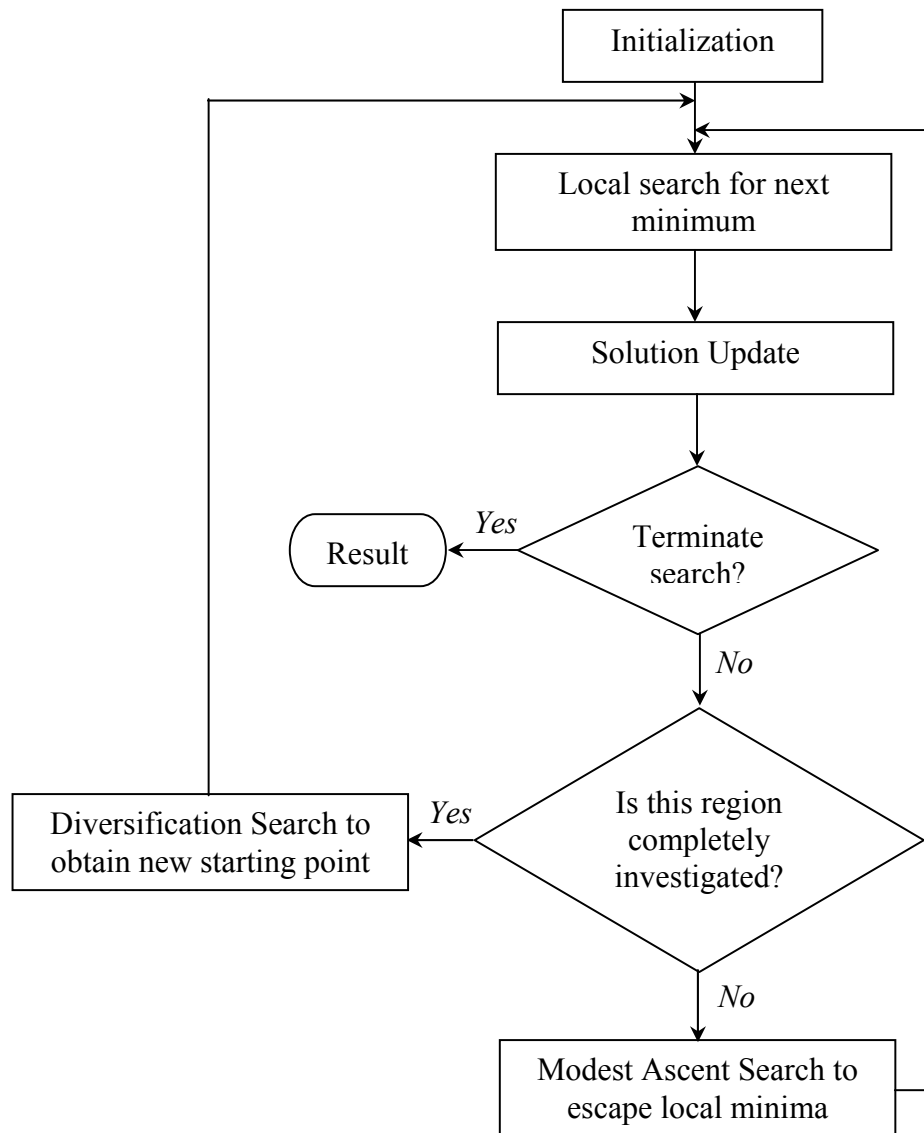


Figure 2-1. Flowchart of the new Tabu Search methods.

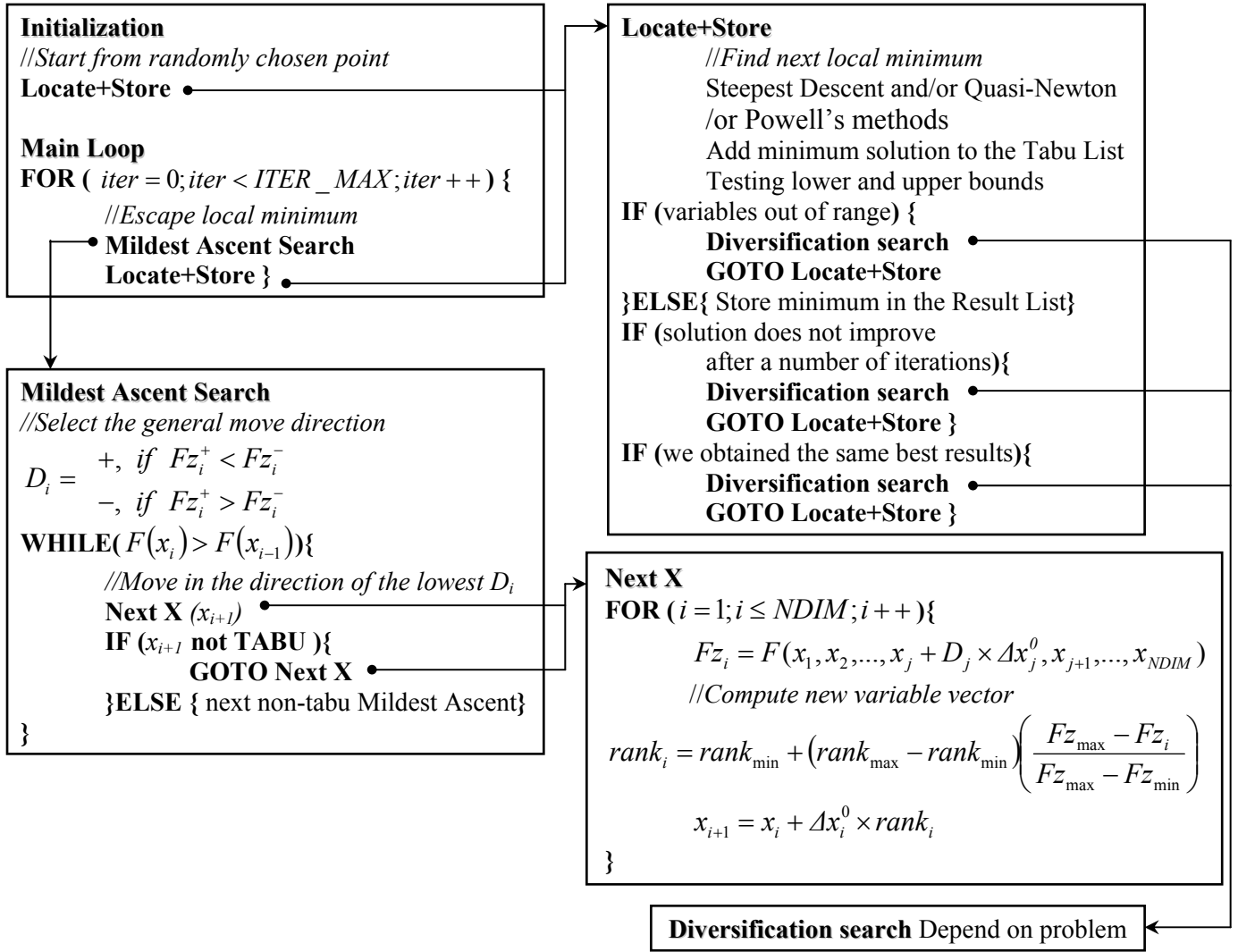


Figure 2-2. Pseudo-code of the Gradient Tabu Search algorithm.

2.1.1.1 Modest Ascent and Local Search Strategy

In the GTS, the definition of the neighbourhood is crucial. For a point $\vec{x} = (x_1 \dots x_i \dots x_N)$ the neighbourhood is often defined by those points for which all x_i are increased by a given Δx_i . From this neighbourhood, the GTS either selects the move that leads to the largest decrease in $F(x_i)$ or accepts the one that leads to the mildest increase in $F(x_i)$. A reduction of the number of steps is obviously obtained if within one step all x_i 's are appropriately modified and if computations of the sizes of the Δx_i 's take into account the shape of $F(x_1 \dots x_i \dots x_N)$. Both improvements can be achieved if gradients and/or second derivatives are employed.

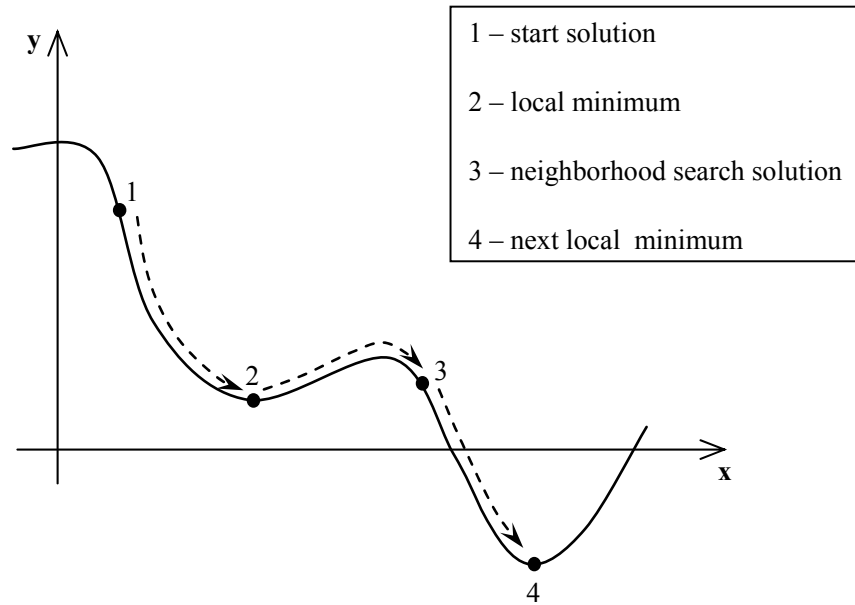


Figure 2-3. Schematic search strategy illustration.

The rough search scheme is presented on Figure 2-3. Moves from the solution 1 to 2 and 3 to 4 are carried out by Local Search methods, which are listed above, whereas displacement 2 to 3 is realized with the Modest Ascent strategy.

The Steepest Descent and the Quasi-Newton method are known to be very efficient for local optimization problems. Details of the both methods are described in Ref. 22-25. Steepest Descent is an algorithm for finding the nearest local minimum of a function which presupposes that the gradient of the function can be computed. This method also called the Gradient Descent method, starts at a point x_0 and, as many times as needed, moves from x_i to x_{i+1} by minimizing along the line extending from x_i in the direction of $-\nabla F(x_i)$, the local downhill gradient. The problem with the Steepest Descent method is that it performs many extremely small steps in going down a long, narrow valley. Quasi-Newton method uses the Quasi-Newton Broyden Fletcher Goldfarb and Shanno (BFGS) approximation to built up the Hessian updates based on past steps. There are many variants of Quasi-Newton methods. All of them approximate the Hessian matrix from the function and gradient values of some or all of the steps previously taken. In the present work the BFGS method was used, in which the model is not stored explicitly, but is calculated by gradients and step directions stored from the past steps. The Quasi-Newton method was chosen because it converges typically quite fast and does not require computation of the Hessian matrix, which can be quite expensive both, in terms of the symbolic computation and numeric evaluation. The disadvantage of the method is that it may diverge away from the minimum

To reinforce the advantages and minimize the disadvantages a combination of both methods was implemented in the GTS to locate the next local minimum. The search starts with the Steepest Descent approach. When the gradient value becomes small enough the program switches to the Quasi-Newton method. Using gradients reduces the number of steps necessary to locate the next local minimum. Another advantage is that many terms which need to be computed can be reused for the calculation of the gradients and Hessians. This accelerates the computation in comparison to approaches which solely depend on function evaluations.

It is well known that leaving the valley of the already found local minimum or the determination of a transition state is considerably more difficult than finding the next local minimum³⁰⁷. In so called eigenvector-following techniques³⁰⁸ eigenvalues and associated eigenvectors of the Hessian are employed to solve this problem. This method evaluates the mildest ascent exactly by computation and by diagonalization of the complete Hessian. The necessary effort to compute the Hessian increases roughly with $NDIM*(NDIM+1)/2$ ($NDIM$ being the dimensionality of the problem) and additionally a $NDIM*NDIM$ matrix has to be diagonalized. Therefore, such vector-following techniques could become more demanding than strategies which are based on less expensive function evaluations. In order to reduce the costs our approach only uses the diagonal elements of the Hessian to determine the direction of the mildest ascent.

Figure 2-4 illustrates the connection between the objective function, its gradient and second derivative in a local optima. The value of a second derivative gives the convexity or concavity of the function on the given interval. At a local minimum the first derivative is equal to zero, while the second derivative is positive, since the function is convex at this point. The diagonal part of the Hessian provides information along which coordinate the function is less convex. To determine the direction the functional values of each coordinate around the local minimum were computed at the beginning of the modest ascent strategy.

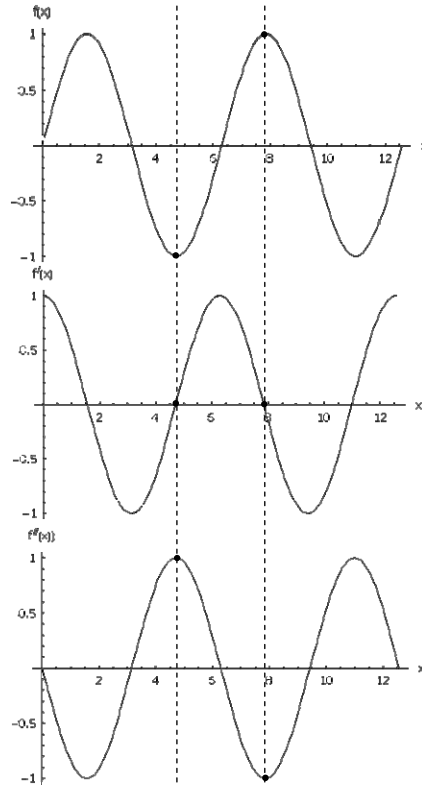


Figure 2-4. Connection between objective function $\sin(x)$, gradient and second derivative.

Figure 2-5 illustrates an example of the direction choice. The local minimum is at the solution (3.14; 4.438). The partial second derivative of F with respect to y with value 0.5005 is less than the partial second derivative of F with respect to x which equals 1.0005. It means that the function is less convex along the y -coordinate so the modest ascent should go either in the positive or in the negative direction along this coordinate according to the calculated function values.

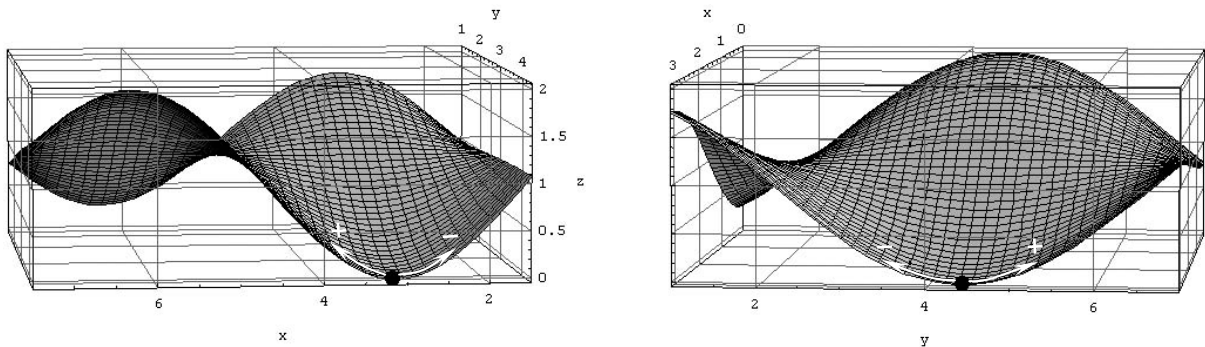


Figure 2-5. Griewangk function. Modest ascent strategy illustration,
where local minimum • $-(x;y)=(3.14; 4.438)$,

$$\frac{\partial^2 f}{\partial x^2} \approx 1.0005, \quad \frac{\partial^2 f}{\partial y^2} \approx 0.5005, \quad \curvearrowright^- \text{ negative move direction, } \curvearrowright^+ \text{ positive move direction.}$$

Thus, in order to escape from local minima via the modest ascent it is necessary to determine a new solution along a direction of the smallest function value with respect to information about partial second derivative. A schematic illustration of the idea used to determine the

direction of the modest ascent from the diagonal part of the Hessian is presented on Figure 2-6.

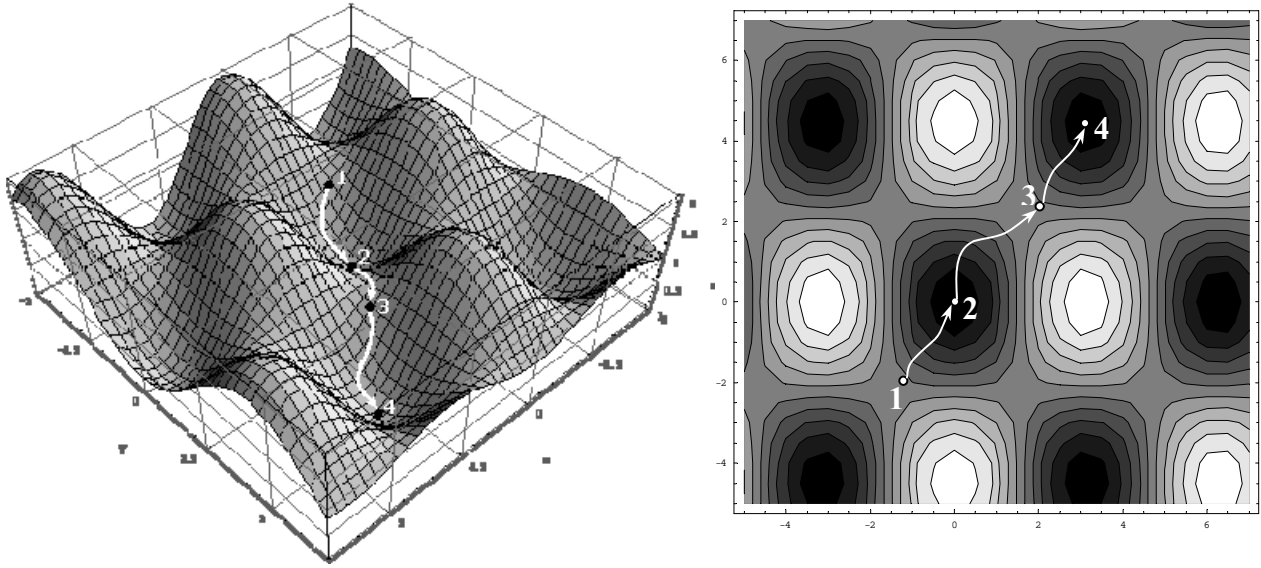


Figure 2-6. Griewangk function. Schematic illustration of moving from one local minimum to another, where 1 – start solution, 2 – local minimum, 3- neighbourhood solution, 4 – next local minimum and → - schematic movement direction.

The values for the single Δx_i 's are determined according to the following formula

$$x_i^{new} = x_i^{old} + \Delta x_i^0 * rank_i$$

$$rank_i = rank_{min} + (rank_{max} - rank_{min}) \left(\frac{\frac{\partial^2 F}{\partial x_{max}^2} - \frac{\partial^2 F}{\partial x_i^2}}{\frac{\partial^2 F}{\partial x_{max}^2} - \frac{\partial^2 F}{\partial x_{min}^2}} \right) \quad (7)$$

The value for Δx_i^0 is user defined and can be tuned for the problem under consideration. The actual step size is determined from Δx_i^0 and $rank_i$, which is computed from $rank_{min}$ and $rank_{max}$. Variable with smallest partial second derivative value has the largest step size while step sizes for all others variables are calculated with respect to partial second derivative value ranks. The second derivative values are indexed before ranking giving the minimal element the index 0 and the maximal one the index $(NDIM-1)$, where $NDIM$ is the dimension of the problem. A linear ranking procedure gives the minimal list element the maximum ranked value $rank_{max}$ and the maximal list element the minimum ranked value $rank_{min}$. The parameters $rank_{min}$ and $rank_{max}$ could also be used as fine tuning parameters but considering their definitions the most meaningful value for $rank_{max}$ is 1. Using $rank_{min} = 0$, the direction possessing the smallest second derivative would not contribute to

the overall direction of the next step. Therefore $rank_{min} = 0.1$ is used. $\partial^2 F / \partial x_{max}^2$ and $\partial^2 F / \partial x_{min}^2$ represent the highest and lowest second derivative at the point under consideration.

If the differences between the second derivatives become very small, this ranking scheme leads to values too small for the higher second derivatives. Therefore, in cases with $\partial^2 F / \partial x_{max}^2 - \partial^2 F / \partial x_{min}^2 < NDIM \cdot rank_{min}$ is redefined as

$$rank_{min} = rank_{max} - \left(\frac{\partial^2 F / \partial x_{max}^2 - \partial^2 F / \partial x_{min}^2}{NDIM} \right) \quad (8)$$

This modest ascent strategy is followed until a positive second derivative indicates that the barrier to the next valley is crossed. From this point, the next local minimum is located using the minimization strategy explained earlier.

This approach lies in between the eigenvector-following techniques and the so-called low mode search (LMOD)³⁰⁹. The latter is often used for conformational analysis. In this approach, eigenvalues and eigenvectors of the Hessian are computed at the minimum. To find a new minimum, the LMOD-procedure follows the eigenvector of one of the low-lying frequency modes until the increase in functional value exceeds a user-defined threshold. During this following, the Hessian is not updated. A minimization starting from this point leads to new minima in most cases. The gradient-only algorithm³¹⁰, which was developed to find saddle points, was also tested. This procedure uses the gradients to follow the weakest ascent along a valley to the next saddle point. The procedure was successful in some test cases but turned out to be extremely dependent on the chosen parameters. Therefore, it was not adopted.

2.1.1.2 Diversification Strategy

The last element of the GTS represents the diversification search (DS), which is used to select new starting points in the solution space. A DS becomes necessary if the solution does not improve after a number of iterations that means that neighbours at hand are worse than the current solution or if all neighbourhood solutions are already set tabu. One way to enhance the convergence is to ensure that the search switches to regions that were not already investigated. To explore the whole search domain, and localize promising areas several strategies can be chosen.

The standard algorithm starts from the N initial solutions (N being the number of parameters) which are randomly generated and uniformly distributed in the solution space. To

avoid returning to already visited areas and to force the search towards solutions sufficiently far away from previous ones all solutions are excluded which already belong to Tabu Regions (TRs). The objective function to be minimized is evaluated for each accepted solution, and the best of them (with the lowest $F(x_i)$ value) becomes the new start solution, even if it is worse than the previous solution.

In a second variant $3N$ solutions are randomly generated. $2N$ solutions are concentrated in two user-defined subspaces of the whole search domain (e.g. positive and negative areas if possible) and additional N points are randomly generated without any restrictions. The solutions are excluded which already belong to TRs as in the previous standard scheme. If a new promising solution with the smallest objective function value is accepted the exploration process starts inside this newly detected area of the promising solution. Depending on the problem Simulated Annealing methods may also be successful variants of the diversification search. However, for the present problems they turned out to be less efficient than the previously mentioned approaches.

2.1.2 TS Memory Elements

The basic idea of the approach is to keep sufficient information of the search within some memory. In the Tabu List (TL) either complete moves or single attributes are set tabu as soon as their complements have been part of selected moves. The moves or attributes stay tabu for a distinct time, i.e. for a number of iterations, until the probability that a solution is revisited is small. The efficiency of the algorithm depends on the choice of the tabu status duration. Using an effective memory conception GTS behaves as an intelligent search technique³⁵. Optimization search methods are divided into two categories of optimization search methods: point-to-point methods and population-based methods. GTS belongs to the first category.

Despite its efficiency at times, the TL seems to be rather limited approach. The standard TL can't avoid the zigzagging move situation near local optima during the descent method since the visited solutions in the TL don't show exact tabu direction. This is no longer sufficient since moves into already visited regions become allowed. To avoid such problems for the new approach new techniques as: Tabu Region (TR) and Tabu Direction (TD) were developed. These techniques enables a more efficient memory use then the standard TL, i.e. with equal memory volume more previous solutions can be set tabu and not only single visited solutions are set tabu but also areas around them and a complete visited route.

Reverse moves and cycles are avoided by the use of a Tabu List (TL), where the moves previously done are memorized. If a new current solution can be found, it is added to

the end of the list until it is full. Thereafter, the added solutions must replace an existing solution in the TL. The TL is managed with the First In First Out (FIFO) principle. The elements in TL are ranked in ascending order according to their recency. The most ancient element in the TL has index equal to 1, the most recent element has index equal to N , where N is the number of TL elements. If the TL is full, the newest solution replaces the tabu solution which has a smallest recency index that satisfies the requirements of the FIFO principle. Experiments with Multi-Ranked Tabu List²⁸⁶ and with modification of Multi-Ranked Tabu List, where also frequency ranking was added, were also carried out, but they offered no particular advantage over the traditional FIFO updating method. For approaches which move in small steps through the function $F(x_i)$ a simple storing of all visited points is sufficient to block already visited regions effectively. In the present approach this is no longer sufficient since the number of steps decreases while the step sizes increase. The specific property of the TL that is implemented in the GTS approach is that not all previously visited solutions are kept in memory but only the local minima solutions and start solutions from which they were found.

To overcome the resulting problems Tabu Regions (TR) and Tabu Directions (TD) are introduced.

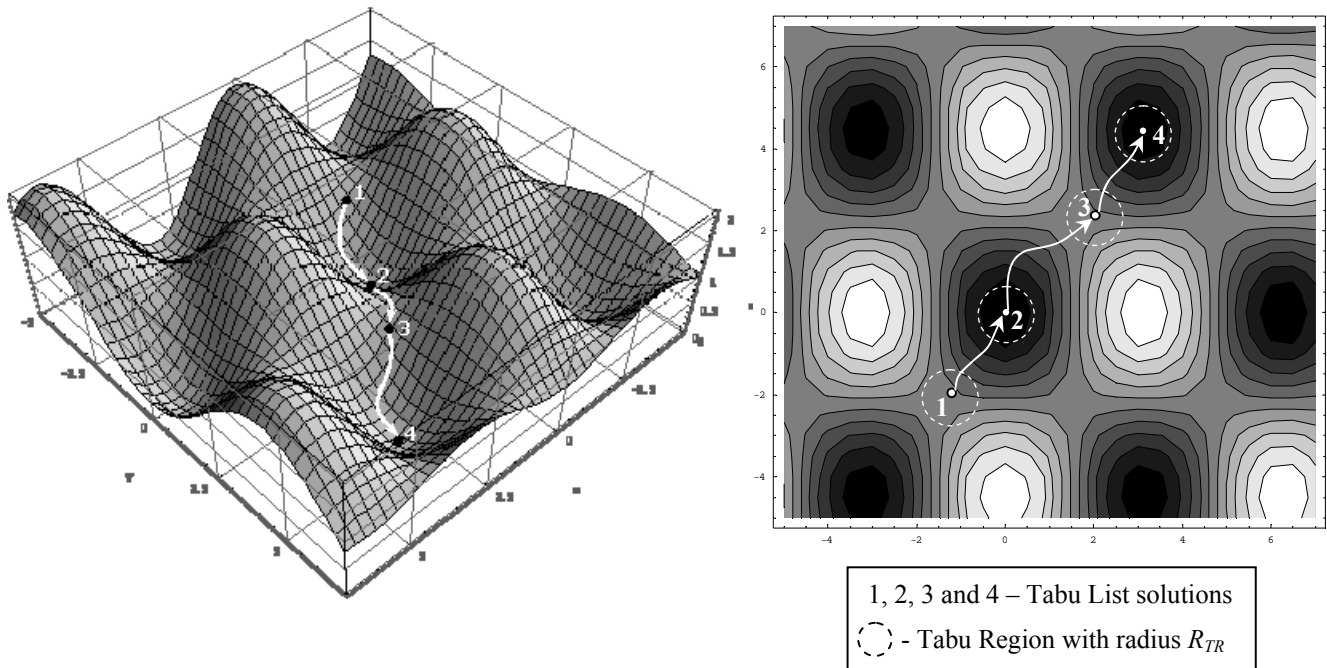


Figure 2-7. Tabu List solutions and Tabu Regions on the example of surface contour of the Griewangk function.

The conception of TD is closely connected with the Tabu Direction Vector (TDV) and New Move Direction Vector (NMDV) terms. To block already visited regions more effectively, the visited minima a_{min} are stored in the TL together with the starting points of the gradient

optimization from which the respective minimum was found (b_{start}). To ensure that the search for the transition state starting from the current minimum a_{min} does not move back, not only the already visited points but the whole direction is set tabu. The TDV and the NMDV are used for this purpose. The first vector, the TDV, is direction from the already visited local minimum solution a_{min} to the start solution b_{start} from which this local minimum was reached. The second vector, the NMDV, is the vector $\overrightarrow{a_{min}c_{new}}$, which connects a_{min} with the next trial point c_{new} . Figure 2-8 illustrates TDV $\overrightarrow{a_{min}b_{start}}$ and NMDV $\overrightarrow{a_{min}c_{new}}$ on the surface of the Rastrigin function.

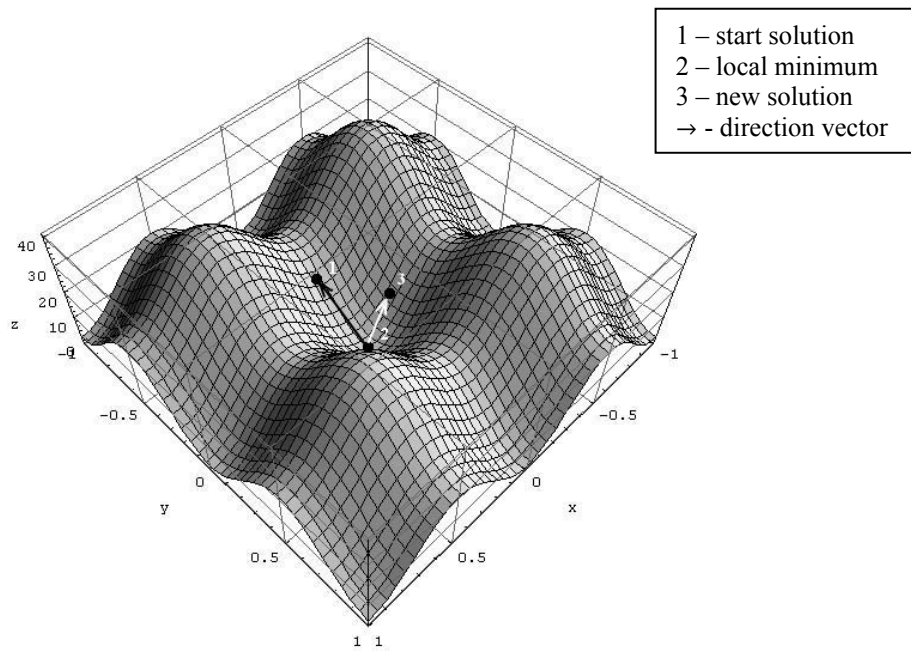


Figure 2-8. Surface of the Rastrigin function. Tabu Direction Vector $\overrightarrow{21}$ from the local minimum to the start solution and New Move Direction Vector $\overrightarrow{23}$ from the local minimum to the new solution.

To decide of the move to c_{new} both vectors (TDV and NMDV) are computed and normalized. If for the scalar product of both normalized vectors the equation

$$(1 - \alpha) < \overrightarrow{a_{min}b_{start}} \cdot \overrightarrow{a_{min}c_{new}} \text{ with } (\alpha > 0) \quad (9)$$

holds the new direction is tabu. Depending on α , which is given as input for fine tuning, the region which is set tabu varies. Visualization is given in Figure 2-9.

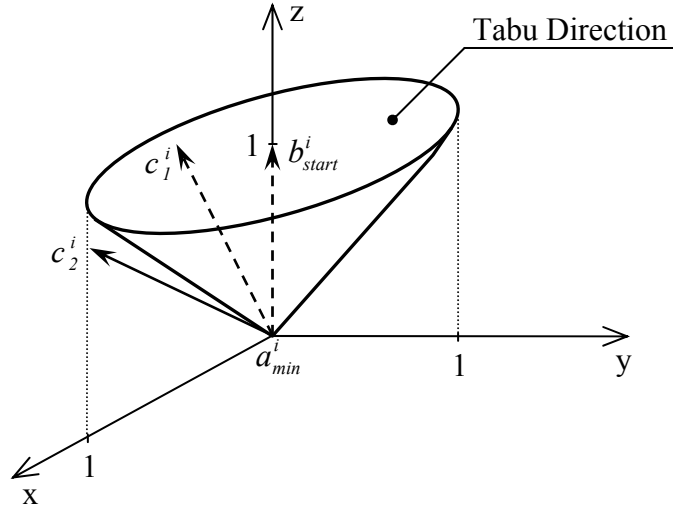


Figure 2-9. 3D geometrical interpretation of TD. For $\alpha=0.29$ the shaded part is tabu.

For $\alpha=0$ it is only forbidden to move exactly in the direction of $\overrightarrow{a_{min} b_{start}^i}$. With increasing α , a cone is becoming tabu. In Figure 2-9, α was set equal to 0.29. Here the solution c_1 is tabu, since the vector $\overrightarrow{a_{min} c_1^i}$ lies in the cone while solution c_2 is allowed.

Since only a few points are visited on the way from b_{start} to a_{min} it is also necessary to define Tabu Regions (TR) around those points for the diversification strategy. Figure 2-7 provides some examples of TL solutions and TRs. For this instance, a radius R_{TR} is defined for each point in the TL. The radii R_{TR} are computed according to

$$R_{TR} = \frac{(R_{up} - R_{low})}{coeff} \quad (10)$$

R_{up} and R_{low} define the range in which a parameter is allowed to vary. *coeff* is a user defined variable which accounts for the range in which $F(x_i)$ is defined and the density of local minima. The actual numbers employed for the test cases are given in Appendix A together with the test cases. Please note that the TR's are only needed for the diversification strategy.

2.1.3 Implementation and Experiments

The efficiency of the present implementation was investigated with the help of various test functions^{286,291,292,311,312,313,314,315,316,317,318}. The tests comprise the Ackley (AK_n), the Branin (BR), the Goldstein-Price (GP), the Griewangk (G_n), the Rastrigin (R_n), the Hansen (H), and the Levy (L_n) functions. The corresponding expressions are given in appendix A, which also reports the known global minima and the search ranges for each of the functions. A characterization of the computational efforts, which were necessary to determine the global

minima, can be seen in Table 2. As mentioned earlier, the GTS algorithm in its present implementation contains parameters. For a solid assessment of the success of the new algorithm their influence on the efficiency of the search has to be investigated.

Table 2. Characterization of the computational effort necessary to determine the absolute minima of the various test functions. All values are averages over 100 trial runs. For more information see text.

Test function	NDIM	Number of steps	Number of function evaluations	Number of first derivative calculations	Number of second derivative calculations
Rastrigin (R_n)	2	41.35	42.07	6.88	2.76
	10	191.64	143.95	29.98	61.09
	20	273.38	220.40	33.30	67.72
	30	347.59	293.36	34.23	69.15
	50	530.28	470.43	38.17	76.06
Ackley (AK_n)	2	155.81	138.4	63.54	24.71
	10	527.29	499.05	185.12	38.58
	20	839.96	793.17	214.9	62.14
	30	1009.73	949.12	227.52	80.74
	50	1522.36	1335.84	228.74	114.19
Griewangk (G_n)	2	1446.92	269.32	132.49	1236.36
	10	1490.95	944.21	669.85	568.78
	20	1978.73	1062.07	207.7	942.18
Levy (L_n)	4	5917.58	7166.64	4586.28	868.9
	7	5740.58	5716.27	4931.76	711.74
Branin (BR)	2	34	26.83	11.67	9.4
Goldstein-Price (GP)	2	56.38	49.27	16.13	10.08
Hansen (H)	2	563.86	387.52	117.46	211.17

2.1.3.1 Parameters

The parameters are summarized in the Table 3 together with brief explanations. Parameters as the dimension of the problem ($NDIM$), or the upper and low bounds of the search space (R_{up} , R_{low}) are determined by the concrete task. Taken the example given in the introduction (conformational search for large molecules) $NDIM$ would be the number of freely rotatable bonds and R_{up} and R_{low} would be the physically meaningful range in which they can be rotated. The parameters $rank_{max}$ and $rank_{min}$ are ranking parameters from the

linear ranking procedure [Eqs. 9 and 10]. They were set to 0.1 and 1.0 respectively, as discussed earlier. The parameters $Iter_{main}$, $Iter_{worst}$, $Iter_{loc}$, and $Iter_{MAS}$ are loop termination numbers. They restrict the number of allowed iterations but do not influence the convergence of the new algorithm. The number of elements in the TL ($LSIZE$) should not be set too small to avoid that the search spins around but with $LSIZE = 10 * NDIM$ no problems were found.

Table 3. Description of the parameters of the approaches.

Parameter	Purpose	Recommended values
<i>Parameters determined by the problem</i>		
$NDIM$	dimension	determined by the problem
R_{up}	upper bound for each variable (see appendix A)	
R_{low}	lower bound for each variable (see appendix A)	
$Iter_{main}$	loop termination numbers	
$Iter_{worst}$		
$Iter_{loc}$		
$Iter_{MAS}$		
<i>Parameters with negligible influence</i>		
$LSIZE$	number of elements in tabu list	$10*NDIM$
$rank_{max}$	maximum recency ranked value	1.0
$rank_{min}$	minimum recency ranked value	0.1
N_{trial}	number of trial points at the diversification search	$3*NDIM$
<i>Parameters with strong influence^a</i>		
$coeff$	coefficient used to compute the radius R_{TR} (Eq. 7)	calculates with respect to $R_{TR}=0.1$
$alam$	first step size at the line search	1.0
Δx_i	step size at the mildest ascent strategy	0.1
α	TD coefficient, used in formula (Eq. 8)	0.4

The parameters N_{trial} and $coeff$ are only connected with the diversification search. In our test cases, N_{trial} was always set to three times of the dimensionality of the problem ($3 * NDIM$), but not larger than 250. The parameter R_{TR} , which is computed with the help of $coeff$, R_{up} , and R_{low} (Eq. 7), controls the size of the TR within diversification runs. If R_{TR} is too large, the TR become so large that minima lying close by already visited points are overlooked since they lie in a TR. Naturally, this happens most frequently for problems with

^a investigated in Table 4 (for GTS) and Table 8, Figure 2-15, Figure 2-16 (for GOTS and TSPA)

many close lying minima. Since some information about the overall shape of the surface can easily be seen in a first scan, R_{TR} can be adjusted quite easily to the problem at hand. To ensure that no important minima are overlooked, one could make a test run with a quit small value of R_{TR} starting the search from the lowest minima found in previous runs. For the Rastrigin and the Ackley functions, the influence of R_{TR} is depicted in Table 4. It is seen that R_{TR} influences the convergence to some extent. It is also seen that if R_{TR} becomes too large, the absolute minimum is missed. Therefore, a smaller value for R_{TR} should be chosen. For the present cases, $R_{TR}=0.1$ was always chosen.

Table 4. Influence of the user-defined parameters for thirty dimensional functions observed during 10 trial runs.

Parameter	Value	Rastrigin		Ackley	
		Number of function evaluations	Success	Number of function evaluations	Success
R_{TR}	1.0	412	100%	1479	50%
	0.2	478	100%	1269	100%
	0.1	293	100%	949	100%
$alam$	2	277	100%	984	100%
	1	293	100%	949	100%
	0.2	302	100%	1330	100%
Δx_i	0.5	-	0%	1570	80%
	0.2	281	100%	1384	100%
	0.1	293	100%	949	100%
α	0.8	671	100%	1095	100%
	0.4	293	100%	949	100%
	0.1	218	100%	1011	100%

The influence of the fine tuning parameters $alam$, Δx_i , and α , which directly influences the search part, is also depicted in Table 4 using the Rastrigin and the Ackley function (each with a dimension of 30) as examples. The parameter $alam$ represents the first step size in the line search performed in the Steepest Descent and Quasi-Newton part. For small values, the local minimum search becomes more accurate, but more steps are necessary. Additionally, for a larger step size narrower and shallower minima are missed. This is no disadvantage, since the deeper minima is of main interest. Decreasing $alam$ from 2 to 0.2 the absolute minima are

always found but the necessary number of function evaluations increases (8% for Rastrigin, 35% for Ackley). As expected, the correlation between the number of steps and the value of $alam$ is very similar for all cases. Although not optimal for all test computations discussed $alam = 1.0$ (full Newton step) was used for all of them. The parameter Δx_i gives the step size during the mildest ascent search. Its influence is not surprising, since the search for a saddle point is a quite delicate task. One would expect that the number of steps necessary to reach the global minimum increase if a smaller step size, Δx_i , is chosen. But this is only found for the Rastrigin function. Additionally, while for increasing $alam$ the global minimum was always found, this is not the case for Δx_i . Therefore, Δx_i should not be set too large, since the next saddle point could be missed. For the following test cases Δx_i was always set to 0.1.

The parameter α ($0.0 < \alpha \leq 1.0$) determines the size of the cone of the TD (Eq. 8, Figure 2-9). One would expect that the number of steps decreases if a wider cone is set tabu; however, the opposite is found. This unexpected behaviour results since large cones restrict the flexibility of the search, i.e. the search must take detours to the global minimum since more direct ways are set tabu by accident. This is in line with the finding that for quite large cones, the minimum is not found anymore. For all test cases discussed later, α was set equal to 0.4. Again, this value was not optimized by preliminary test runs.

In summary, Table 4 indicates that the new approach is quite stable with respect to the actual values of the fine tuning parameters. In a wide range, it finds the global minimum also for nonoptimal fine tuning parameters; only (for some functions) the number of necessary steps increases.

2.1.3.2 Numerical Results

Table 2 summarizes the computational efforts, which were necessary to determine the global minima of the various test functions. In most real life applications of optimization routines, it remains unclear if the global minimum is actually found. Therefore, most optimization routines possess stop criterions, which control when a diversification is made and when the program stops. In the present implementation, the program performs a diversification run if after three new local minima no better solution was found. The program finally stops after $Iter_{main}$ runs were performed. For the test cases for which the global minima are well known, Table 2 lists the number of steps that were made until the global minimum was reached for the first time. This number was determined by an analysis of the run. For the computation double precision variables were used. The global minima were always reached within numerical accuracy.

Table 5. Comparisons with the literature for the Rastrigin function. The column “Evaluations” gives the function evaluations. For the present work it also comprises the sum of function, gradient and Hessian computations as second number.

NDIM	Algorithm	Evaluations	Source
2	Pure random search	5964	315
	Fast annealing evolutionary algorithm	544	314
	Random tunneling algorithm	383	314
	TRUST	59	314
	Tabu Search	540	291
	Differential evolution	489	324
	Random search	610	324
	Simulated Annealing	260	324
	Gradient Tabu Search	42/61	333
10	Tabu Search-MS 400	11624	316
	Tabu Search-REM	1981	316
	Genetic Algorithm	4488	319
	Gradient Tabu Search	143/1053	333
20	Genetic Algorithm	>100000	321
	Cooperative Coevolutionary GA	20000	321
	Tabu Search-MS 200	60446	316
	Tabu Search-REM	7961	316
	Parallel Genetic Algorithm	9900	322
	EASY Genetic Algorithm	6098	323
	Breeder Genetic Algorithm	3608	323
	Gradient Tabu Search	220/2240	333
30	Gradient Tabu Search	293/3394	333
50	Gradient Tabu Search	470/6181	333

The number of variables of the various functions was varied to study the behaviour of the new approach with respect to the dimensionality of the problem. In each case, the present approach found the absolute minimum. Naturally, the number of steps needed to determine the absolute minimum depends on the starting value. To obtain statistical results, for each case (function and dimension of the problem), 100 independent trials were carried out. Each trial corresponds to a new starting solution, chosen randomly within the search space.

For the Rastrigin function with two variables, the present approach needs about 42 steps to find the global minimum. Within the search, about 42 function evaluations and seven first derivative evaluations are performed. The Hessians are computed about 3 times. Please note that in each first and second derivative evaluation, a vector of the dimension of the problem has to be computed. As expected, the number of steps increases with the dimension of the problem; however, the increase is considerably less than linear. If the dimension is increased by a factor of 25, the number of steps increases only by a factor of about 13. If the number of variables is below 30, the wall clock times needed to determine the global minimum are well below 1 s. For 30 and 50 variables, 1–2 s were needed.

Table 6. Comparison with the literature for the Ackley function. The column “Evaluations” gives the function evaluations. For the present work it also comprises the sum of function, gradient and Hessian computations as second number.

NDIM	Algorithm	Evaluations	Source	Wall clock time [sec]
2	Random search	1257	324	3.3
	Differential evolution	311	324	0.4
	Gradient Tabu Search	138/315	333	<<0.1
10	Differential evolution	4214	324	14.8
	Gradient Tabu Search	499/2736	333	<<0.1
20	Differential evolution	9957	324	80
	Gradient Tabu Search	793/6334	333	<<0.1
30	Genetic Algorithm	>100000	321	
	Cooperative Coevolutionary GA	50000	321	
	Breeder Genetic Algorithm	19420	323	
	Easy Genetic Algorithm	13997	323	
	Tabu Search-MS 100	22842	316	
	Tabu Search-REM	17941	316	
	Differential evolution	14957	324	222.7
	Gradient Tabu Search	949/10197	333	≈ 1
50	Differential evolution	22557	324	888.9
	Gradient Tabu Search	1336/18482	333	≈ 2

For the Ackley function, the new approach needs more steps to find the global minimum but the increase of the number of steps with respect to the dimensions is again considerably lower than linear. It is important to note that the number of first and second derivate computations even increases less than the number of function evaluations. The Levy function seems to be the most complicated case. To determine the global minimum for four variables, the present approach needs already about 5917 steps. This is considerably more

than it needed for all other test cases. For the Branin, the Goldstein-Price, and the Hansen functions, a similar behaviour as for the Ackley and Rastrigin function is obtained. For the Griewangk function, the number of second derivative calculations decreases considerably if one goes from 2 to 10 dimensions. Additionally, the number of steps increases quite slowly. This happens since during the gradient optimization, the search jumps over various local minima. It seems that these local minima are surrounded by quite narrow valleys. Nevertheless, the global minimum was found in each run.

Table 5 - Table 7 summarize comparisons with other optimization routines. An objective comparison of the computational efforts needed to find the global minima is only given by wall clock times taken on the same machine since the computational overhead for the steps varies from method to method. Since this is nearly impracticable, the number of function evaluation is normally taken for comparison. These are summarized in the column *Evaluations* of the tables. The data obtained within the present study were averaged over 100 trial runs, but the numbers are rounded to integer. To estimate the effort for the present approach it has to be considered that a gradient or Hessian evaluation comprises the computation of a vector with the size of the dimension of the problem. Therefore, for the present approach, this column also gives the sum of all computed elements (=number of function evaluations + gradient evaluations * dimension + Hessian evaluations * dimension) to get a better insight into the computational effort. This is upper limit for the effort since various terms already computed for the function evaluation can be reused for the analytical gradients and Hessians. Additionally to the numbers given in Table 5 - Table 7 the convergence of the present approach is also compared with convergence of the differential evolution algorithm (Figure 2-10, Figure 2-11).

The Rastrigin function was often employed to test the efficiency of optimization routines^{291,292,314,315,316,319,320,321,322,323}. On the basis of the number of function evaluations, the present implementation represents the most efficient algorithm if a Rastrigin function with two variables is used. Even if the number of evaluations used in the present approach includes gradient and Hessian computations only the Terminal repeller unconstrained subenergy transformation algorithm used by Jiang et al.³¹⁴ needs fewer evaluations. Figure 2-10 compares the convergences of the GTS and of the *Differential evolution* algorithm for the two dimensional Rastrigin function. The *Differential evolution* algorithm runs were performed with the Mathematica³²⁴. It clearly demonstrates the excellent convergence of the GTS, which shows hardly any oscillation. The objective function of *Differential evolution* algorithm decreases much slower. Additional quite a few oscillations can be seen.

Table 7. Comparison with the literature for the Branin, the Goldstein-Price, and the Hansen functions. The column “Evaluations” gives the function evaluations. For the present work it also comprises the sum of function, gradient and Hessian computations as second number.

Function	Algorithm	Evaluations	Source
BR	Tabu Search-CJ	492	291
	Pure random search	4850	
	Simulated Annealing 1	2700	
	Multi-start	1600	317
	Simulated Annealing 2	505	
	Tabu Search	492	
	Paper work	144	
	TRUST	55	318
	Tabu Search-REM	192	316
	Tabu Search-MS20	333	
	Mod. Annealing	144	317
	Simulated Annealing	324	
	Random search	15	324
	Differential evolution	10	
	Gradient Tabu Search	27/69	333
GP	Simulated Annealing 1	5439	317
	Pure random search	5125	
	Multi-start	4400	318
	Tabu Search-REM	894	316
	Simulated Annealing 2	563	
	Tabu Search	486	318
	Mod. Annealing	486	
	Enhanced Continuous Tabu Search	231	290
	Directed Tabu Search	230	286
	Tabu Search-MS 500	195	316
	Paper work	112	317
	Tabu Search-CJ	112	291
	TRUST	103	318
	Gradient Tabu Search	49/102	333
H	Random search	420	324
	Simulated Annealing	412	324
	Differential evolution	291	324
	Gradient Tabu Search	217/425	333

The test runs made for a 10 and 20 dimensional Rastrigin function allow a comparison with previous modifications of the TS^{291,316} and various genetic algorithms^{319,321,322,323}. The so-called Tabu Search-REM method³¹⁶ is much better than the TS-MS 400 approach³¹⁶; however, the number of evaluations needed by the present approach is about a factor of 2–3 less. The present method also outperforms the genetic algorithms. For even higher dimensional Rastrigin functions, no other results could be found.

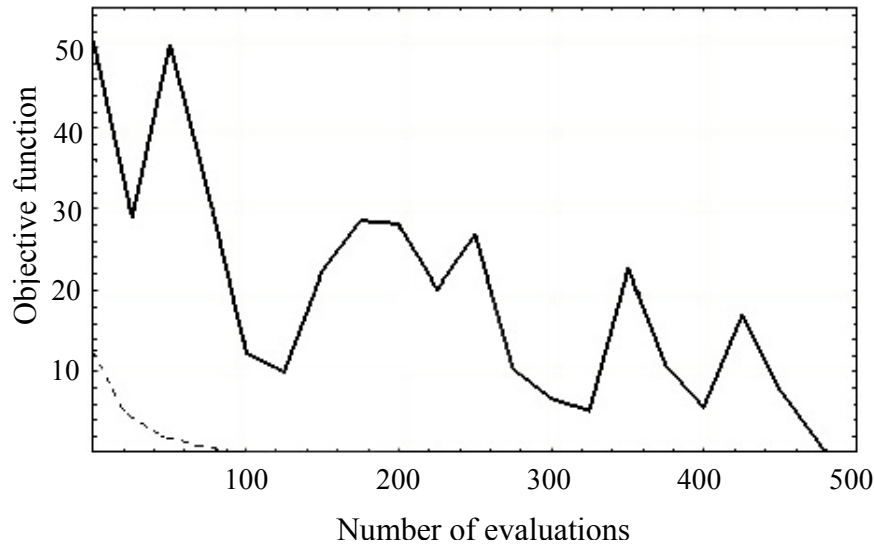


Figure 2-10. Comparison of the convergence of the algorithms for two dimensional Rastrigin function. The stars indicate the values of the objective function for the local minima which were found during the GTS.

The thirty dimensional Ackley function was used as test case for some genetic algorithms^{321,323} and improved TS approaches³¹⁶. To get more information, test runs were also performed with some algorithms implemented into the Mathematica 5.1 program package. These tests were performed with the *Random search*^{7,325,326,327} and the *Differential evolution*^{328,329,330, 331,332} approaches. While the former were only used for the two dimensional case, the latter were employed for all dimensionalities.

As already discussed the Ackley function seems to be a more difficult test case for the present approach since it needs considerably more steps to determine the global minimum of the Ackley function than for the Rastrigin function. Indeed, while for the Rastrigin function, the present approach outperformed nearly all methods if the number evaluations were compared, for the Ackley function it is still better than most other approaches but the differences are smaller. For thirty dimensions, the *Differential evolution* approach implemented in the Mathematica 5.1 needs 14957 function evaluations to find the global minimum. For the *easy genetic algorithm* proposed by Voigt³²³ 13997 function evaluations were necessary. With 10197 evaluations (number of function evaluations + gradient

evaluations * dimension + Hessian evaluations * dimension), the present approach needs only a slightly smaller number. Please note that the number of function evaluation (949) and also the approximated number of steps (1010) are much smaller. The graphical plot given in Figure 2-11 compares the number of evaluation and function evaluation of the GTS with the number of function evaluations used by the *Differential evolution* algorithm. Again, the convergence of the GTS is better even if the number of all evaluation of the GTS is compared with the number of the function evaluations of the Differential evolution algorithm.

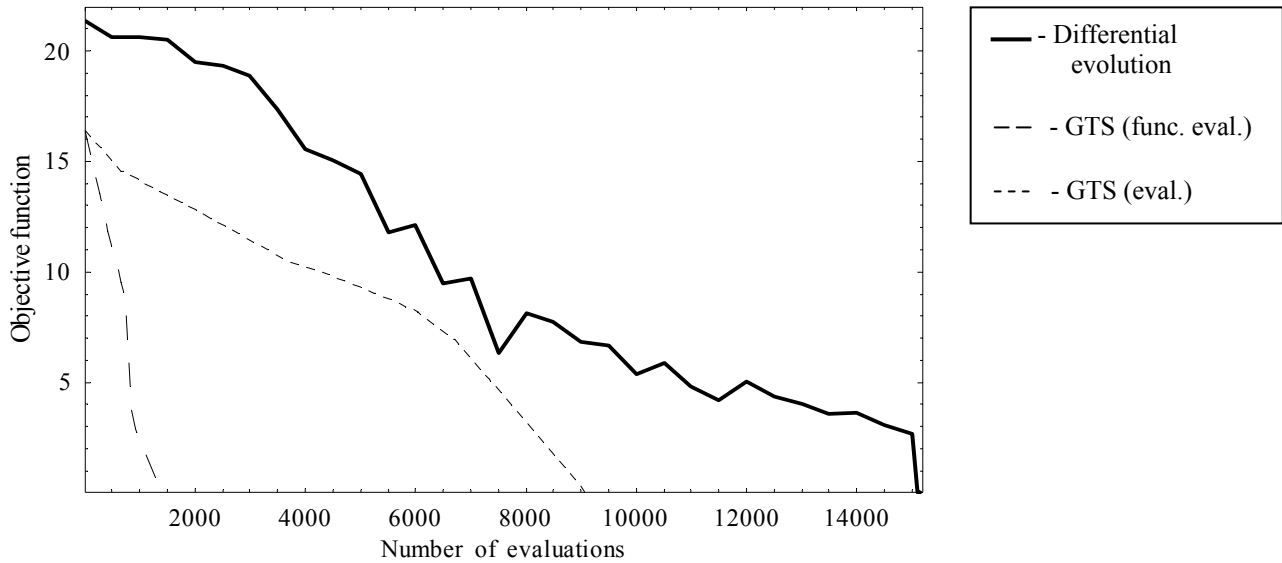


Figure 2-11. Comparison of the convergence of the algorithms for thirty dimensional Ackley function. The stars indicate the values of the objective function for the local minima which were found during the GTS.

While the present approach and the *differential evolution* ansatz taken from the Mathematica³²⁴ program package seem to be similarly efficient on the basis of the number of evaluations, if real wall clock times are compared, our approach seems to be much faster (1s in comparison to 222 s). This difference may result since for the present approach, the number of additional operations (beside the evaluations themselves) is quite small. Furthermore, it may result since many terms which appear in a function evaluation also occur in the gradients, i.e. they have to be computed only once. However, possible overhead within the Mathematica program, which represents a more general code for all kinds of computations, may also contribute. The present approach also seems to be more stable with respect to the parameters than the differential evolution ansatz. With the *differential evolution* ansatz sometimes the global minimum could not be reached at all. For some problems, this already happened with the standard parameters included in the Mathematica. In contrast, the present algorithm finds the global minimum also for nonoptimal fine tuning parameters.

For the Hansen (H) and the Goldstein-Price (GP) functions, which are also often used as test cases^{286,291,311,312,314,315,316,317,318}, a comparison between the various methods leads to

similar results as found for Rastrigin and Ackley functions. For the Branin (BR) function, the random search and the *differential evolution* show faster convergence than the GTS approach.

2.1.4 Conclusions

The GTS represents a metaheuristic, which tries to determine a global minimum of a function by the *steepest descent—mildest ascent* strategy. It combines algorithms developed to locate stationary points on quantum chemical hypersurfaces with the strategy of the TS. It uses a combination of the Steepest Descent and Quasi-Newton approach for a fast minimization to the next local minimum. To escape local minima via the mildest ascent, the diagonal elements of the Hessian are employed. For the steepest descent and the mildest ascent in each step, all variables of the function $F(x_i)$ are varied simultaneously. The actual direction is determined by a weighting procedure, which exploits the gradient vector elements and the diagonal elements of the Hessian. To ensure an efficient blocking of already visited regions despite the lower number of steps the GTS introduces Tabu Regions and Tabu Directions as elements for the Tabu List.

Test computations with the Ackley, the Branin, the Goldstein- Price, the Griewangk, the Rastrigin, the Hansen, and the Levy functions in up to 50 dimensions show that the new approach outperforms most previous approaches on the basis of the number of function evaluations. Comparison with the *differential evolution* ansatz proves the efficiency of the present approach on the basis of wall clock timings. For these tests, the *differential evolution* implementation of the Mathematica³²⁴ program package was used.

The present approach uses analytical first and second derivations. Consequently, its area of application is restricted to differentiable functions. Possible applications could lie for example in minimization routines for force field parameters or conformational searches with force fields.

2.2 TSPA and GOTS

In a preceding chapter the Gradient Tabu Search (GTS)³³³ was introduced. It uses analytical gradients for a fast minimization to the next local minimum and the diagonal elements of the analytical Hessian to escape local minima. For the minimization a combination of the Steepest Descent and the Quasi-Newton methods is used^{334,335,336,337}. To follow the modest ascent only the diagonal elements of the Hessian are employed but to determine the direction they are weighted by a linear ranking procedure. Moreover, novel concepts for the Tabu Directions (TD) were introduced to ensure an efficient blocking of already visited regions. The GTS can be classified as a multi-start method. Test computations

for well known functions showed that this new approach converges much faster than other methods taken from the literature. However, while the exploitation of gradients and Hessians speed up the convergence of the TS considerably especially the Hessian and in some cases also the gradients may be difficult to compute or are even not available. This can be tackled by the two new approaches which are given in the present chapter. The GOTS (Gradient Only Tabu Search)³³⁸ still uses the gradients to achieve a fast convergence to the next local minima but instead of the Hessian the approach uses a grid of function evaluations to follow the modest ascent. In the TSPA (Tabu Search with Powell's Algorithm)³³⁸ also the gradients are replaced by a grid of function evaluation.

These two new nonlinear global optimization routines are also based on the Tabu Search strategy which tries to determine the global minimum of a function by the *steepest descent - mildest ascent* strategy. The new algorithms are explained and their efficiency is again compared with other approaches by determining the global minima of various well-known test functions with varying dimensionality. These tests show in most cases that the GOTS possesses a much faster convergence than global optimizers taken from the literature. The efficiency of the TSPA method is comparable to that of genetic algorithms.

2.2.1 Description of algorithms

The search is started from a current solution $F(x_1, x_2, \dots, x_{NDIM})$, where $NDIM$ represents the dimensionality of the problem. A set of neighbours is generated by applying "moves" according to a GTS scheme. Combinations of the Steepest Descent and the Quasi-Newton methods³³⁵ which are known to be very efficient for locating the next local minimum were already successfully applied in the GTS approach and were consequently also adopted in the GOTS. As the GTS, the GOTS starts with the Steepest Descent approach and switches to the Quasi-Newton method if the gradient value becomes small enough. Since gradient may sometimes not be available an approach solely based on function evaluation was also developed. In the present implementation the Direction Set Method with Powell's Algorithm³³⁷ was used, whose choice of successive directions does not involve explicit computations of the function's gradient. The main idea of all Direction Set methods is using the some instructions for updating the set of directions as the method proceeds. This set either includes some very good directions that leads far along narrow valleys, or includes some number of "non-interfering" directions with the special feature (more conventionally called conjugate directions) that minimization along one direction is not "damaged" by subsequent minimization along another one. In the case of a set of linearly independent, mutually conjugate directions a boundless cycling through the set of directions can be avoided. Then,

one pass of line minimizations will put it exactly at the minimum of a quadratic form function. For functions that are not exactly of quadratic form, it does not exactly lead to the minimum; but repeated cycles of line minimizations in due course converge quadratically to the minimum.

To reduce the costs the previously developed GTS only exploits the diagonal elements of the Hessian to estimate step size and direction according to a linear ranking procedure. In the GOTS and TSPA only function evaluations are used to estimate step size and directions to reduce expenses concerned with necessity of Hessian calculation. For the first step of the modest ascent all functional values Fz_i^+ and Fz_i^- are computed to determine the general direction of the following moves. They are obtained by varying each dimension by a user defined step size Δx_i^0 :

$$\begin{aligned} Fz_i^+ &= F(x_1, x_2, \dots, x_j + \Delta x_j^0, \dots, x_{NDIM}) \\ Fz_i^- &= F(x_1, x_2, \dots, x_j - \Delta x_j^0, \dots, x_{NDIM}) \end{aligned} \quad (11)$$

The most advantageous direction for each variable is then determined according to:

$$D_i = \begin{cases} +, & \text{if } Fz_i^+ < Fz_i^- \\ -, & \text{if } Fz_i^+ > Fz_i^- \end{cases} \quad (12)$$

The functional values Fz_i needed for the linear ranking procedure and for further correction of the direction are then computed with

$$Fz_i = F(x_1, x_2, \dots, x_j + D_j \times \Delta x_j^0, \dots, x_{NDIM}) \quad (13)$$

Then step size and direction are estimated according to

$$\begin{aligned} rank_i &= rank_{\min} + (rank_{\max} - rank_{\min}) \left(\frac{Fz_{\min} - Fz_i}{Fz_{\max} - Fz_{\min}} \right) \\ x_{i+1} &= x_i + \Delta x_i^0 \times rank_i \end{aligned} \quad (14)$$

Please note that the D_i 's are only needed for the first step of the modest ascent part since the general direction is clear afterwards. Figure 2-12 illustrates the choice process of the most promising direction for each variable and moving with the ranked steps that gives the possibility to discard the unpromising solutions and decreases the number of function evaluations needed to escape the optimality valley.

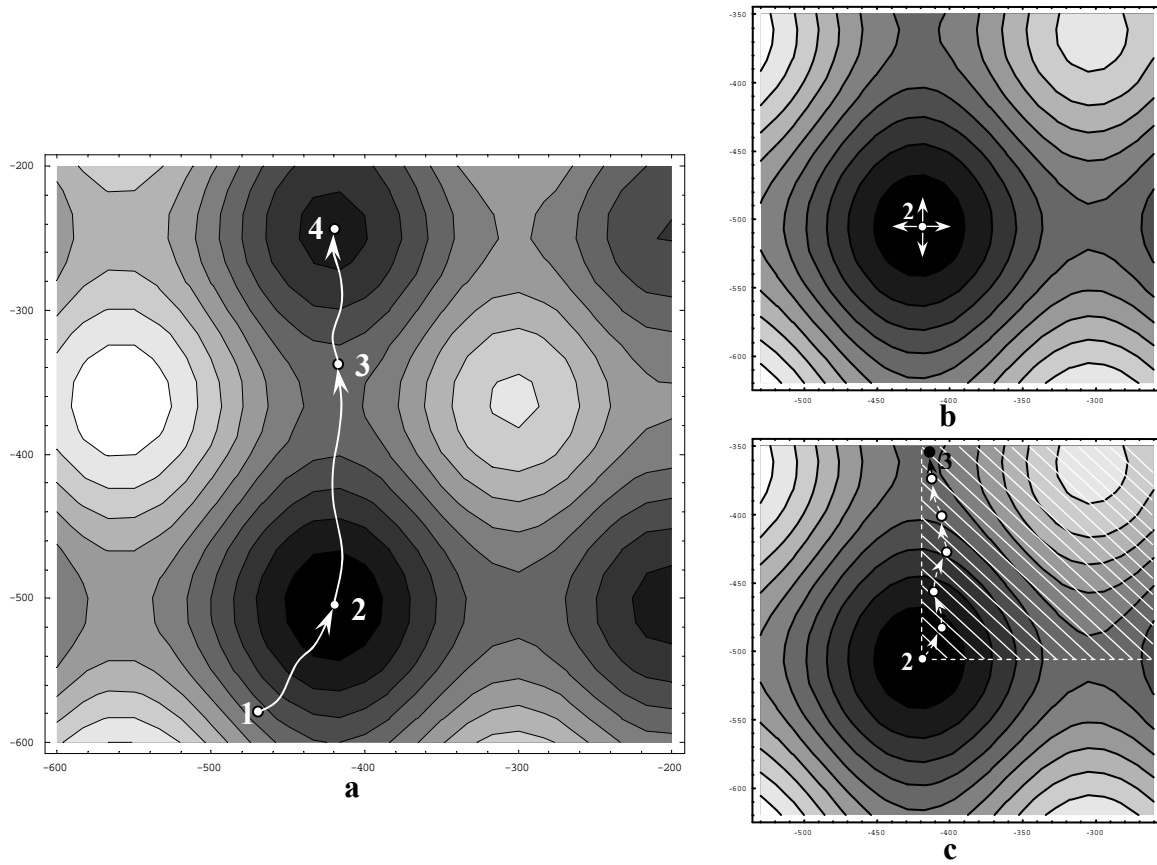


Figure 2-12. Modest ascent strategy of GOTS and TSPA on contour plot of the Schwefel function. a) the search path: start solution 1 → local minimum 2 →neighbourhood solution 3→next local minimum 4; b) computation of the functional values around the local minimum at the beginning of the modest ascent strategy; c) moving with the ranked steps within the promising area.

In Eq. 14 $rank_{max}$ and $rank_{min}$ can be user defined parameters but are normally set to 0.1 and 1. Fz_{max} and Fz_{min} are the maximal and minimal Fz_i values, respectively. If the differences between maximal and minimal function values become too small this ranking scheme leads to too small values for the larger factors $rank_i$. Therefore, in cases with $Fz_{min}-Fz_{max} < NDIM$, $rank_{min}$ is redefined as

$$rank_{min} = rank_{max} - \left(\frac{Fz_{min} - Fz_{max}}{NDIM} \right) \quad (15)$$

This strategy is followed until the new calculated function value is smaller than the previous one which indicates that the barrier to the next valley is crossed. From this point the next local minimum is located using the minimization strategy explained above.

To avoid reverse moves and cycles within the search the TL, the TR, and the TD as it shows Figure 2-13 are used which were already successfully employed in the GTS method³³³ and which related to ideas introduced by F. Glover³³⁹. The TR and the TD are defined as in the GTS method (Eqs.7, 8).

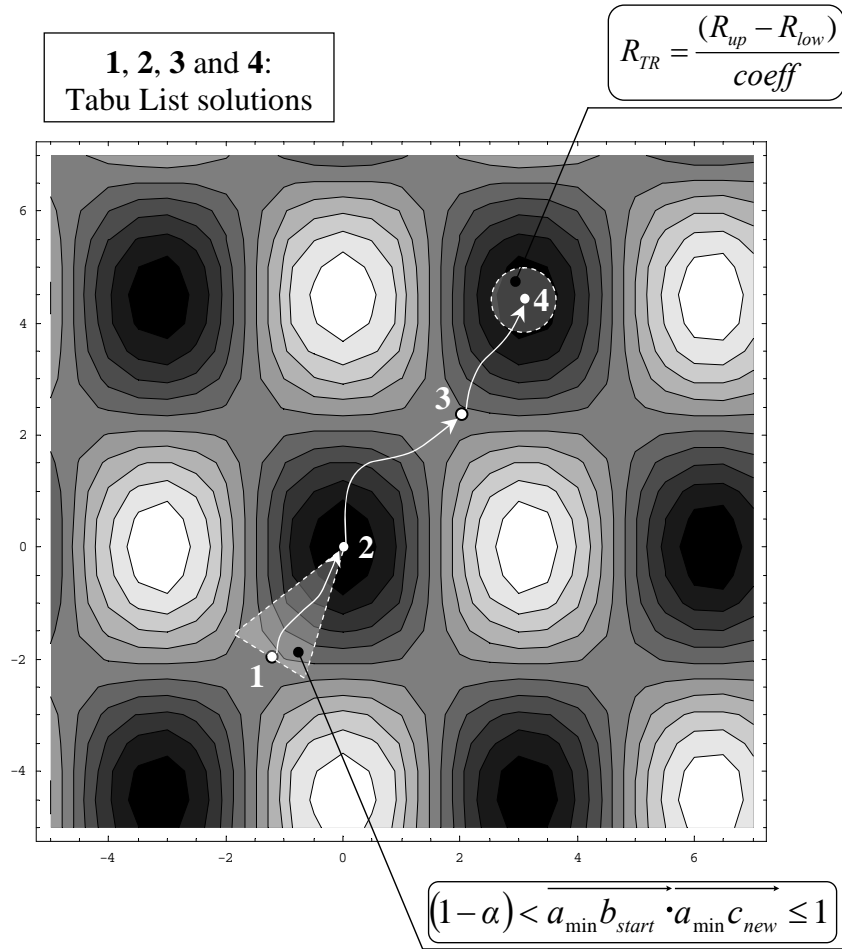


Figure 2-13. TL, TD, TR.

The last element of the GOTS and the TSPA represents the DS for the detection of promising areas to select new starting points in the solution space. For GOTS and TSPA diversification is performed in the same way as in GTS.

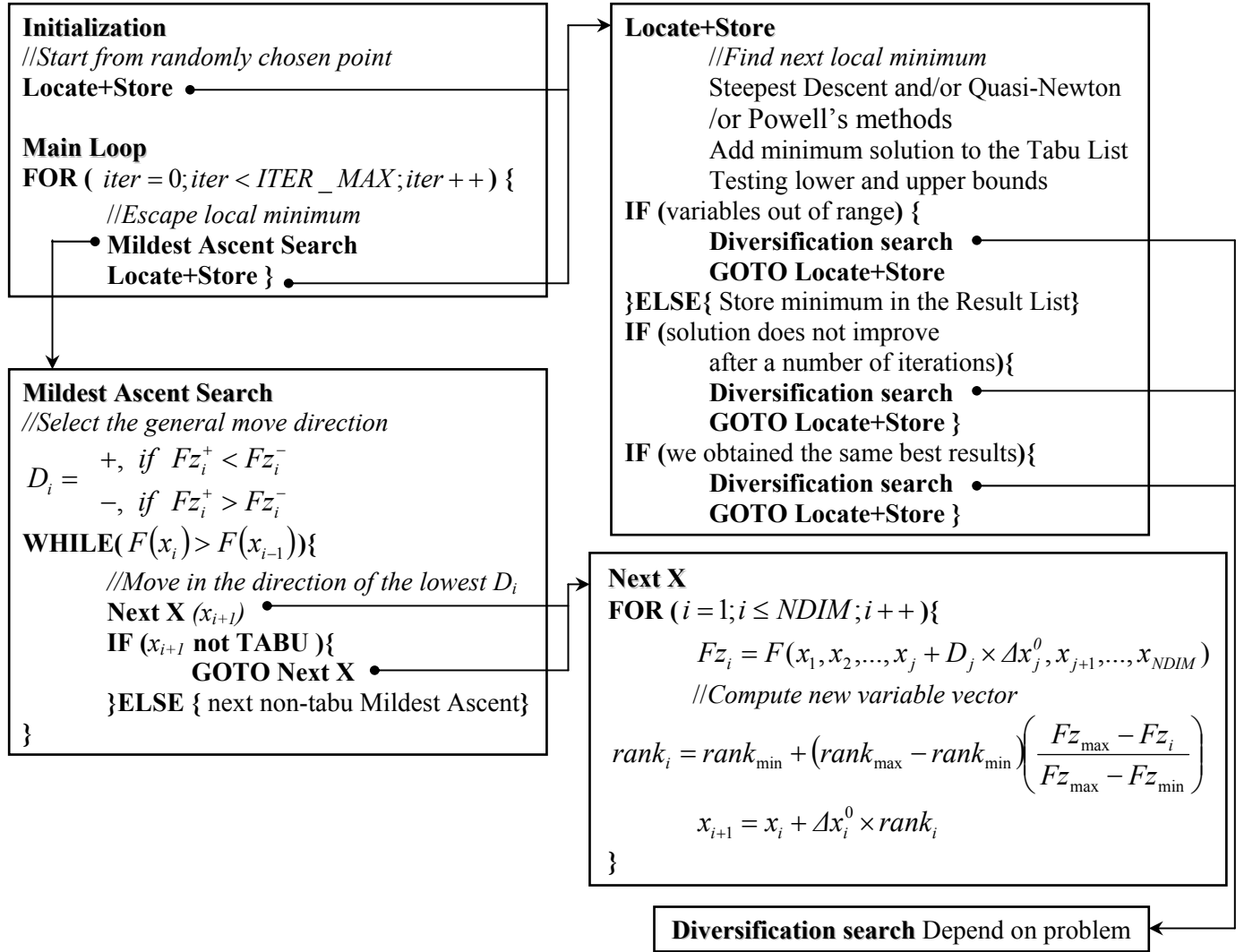


Figure 2-14. Pseudo-code of GOTS and TSPA.

The flowchart of both methods is given in Figure 2-1. Sometimes it is efficient to use DS as an initialization unit for reasons given above. After the initialization one starts with the search for improved minima. This part consists of local searches that are applied to obtain the nearest local minima and the modest ascent searches used to escape to the next valley. For the modest ascent search weighted function values can be used. In between the solution vector and the Tabu List are updated. If the solution does not improve after a given number of iterations or if all neighbourhood solutions are already set *tabu* the search for improved minima is aborted and the diversification search is performed to obtain new starting points. A pseudo-code of the GOTS and the TSPA algorithms is presented in Figure 2-14.

2.2.2 Investigation of the efficiency of the new approaches

Various well known multi-minima functions were used^{285,291,292,311,312,313,314,315,316,317,318,340} to prove the efficiency of the GOTS and the TSPA approaches. It also allows comparing their abilities with already known methods. In addition to test functions which were used for the investigation of the GTS method the Rosenbrock

(Rb_n), the DeJoung (DJ), the Zakharov (Z_n), and the Trid (Tr_n) functions were tested. Additionally, a functional form which was introduced to mimic conformational searches³⁴¹ was used to test the efficiency of the GOTS method for such problems. The corresponding expressions are given in Appendix A where the global minima of the functions are also given. One measure for the efficiency of a global optimization routine is the number of steps (operations) being necessary to reach the global minimum of the above mentioned functions. For the present study such numbers were obtained by running the simulations over a specific period and determining afterwards how many steps were made until the global minimum was passed for the first time. For the computation double precision variables were used and the global minima were always reached within numerical accuracy. The convergence of the optimization routines which is another measure for the efficiency of the algorithm was also tested.

2.2.2.1 Parameters

The proposed algorithms have several parameters and for a solid assessment of the success of the new algorithms their influence on the convergence of the optimization routine has to be tested. The parameters are reported in Table 3. The description of parameters and their influence on the problem are already given in the preceding chapter on the GTS approach. The test results for the GOTS and the TSPA are given below.

The parameters $rank_{max}$ and $rank_{min}$ are ranking parameters from linear ranking procedure (Eqs. 14, 15). They were always set to 0.1 and 1.0. A noticeable influence on the convergence was only found for parameters R_{TR} , $alam$, Δx_i , and α . Their effects are studied in Table 8 and illustrated in Figure 2-15 - Figure 2-18 using the Rastrigin and the Ackley function as examples. The values for the GTS which were taken from the previous chapter³³³ are given for an easier comparison. To estimate the influence on the convergence the tables show how often the function, its gradient, and the diagonal elements of its Hessian had to be computed until the global minimum was reached. The numbers are averaged over 10 test runs. The way of the calculation or choice of the parameter R_{TR} is identical to the manner in the GTS approach. Due to these reasons a value of $R_{TR} = 0.1$ seems again to be a good choice for a large variety of different problems in the cases of the GOTS and TSPA methods.

A similar behaviour is found for the parameter α ($0.0 < \alpha \leq 1.0$) which determines the size of the cone of the tabu directions³³³. Similar to large tabu regions too large cones restrict the flexibility of the search. The search must take detours to the global minimum since more direct ways are set tabu by accident. This is in line with the finding that for quite large cones the minimum is not found anymore (not shown in Table 3). Within our tests $\alpha = 0.4$ was found

to be a good choice. This value was used for all test cases for GOTS and TSPA discussed below.

The value of the first step size in the line search performed in the Steepest Descent and the Quasi-Newton part is defined by the parameter *alam*. It therefore influences only the GTS and the GOTS. For small values the local minimum search becomes more accurate, but more steps are necessary. Increasing *alam* from 0.2 to 2.0 the necessary number of function and gradient evaluations decreases considerably while the absolute minimum of both functions are still always found. The speed up may result since for a larger step size narrower and shallower minima are missed. Since deeper minima are of main interest this is no disadvantage. With a value of *alam* = 1.0 (full Newton step) one is on the safe side for all problems tested so far without a tremendous loss of efficiency.

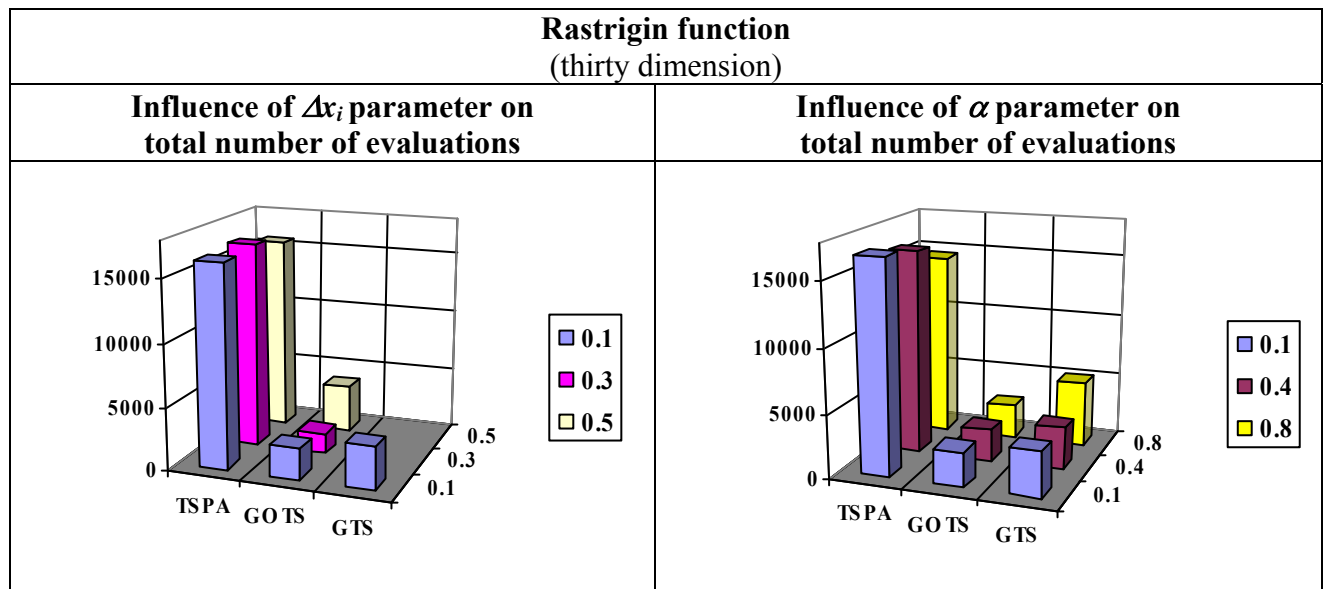


Figure 2-15. Influence of Δx_i and α parameters on the sum of evaluations for the thirty dimensional Rastrigin function.

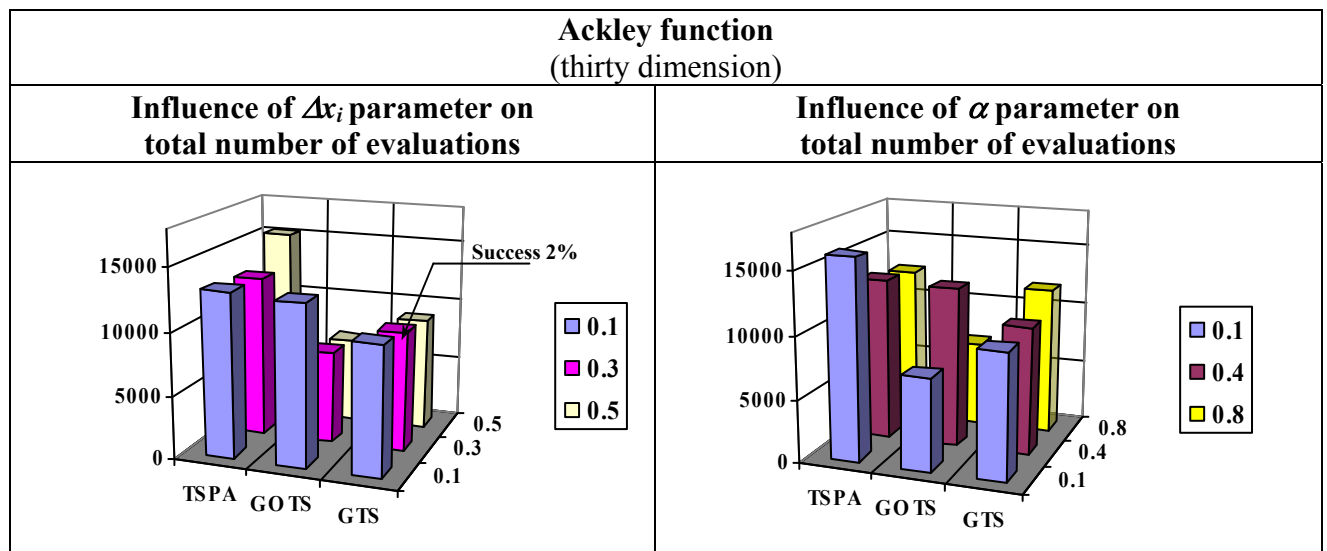


Figure 2-16. Influence of Δx_i and α parameters on the sum of evaluations for the thirty dimensional Ackley function.

Table 8. Influence of user-defined parameters for the thirty dimensional functions.

Function	Parameter	Value	TSPA	GOTS		GTS		
			Function	Function	Gradients	Function	Gradients	Hessian
Rastrigin	R_{TR}	1.0	20095	1398	39	863	73	49
		0.2	15977	1386	40	366	34	42
		0.1	16128	1295	40	293	34	69
	$alam$	2		1423	48	388	38	62
		1	-	1295	40	293	34	69
		0.2		1362	74	387	56	62
Ackley	R_{TR}	1.0	15974	1509	233	1596(60%)	277	60
		0.2	13769	1211	188	1173	219	108
		0.1	13033	3230	317	949	228	81
	$alam$	2		1273	206	1095	271	58
		1	-	1189	186	949	228	81
		0.2		1789	464	1177	406	114

The dependency of the total number of calculations (number of function evaluations + number of gradient evaluations) on Δx_i is shown in Figure 2-15 and Figure 2-16. The parameter Δx_i gives the step size during the modest ascent search. With a larger step size fewer steps are necessary to leave a local minimum; however, the number of steps needed to reach the global minimum could rise anyway since the search for the lowest transition state to the next local minimum becomes less accurate. This is seen if the GOTS is used to find the global minimum of the thirty-dimensional Rastrigin function. For $\Delta x_i = 0.1$ 1295 function evaluations and 40 gradient evaluations are necessary to reach the global minimum. For $\Delta x_i = 0.3$ only 647 function evaluations and 33 gradient evaluations are necessary but for $\Delta x_i = 0.5$ the effort again increases to 1396 and 79. The influence of Δx_i on the speed and the cost of the optimum reaching is large; the global minimum can even be missed as seen for the GTS method. The value $\Delta x_i = 0.1$ is recommended as a standard value.

For the Ackley function similar trends are found for all discussed parameters. In summary, Table 8, Figure 2-15 and Figure 2-16 indicate that the previous GTS and the new approaches GOTS and TSPA seem to be quite stable with respect to the actual values of the fine tuning parameters. Only the number of necessary steps varies, but the global minimum is always found.

2.2.2.2 Tests of the efficiency

Table 9 allows a comparison of the three different Tabu Search algorithms the TSPA, the GOTS, and the GTS. The values for the GTS³³³ are given for an easier comparison. As test cases different functions often applied for such purposes were used and the dimensionality of the functions was varied. As Table 8, Table 9 gives how often the function, its gradient, or the diagonal elements of its Hessian had to be computed until the global minimum was reached. The computation of the gradient or the diagonal elements of the Hessian represents a calculation of a vector of the dimension $NDIM$. To take this effort into account Table 9 also contains the columns “Sum” which give the sum of all necessary evaluations. For the GOTS “Sum” is given as the number of function evaluation plus the number of gradient evaluation times $NDIM$. For the GTS “Sum” additionally includes the number of Hessian evaluations times $NDIM$. To perform the tests on an equal footing the same parameter set were used for all approaches ($\alpha=0.4$, $\Delta x_i=0.1$, $R_{TR}=0.1$, $alam = 1.0$). For each case (function and dimension of the problem) 100 independent trials were carried out and averaged to obtain statistical results. Each trial corresponds to a new starting solution, chosen randomly in the search space.

Table 9. Comparison of the three different Tabu-Search algorithms. All values are averaged over 100 trial runs. Percentages in brackets indicate the probability to find the global minimum.

Test function	NDIM	TSPA	GOTS			GTS			
		Function	Function	Gradient	Sum	Function	Gradient	Hessian	Sum
Rastrigin (R_n)	2	230.4	46.18	10.95	68	42.07	6.88	2.76	61
	10	3765.24	304.07	28.21	586	143.95	29.98	61.09	245
	20	9084.35	820.17	33.92	1498	220.40	33.30	67.72	2241
	30	16128.21	1295.08	40.36	2506	293.36	34.23	69.15	3394
	50	29604.20	2412.14	50.08	4916	470.43	38.17	76.06	6182
Ackley (AK_n)	2	265.33	214.45	78.50	372	138.40	63.54	24.71	315
	10	3366.52	826.98	196.91	2796	499.05	185.12	38.58	2736
	20	9100.18	1662.23	253.16	6725	793.17	214.90	62.14	6333
	30	13033.77	3230.11	317.05	12742	949.12	227.52	80.74	10197
	50	20764.25	3808.75	225.42	15080	1335.84	228.74	114.19	18482
Griewangk (G_n)	2	2364.70	1310.44	165.67	1641	269.32	132.49	1236.36	3007
	10	48960.60	1831.97	200.35	3835	944.21	669.85	568.78	13330
	20	61902.11	540.60	13.60	812.6	1062.07	207.7	942.18	24060
Levy (L_n)	4	21994.00 (5%)	11611.08 (75%)	5817.01	34879.1	7166.64	4586.28	868.90	28987
	7	37778.22 (9%)	15000.81 (54%)	10064.83	85454.6	5716.27	4931.76	711.74	45221
Branin (BR)	2	294.45	284.03	23.65	331	26.83	11.67	9.40	69
Goldstein-Price (GP)	2	3833.19	79.87	20.15	120	49.27	16.13	10.08	102
Hansen (H)	2	2872.55	614.14	154.31	923	387.52	117.46	211.17	1046

Overall Table 9 shows that the GTS and the GOTS are considerably more efficient than the TSPA. This could be expected since gradient-based approaches are known to be much more efficient for minimization problems. The difference between both the GOTS and the GTS depends on the function and its dimensionality ($NDIM$). Generally, the GTS seems to be more efficient for small $NDIM$ but for larger $NDIM$ the GOTS becomes the most efficient approach. The differences between the numbers of gradient evaluations indicate that both approaches take different ways to the global minimum (see below).

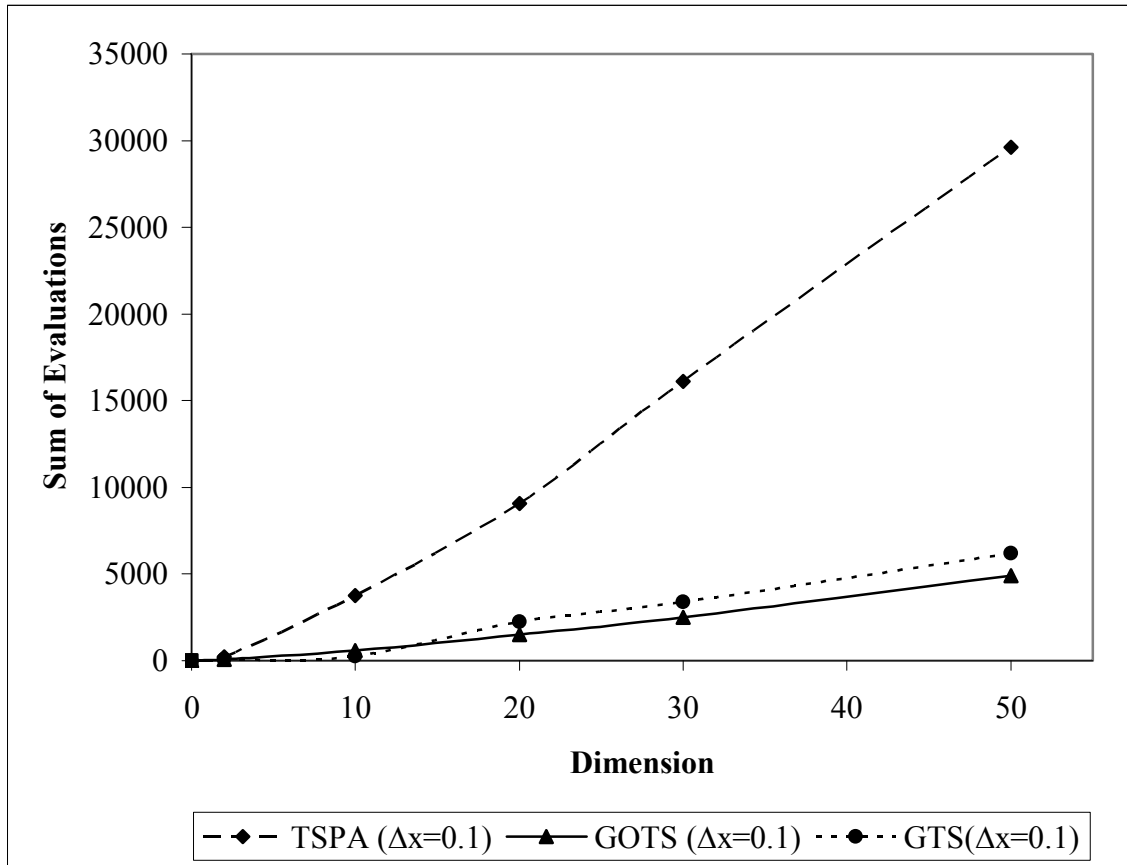


Figure 2-17. Dependency of sum of evaluations on the dimension of the Rastrigin function.

For the Rastrigin function the GTS requires a lower number of evaluations than the GOTS for $NDIM=2$ and 10 but for the higher dimensionalities the GOTS becomes more effective. In all cases the dependency of the sum over all evaluations (column “Sum”) on $NDIM$ is about linear but with varying prefactors. For the Rastrigin function (Figure 2-17) the sum of all evaluations needed by the GOTS increases by a factor of about 8 if the dimension increases from 2 to 10. Going from $NDIM=10$ to 50 the sum of all evaluations increases from 586 to 4916, i.e. again by a factor of about 8. The GTS approach behaves very similar. For the TSPA algorithm the dependency is much stronger. Going from 2 to 10 dimensions the sum of evaluations increases by a factor of 16, i.e. the increase is nearly quadratic. However, going from 10 to 50 dimensions the sum of evaluations increases only by a factor of 8.

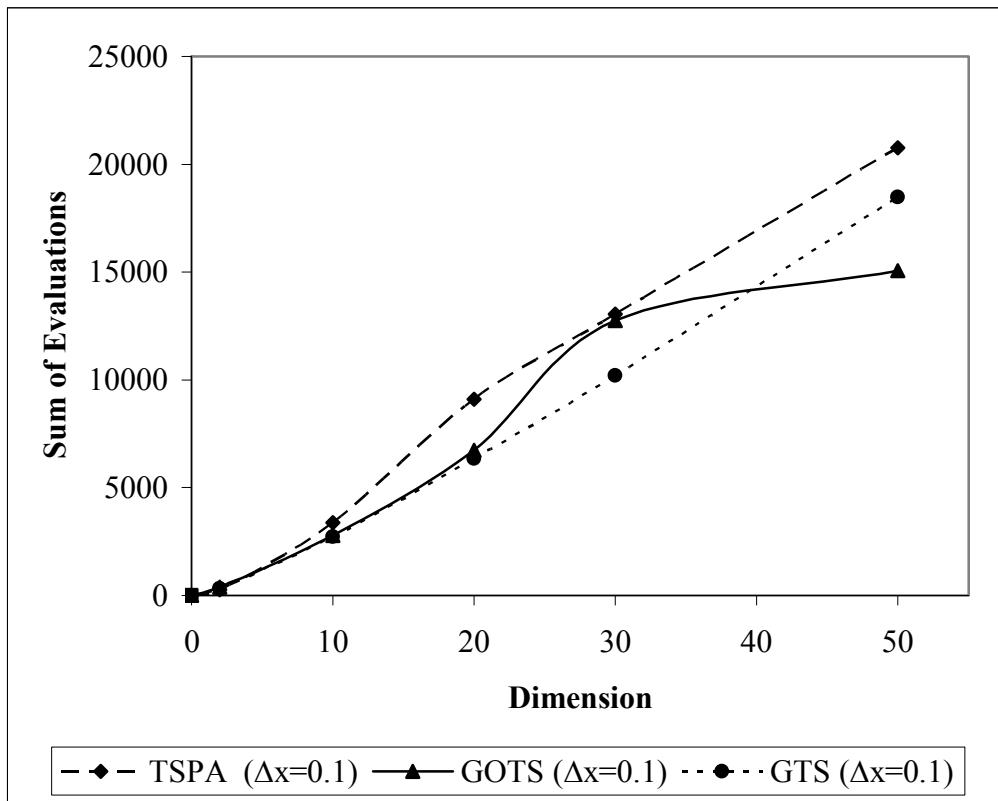


Figure 2-18. Dependency of sum of evaluations on the dimension of the Ackley function.

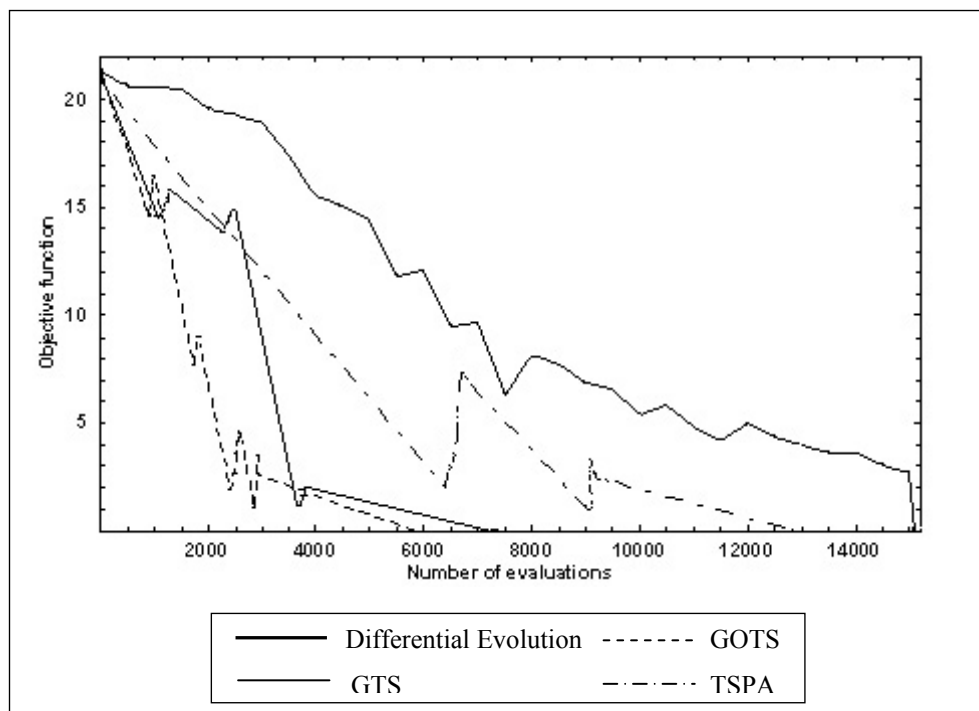


Figure 2-19. Comparison of the convergence of the algorithms for thirty dimensional Ackley function. The symbols indicate the values of the objective function at the local minima which were found during the GOTS.

For the Ackley function similar trends are found but the slope of the dependency on the dimensionality (Figure 2-18) is considerably steeper indicating that this function

represents a more complicated problem. This is also seen from the absolute numbers given in Table 9. An insight into the convergence of the various approaches is also given in Figure 2-19. It sketches the computed function values of the Ackley function along a given search. All approaches start from the same point. Of course the taken routes to the global minimum depend on the starting points but Figure 2-19 gives a representative example. It is directly seen that although starting from the same starting point GTS, GOTS, and TSPA take quite different routes. The route taken from the GOTS converges most rapidly and already after about 3000 evaluations the functional values decreases to about 1. For the GTS about 3500 evaluations are needed and for the TSPA more than 6000 evaluations are necessary. Figure 2-19 also proves that all approaches are capable to escape low lying regions quite efficiently and shows that they converge more rapidly than the differential evolution algorithm^{328,329,330,331,332} implemented in the Mathematica³²⁴ (see below).

Table 9 shows that the Griewangk function seems to be ideally suited for the GOTS. It finds the global minimum very efficiently. Going from 10 to 20 dimensions the number of necessary evaluations even decreases. This happens since the search jumps over various local minima which are very narrow. This does not represent a disadvantage since only the global minimum, which was found in each run, is of interest. For the Griewangk function the GTS and the TSPA need considerably more evaluations than for the Rastrigin or the Ackley functions. This again shows that the three approaches take completely different routes although similar strategies are used.

The Levy function seems to be a difficult case for all approaches. However, while the GTS still finds the global minimum in all cases for $NDIM=4$ the TSPA finds the minimum only in 5 % of all test runs. For $NDIM=7$ it finds the global minimum only in 9 % of the cases. The GOTS finds the minimum in only 75 % ($NDIM=4$) and 54 % ($NDIM=7$). If $\Delta x_i=1.0$ instead of 0.1 is used the GOTS and the TSPA also always find the global minimum. In this case for $NDIM=4$ the GOTS needs 3403 evaluations to reach the global minimum while 10584 are needed for $NDIM=7$. The TSPA needs 18507 and 38930 evaluations, respectively.

For the Branin function the outcomes are very similar to those for the Rastrigin function. The Goldstein-Price function seems again to be more difficult for the TSPA. The numbers indicate that the Hansen function is a difficult test function for all approaches. For the two latter functions the influence of the step size is quite strong for the TSPA approach. Increasing Δx_i to about 1.0 the number of function evaluations drops to about 335 and 533, respectively. GOTS tests for the DeJoung, the Rosenbrock, the Zakharov, and the Trid functions also underline the efficiency of this approach (Table 13).

Table 10 - Table 13 summarize comparisons with optimization routines taken from the literature. An objective comparison of the computational effort needed to find the global minima is only given by wall clock times taken on the same machine since the computational overhead for the steps varies from method to method. Since this is nearly impracticable the number of function evaluations is normally taken for comparison. They are summarized in the column *Evaluations* of the tables. For GOTS and GTS the tables also give the sum over all necessary evaluations (see Table 9). The data obtained within the present study were averaged over 100 trial runs and the numbers are rounded to integer. A comparison of the convergences of the GTS, the GOTS, the TSPA, and the differential evolution algorithm as implemented in the Mathematica program³²⁴ were already given in Figure 2-19.

Table 10. Comparisons with the literature for the Rastrigin function. All values are averaged over 100 trial runs. The column “Evaluations” gives the function evaluations. For the GTS it comprises the sum of function, gradient and Hessian computations as second number. For the GOTS it also comprises the sum of function and gradient computations as second number.

NDIM	Algorithm	Evaluations	Source
2	Pure random search	5964	315
	Fast annealing evolutionary algorithm	544	314
	Random tunneling algorithm	383	318
	TRUST	59	
	Tabu Search	540	291
	Differential evolution	489	
	Random search	610	324
	Simulated Annealing	260	
	GTS	42/61	333
	GOTS	45/66	338
	TSPA	230	
10	Tabu Search-MS 400	11624	
	Tabu Search-REM	1981	316
	Genetic Algorithm	4488	
	Differential Evolution	14017	331
	Differential Evolution – Quasi Newton	13955	344
	GTS	143/1053	333
	GOTS	276/656	338
	TSPA	3765	
20	Genetic Algorithm	>100000	321
	Cooperative Coevolutionary Genetic Algorithm	20000	322
	Tabu Search-MS 200	60446	316
	Tabu Search-REM	7961	
	Parallel Genetic Algorithm	9900	322
	EASY Genetic Algorithm	6098	323
	Breeder Genetic Algorithm	3608	
	Differential Evolution	23870	331
	Differential Evolution – Quasi Newton	15200	344
	GTS	220/2240	333
	GOTS	553/1387	338
	TSPA	9084	
30	GTS	293/3394	333
	GOTS	1397/3776	338
	TSPA	16128	
50	GTS	470/6181	333
	GOTS	1876/5079	338
	TSPA	29604	

Table 11. Comparisons with the literature for the Ackley function. All values are averaged over 100 trial runs. All tests were carried out on the same computer. The column “Evaluations” gives the function evaluations. For the GTS it comprises the sum of function, gradient and Hessian computations as second number. For the GOTS it also comprises the sum of function and gradient computations as second number.

NDIM	Algorithm	Evaluations	Source	Wall clock time [sec]
2	Random search	1257	324	3243.3
	Differential evolution	311		0.4
	GTS	138/315	333	<<0.1
	GOTS	164/300	338	<<0.1
	TSPA	265		<<0.1
10	Differential evolution	4214	324	14.8
	GTS	499/2736	333	<<0.1
	GOTS	649/2536	338	<<0.1
	TSPA	3367		<<0.1
20	Differential evolution	9957	324	80
	GTS	793/6333	333	<<0.1
	GOTS	968/5187	338	<1
	TSPA	9100		<1
30	Genetic Algorithm	>100000	321	
	Cooperative Coevolutionary GA	50000	322	
	Breeder Genetic Algorithm	19420	323	
	Easy Genetic Algorithm	13997		
	Tabu Search-MS 100	22842	316	
	Tabu Search-REM	17941		
	Differential evolution	14957	324	222.7
	GTS	949/10197	333	≈ 1
	GOTS	1189/6768	338	≈ 1
	TSPA	13034		≈ 1
50	Differential evolution	22557	324	888.9
	GTS	1336/18482	333	≈ 2
	GOTS	1792/11785	338	≈ 1
	TSPA	20764		≈ 3

Table 12. Comparisons with the literature for the Branin and the Goldstein-Price functions. All values are averaged over 100 trial runs. The column “Evaluations” gives the function evaluations. For the GTS it comprises the sum of function, gradient and Hessian computations as second number. For the GOTS it also comprises the sum of function and gradient computations as second number.

Function	Algorithm	Evaluations	Source
BR	Tabu Search-CJ	492	291
	Pure random search	4850	
	Simulated Annealing 1	2700	
	Multi-start	1600	317
	Simulated Annealing 2	505	
	Tabu Search	492	
	Paper work	144	
	TRUST	55	318
	Tabu Search-REM	192	316
	Tabu Search-MS20	333	
	Mod. Annealing	144	317
	Simulated Annealing	324	
	Random search	15	324
	Differential evolution	10	
	GTS	32/69	333
	GOTS	44/65	338
	TSPA	203	
GP	Simulated Annealing 1	5439	
	Pure random search	5125	317
	Multi-start	4400	
	Tabu Search-REM	894	316
	Simulated Annealing 2	563	
	Tabu Search	486	317
	Mod. Annealing	486	
	Enhanced Continuous Tabu Search	231	290
	Differential Evolution	18123	331
	Differential Evolution-Quasi Newton	14781	344
	Simulated Annealing	563	320
	Directed Tabu Search	230	286
	Tabu Search-MS 500	195	316
	Paper work	112	317
	TRUST	103	318
	GTS	49/102	333
	GOTS	52/73	338
	TSPA	336	
	C-RTS-min	171	283
	ECTS	231	
	DOPE	248	290
	ESA	783	
	CTSS _{multiple}	119	312
	CHA	259	342

The Rastrigin function was often employed to test the efficiency of optimization routines^{291,292,314,315,316,319,321,322,323}. According to the number of function evaluations or also to the number of total evaluations GTS and GOTS performs much better than the approaches taken from the literature (Table 10). The Terminal Repeller Unconstrained Subenergy Transformation algorithm³¹⁸ includes gradient computations as GTS or GOTS but still needs 20 % more function evaluations. The number of total evaluations was not given in the literature. The TSPA needs more evaluations but is still considerably faster than e.g. the Fast annealing evolutionary algorithm or the original Tabu Search implementation²⁹¹. The test runs made for the 10- and 20-dimensional Rastrigin function allow a comparison with several previous modifications of the Tabu Search^{291,316} and various genetic algorithms^{319,321,322,323}. The so called Tabu Search-REM method³¹⁶ is much better than all previous Tabu Search approaches³¹⁶. Nevertheless, the GTS and the GOTS are still considerably faster. The TSPA approach needs more function evaluations than the Tabu Search-REM. GOTS and GTS also outperform the genetic algorithms. For even higher dimensional Rastrigin functions no other tests could be found.

The thirty dimensional Ackley function (Table 11) was also used as test case for genetic algorithms^{321,323} and improved Tabu search approaches³¹⁶. Please note, that the GOTS results were obtained with $x_i=0.5$ while $x_i=0.1$ was used for GTS and TSPA. All other settings are equal to those employed in Table 9. To get more information the *Random search*^{7,325,326,327} and the *Differential evolution*^{328,329,330,331,332} approach implemented in the Mathematica 5.1 program package were also used. While the former were only used for the two dimensional case the latter were employed for all dimensionalities. The differences between the various approaches are similar to those found for the 20-dimensional Rastrigin function. For all approaches the Ackley function seems to be a more difficult problem than the Rastrigin function.

Table 13. Comparisons with the literature for the Hansen, the DeJoung, the Rosenbrock and the Zakharov functions. All values are averaged over 100 trial runs. The column “Evaluations” gives the function evaluations. For the GTS it comprises the sum of function, gradient and Hessian computations as second number. For the GOTS it also comprises the sum of function and gradient computations as second number.

Function	NDIM	Algorithm	Evaluations	Source
H	2	Random search	420	
		Simulated Annealing	412	324
		Differential evolution	291	
		GTS	217/425	333
		GOTS	321/496	338
		TSPA	534	
DJ	3	ECTS	338	311
		DOPE	131	294
		CTSS _{multiple}	155	312
		CHA	371	342
		GOTS	23/62	338
Rb _n	10	ECTS	15720 (85%)	311
		DOPE	8695	294
		ESA	12403	290
		DTS _{APS}	9037 (85%)	286
		Differential Evolution	67310	331
		Differential Evolution-QN	16349	344
		Simulated Annealing	12403	320
		CGA	21563 (80%)	343
		GOTS	379/1629	338
Z _n	10	ECTS	4630	311
		DOPE	5133	294
		Differential Evolution	10088	331
		Differential Evolution-QN	10013	344
		Simulated Annealing	15820	320
		CGA	6991	343
		GOTS	61/161	338
Tr _n	5	GOTS	38/93	338
		EM without LOCAL procedure	968	345
	20	GOTS	125/745	338
		EM without LOCAL procedure	43354	345
Test function for conformational search	7	Deterministic algorithm	168	341
		GOTS	131/271	338
	10	Deterministic algorithm	497	341
		GOTS	192/412	338
	20	Deterministic algorithm	174857	341
		GOTS	2038/5378	338

On the basis of the number of evaluations GTS and TSPA seem to be similar in efficiency to the *differential evolution ansatz* taken from the Mathematica 5.1 program package. However, if real wall clock times are compared our approaches needs considerably less time (1 second in comparison to 222 seconds). This difference may arise since for the present approaches the number of additional operations (beside the evaluations themselves) is quit small, but it may also result from the implementation and the employed computer. Another advantage of our approaches is their stability with respect to the fine tuning parameters. They find the global minimum also for nonoptimal fine tuning parameters. In contrast the *differential evolution ansatz* sometimes misses the global minimum if the default parameters implemented in the Mathematica 5.1 are used. The differences in the convergence were already discussed in the context of Figure 2-19.

The investigations performed with the Branin, the Hansen, the Goldstein-Price, the Trid, the Rosenbrock, the DeJoung, and the Zakharov functions lead to results similar to those found for Rastrigin and Ackley function (Table 12-Table 13). For the Branin, the Hansen and the Goldstein-Price functions was employed $\Delta x_i = 1.0$ as more efficient for GOTS and TSPA.

The GOTS method was additionally applied to a functional form which was introduced to mimic conformational searches³⁴¹. The results are convincing (Table 13). The global minimum is always found quite effectively. The larger numbers for $NDIM=20$ result since the number of computations vary considerably. They range from about 900 to about 8000 within the 100 trial runs.

2.2.3 Conclusions

Two new global optimization routines; the Gradient only Tabu Search (GOTS) and the Tabu Search with Powell's Algorithm (TSPA) are described in this chapter. They are based on the Tabu Search strategy which tries to determine a global minimum of a function by the *steepest descent - mildest ascent* strategy. In this strategy the steepest descent is followed to find the next local minimum. To escape the local minimum the mildest ascent is used. Already visited regions are set tabu using a Tabu List which memorizes the already visited regions. The GOTS uses a combination of the Steepest Descent and the Quasi-Newton method to find the next local minimum. A step-wise approach is employed to realize the modest ascent strategy. The TSPA uses the same implementation for the modest ascent strategy but Powell's algorithm for the steepest descent part, *i.e.* the TSPA only needs function evaluations but no gradients. The algorithms are explained and their efficiency is compared to other approaches by determining the global minima of various well-known test functions. These tests show that the GOTS is more efficient than previous global optimizers.

For functions with higher dimensionality it is even more efficient than the previously developed GTS which uses the diagonal elements of the Hessian to escape from a local minimum via the modest ascent. The efficiency of the TSPA is comparable to Genetic Algorithms. Test computations also indicate that the GOTS is an efficient approach for conformational searches.

2.3 Conclusions

This chapter shows that the new approaches: GTS, GOTS, and TSPA can be efficiently applied to the optimization of continuous multim minima functions. These techniques can be useful for engineers or scientists, who are interested in the global solution but also nearby local optima. Efficient results were obtained within a reasonable amount of computational time. The proposed approaches are flexible and easy to use. Moreover, the proposed approaches could be enhanced even further if more sophisticated descent method and diversification strategies are applied instead.

The Figure 2-20 gives the graphical illustration of convergence of the all new approaches described above in comparison with Differential Evolution, Linear and Quadratic convergences. The dependency of sum of evaluations on the dimension of the Ackley function for GOTS it is close to linear. It is obvious that the GOTS method achieves better solutions than already known pure and hybrid global methods, as well as GTS and TSPA methods.

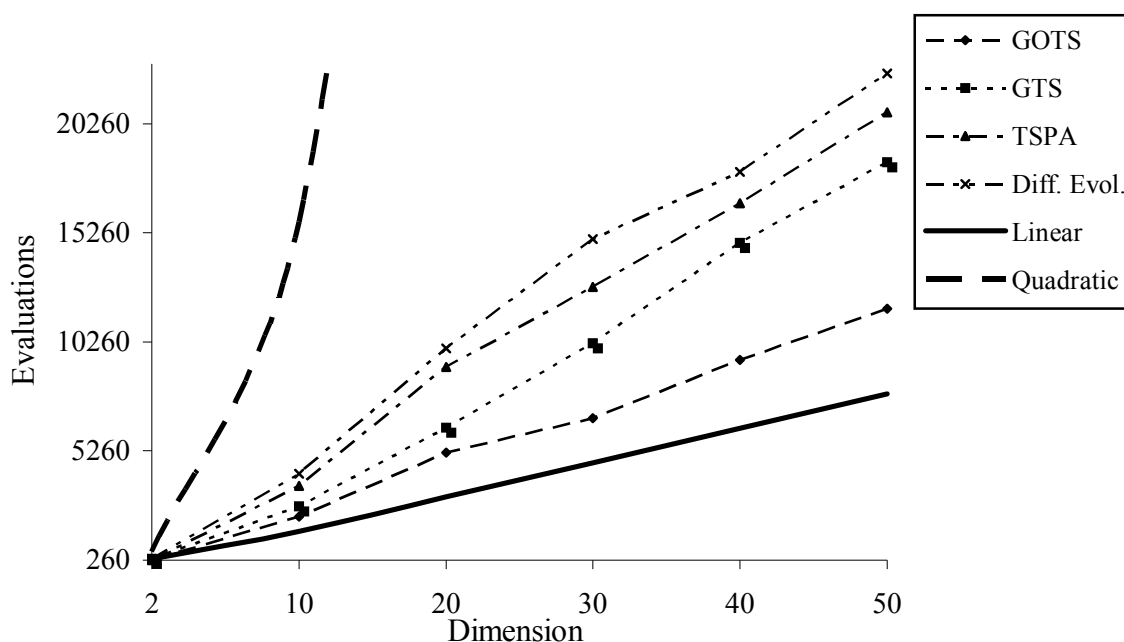


Figure 2-20. Graphical illustration of convergence of the algorithms.

In order to test the appropriateness of the GOTS method for molecular geometry problems, the optimization of a function with a functional form similar to general potential energy functions and whose global minimum is known was carried out. The number of its local minima increases exponentially with the size of the problem, which characterizes the principal complexity of the minimization of the molecular potential energy functions. The utilization of a GOTS guarantees the global optimality and shows the difficulty in obtaining the global minimum of such function. It was successfully applied to the n -chain problem ($n = 7, \dots, 20$) with decided superiority over Deterministic algorithm that uses the Branch and Bound scheme with techniques of interval analysis to provide the lower bounds. In all the cases the global minimum was obtained and the amount of function evaluations required determining the global minima increases by a factor of about 11 if the dimension increases from 10 to 20 whereas this factor for Deterministic algorithm is 352.

The examples show that the GOTS is able to provide solutions to difficult global optimization problems with minimal computational effort. The algorithm can be suitably modified to incorporate more rigorous mechanisms to handle very large-scale problems. Because of problem-specific knowledge GOTS deals with variables and constraints effectively. However, the trade-off between effectiveness and generality requires careful consideration, and the junction with other techniques is necessary in order to build general methodology. The recommended default parameters were given as appropriate for several test cases. Large-scale test cases show that the initial parameter settings for GOTS may be overcautious; however, in any case, they locate the global optima quite rapidly and with high probability. The variable choice and parameter settings of GOTS is always problem dependent. The optimal parameter values may require some adjustment to starting values indicated in this chapter. As is obvious from the preceding discussion, GOTS determines the best energy minimum of the molecular geometry problems efficiently. The next chapter presents details for applying GOTS to chemical optimization problems on potential hypersurfaces and provides a systematic approach to variables choice and setting the adjustable parameters.

Chapter 3 Application and Discussion

3.1 Introduction

An efficient search for the global minimum of a highly dimensional function with many local minima is central for the solution of many problems in computational chemistry. Well known examples for such global optimization problems¹⁰ are the conformational search for molecules with a high number of freely rotatable bonds^{346,347}, the optimization of the parameters of a force field or the determination of all possible reaction paths between reactants and products^{348,349,350}. Conformational analysis is the study of molecule conformations and their influence on molecular properties. Fundamental for all conformational analysis is the search for the most important conformers of the molecule. This requires the location of the corresponding energy minima^{309,351} on the potential energy surface (PES)³⁵². In most cases the energetically lowest lying conformers are the most important ones, since they are populated at e.g. room temperature and, hence determine most of the properties of the molecule^{353,354}.

Search for the energetically lowest lying conformer means to find the global energy minimum of the energy hypersurface of the molecule. Hence it is a global optimization problem in which the potential energy function is the objective function and the coordinates, that are used to represent the conformation of the molecule, are the variables. For larger molecules with many freely rotatable single bonds the number of minima may be so large that it is nearly impossible to find all of them. In order to locate the global energy minimum or to locate the lowest minima, a large number of starting conformations that are equally distributed on the energy surface is generated. Each of them is minimized using local optimization techniques³⁵⁵ to the nearest local minima, and then all duplicated structures are rejected. Even for small molecules the disclosure of the global energy minimum or all low energy conformations in a multi-dimensional PES is a problem that requires considerable computational effort. A complete search is unrealizable since the search space increases exponentially with the number of degrees of freedom (e.g. torsion angle), which is typically proportional to the size of the molecule. It can be referred to nondeterministic polynomial-time hard problems³⁵⁶, where the total number of possible conformations increases exponential with the total number degrees of freedom. The time required to solve it increases

exponentially with the size of the molecule. This is known as *combinatorial explosion*³⁵⁷. Therefore, to explore the complete conformational space and to obtain all the low energy conformations at tractable computational cost some specialized conformational search algorithms are needed. Over the past several years, a multitude of conformational search techniques have been developed for this purpose^{358,359,360,361,362,363,364,365}. A brief review of some of the methods is given below.

3.1.1 Conformational search techniques

Conformational search methods can be roughly divided into stochastic and deterministic techniques. Stochastic methods are optimization algorithms which incorporate probabilistic (random) elements to locate the global minimum either in the problem data or in the algorithm itself, or in both. Hence there is no natural endpoint of the procedure. In the deterministic approach the values of the objective function are assumed to be exact, and the computation is completely determined by the values sampled so far.

- *Systematic (or grid or exhaustive) search methods*^{366,367} possess as first step the generation of a large number of starting geometries using combinations of selected values of variable internal coordinates. In most cases only the torsion angles, which describe the rotation around the single bonds are used. Each torsion angle is systematically incremented by a fixed amount until all possible combinations of torsion angles for the chosen increment have been generated. Then each structure is subjected to an energy minimization to yield the corresponding low energy final conformations. The total number of search steps of such searches must be predefined at the beginning of the search when the starting geometries are generated. Except for simple molecules, the number of steps that are necessary to find all conformations is unknown at the outset and may be so large that the energy minimization stage of the search becomes prohibitive. The necessity of using many values for each torsion angle reinforces the problem since low energy conformations of flexible molecules commonly incorporate relatively high-energy torsional arrays. Thus, the method is limited to very small molecules or molecule segments.
- *Build-up procedures*³⁶⁸ are based on the assumption that the conformations of different fragments of the molecule are independent of the other ones. Comprehensive energy minimizations are carried out for all fragments, and only those local minima whose energies lie within an appropriately chosen upper bound of the lowest minimum are retained. To obtain starting points for the energy minimization of larger fragments the restricted set of local minima of one fragment is combined with the set from another

fragment. If the assumption holds the global minimum of the whole chain can be built up from its components by subsequent processes without the need to explore more than a small part of the conformational space of the whole molecule. However, in the realization, the build-up procedure has some difficulties. The main one is the exponential increase of the number of minima which have to be retained at each step to obtain an acceptable value for the cut-off energy. Additionally, two atoms can come too close if the fragments are combined in an arbitrary manner. A final difficulty concerns the correct value for the cut-off energy.

- *Monte Carlo methods (the Metropolis algorithm)*^{369,370,371,372,373} are stochastic techniques that are able to generate a random sample of low-energy conformations for molecules that are too large and flexible to be explored systematically. In these methods the energy E_0 is calculated first for an arbitrary conformation. In each step of the algorithm, several torsion angles (coordinates in Cartesian space) are randomly varied leading to a new conformation with energy E . This new conformation is accepted as new starting point or rejected depending on the rules described in the first chapter (Chapter 1 - Eq. 6). For macrocyclic molecules, random variations of torsional angles often result in conformations for which the ring-closure constraints are violated. Such structures cannot be used for minimization and are rejected. The Monte Carlo step has to be repeated until a set of low energy conformers has been generated. Several modified Monte Carlo methods have been applied for peptides and proteins³⁷⁴.
- *The Simulated annealing method*^{219,375,376} is based on a connection between statistical mechanics and the crystallization process. The main principles of this technique were described in the first chapter. By analogy with this physical process, in a combinatorial optimization context, a solution corresponds to a state of the physical system and the solution cost to the energy of the system. At each iteration the current conformation is replaced by a randomly selected trial conformation accepted according to the so-called Metropolis criterion. The series of accepted steps is an “energy directed random walk” which explores the conformational space. The simulated annealing technique exploits both the energy and temperature dependency of the Boltzmann distribution. At a given temperature, the Metropolis algorithm is used to imitate the conformational equilibrium using the energy dependency of the Boltzmann distribution. Then at progressively lowered temperature during the process according to some set cooling schedule is used to profit by the temperature dependency of the Boltzmann distribution. The method is widely used for the conformational search problems of molecules^{219,376}.

- *Internal coordinate tree search*³⁷⁷ is the more systematic version of an internal coordinate Monte Carlo search and a rapid method for generating molecular geometries which approximate the low-lying conformers of small to medium-sized organic molecules. It generates starting conformations for subsequent minimizations by varying the torsion angles of the rotatable bonds in series of possible values (e.g., 0°, 60°, 120°, 180°, . . .). Bond lengths and angles are kept fixed. In the case of N rotatable bonds and d possible values of each torsion angle up to d^N conformations may be generated and minimized. In worst case, every combination of all torsion angle values must be investigated. A torsion angle ordering allows the complete conformer generation process to be graphed as a tree structure. At the lowest level of the tree, the leaves stand for the resulting conformations of the generation process. Each edge stands for the rotation of one torsional angle. Intermediate nodes in the tree represent partially completed structures. For cyclic molecules the conformations which violate the ring-closure constraint have to be excluded. This method can be quite expensive for macrocyclic molecules but can lead to the disclosure of previously non-described low-energy conformers. The internal coordinate conformer generator has a number of advantages:

1. It is quite fast because it demands no energy gradient evaluations and no solution of matrix of distance equations that are required e.g. for the distance geometry methods (see below). That increases the speed of conformer generation which ranges from 10- to 100-fold over molecular dynamics or distance geometry methods
2. Internal coordinates handling assures the sampling over all accessible regions of conformational space
3. It is easily applied as a tree search, which enable analysis of acyclic and cyclic molecules.

However, since this method is a probabilistic one it gives no guarantee that all low-energy minima have been found.

- *Genetic algorithms*^{378,379,380} follow standard procedures that are derived from the principles of natural evolution. The main principles of the algorithms were already described in the first part of the work. Here the explanation of the implementation specificity of such methods for the conformation search is given. An initial (parent) population is built by assigning each main-chain torsion angle to a value randomly chosen from allowed ones. The fitness of each conformation is then calculated using the energy of the minimized structure, i.e. each of the initial conformations is minimized using the local minimization procedure. The optimized conformations form the first parent generation as well as a pool of random conformations. New child conformations are

produced from the parent population using standard genetic algorithm operations. The probability that a parent conformation is selected for the next generation is proportional to the Boltzmann factor of its energy so that low-energy conformations are more likely to be selected than high-energy conformations. Each parent conformation is selected only once. To keep up the diversity, before each crossover operation, one of the two selected parent conformations can be replaced by a conformation, which has a small probability, randomly chosen from the pool of random conformations. The parents selection and crossover process is repeated until all conformations of the parent generation are selected. The reason combining the parents and newly generated child conformations, which were already sampled in the previous populations, with each other is the attempt to obtain promising structures. This shall assure that the fitness of the new generation is not worse than that of the parents. Then a “cleaning” process is performed that consist in rejecting conformations with a higher energy from any pair of structures which is similar to each other. The next process iteration uses the new generation as parents. The cycle is repeated until convergence. The best fit population is finally obtained or a maximum number of iterations has been reached. The application of the genetic algorithm in peptides and protein structure prediction was reviewed by Le Grand and Merz³⁷⁹, and Schulze-Kremer³⁸⁰.

- *Distance geometry methods*^{381,382,383,384,385} represent molecules as distance matrixes containing lower and upper bounds of the distances between every pair of atoms. These distance bounds consist of bonded constraints, such as bond lengths and angles, as well as nonbonded constraints, such as van der Waals radii. It is significant that the distance information alone is unable to differentiate between a structure and its mirror image. Hence, chirality constraints are included to eliminate this disadvantage. The constraints are given beforehand on the basis of the known stereochemistry of the chiral centres. The distance geometry representation can be applied to cyclic molecules as well as to acyclic molecules. At the first stage a distance matrix, which describes the structure of the molecule, is created whose elements d_{ij} are the distances between atoms i and j . Then a matrix with upper and lower bounds for each interatomic distance is determined. New conformations are typically generated by assigning new interatomic distances which lie between the upper and lower bounds. Then those conformations are generated that best approximate these distances. The resulting conformations are finally energy-minimized. Taylor and Aszodi³⁸¹ applied this method to polypeptides.
- *Smoothing/deformation methods* rely on the assumption that the position of the global minimum of a deformed energy hyper surface can be related to the position of the global

minimum of the original function. The algorithm is based on the deformation of the original potential energy hypersurface in such a way that the number of the local minima is reduced. Only the deepest one is retained, which, in most cases, is related to the global one. A local minimization procedure applied to the modified energy function leads from any starting point to this single minimum instead of ending up in some higher-energy local minimum. The hypersurface of a molecule is deformed (or smoothed) by applying certain mathematical operators (e.g. diffusion equation) with the original form of the hypersurface having the meaning of the initial concentration (or temperature) distribution. However, the position of the minimum in this deformed function may have changed during deformation and is usually different from that of the global minimum in the original function, therefore a reversing deformation procedure is used to trace back from the found minimum in the final deformed function to the related minimum in the original function which is in most cases the true global minimum. An extremely important characteristic of this method is the abilities of the deformation of the hypersurface without investigation of large quantity of local minima or any information about their positions beforehand. A number of smoothing procedures have been developed and applied to peptides and proteins³⁸⁶.

- *Molecular dynamics*^{387,388,389} is a stochastic method that simulates the physical behaviour of molecules in a thermal bath. Molecular dynamic procedures search the conformational superlattice of energy minima using an MD-like sampling strategy. It exploits the additional information about the conformational space using the MD velocities. They reflect, to varying degrees, all components of the system together that increases the efficiency of conformational sampling in more complicated systems. Molecular dynamics investigate the conformational surface incrementally in three-dimensional space. Multidimensional conformational space is described in terms of a reduced representation of conformational states. The algorithm initiates its move from a conformation state that is a local minimum-energy configuration by stepping along the MD velocities. The atoms in the molecule are usually constrained using a force field defined as a function of atom type, bond type, torsion type and interatomic distance. Typically for this technique is the application of Newton's second law of movement to all atomic degrees of freedom under which the new positions and velocities of the atoms are calculated. The atoms are moved to these new positions and the cycle is repeated. Thus, the dynamic behaviour of the molecule at the desired temperature can be reproduced by performing this process for some time space. Conformations are selected from the trajectory and minimized at each time step. An elevated temperature is used in order to overcome potential energy barriers

and to reach new regions of the PES that may contain better lower-energy minima conformers than the current region. Since the Molecular dynamics require a time step of the order of 1 fs, it is computationally expensive and is often limited to peptides^{358,360,361,362,363,364,365,357,390}.

- *Corner flapping*³⁹¹ produces a new conformation that is then subjected to energy minimization. It is an intuitional flap movement of ring atoms of a cyclic molecule relative to the average ring plane, ring atoms occupying the "corners" of the bonds. It is preferable to define a local plane based on several ring atoms near the corner and to move the corner atom vertically to the mirrored position on the other side of the plane (Figure 3-1). The new conformation is optimized and compared with the saved conformers relying on usual criteria such as symmetry considerations, dihedral angle pattern, and steric energy. Only new conformers are saved while repeated conformers are rejected. When all corners have been flapped, this molecular structure is marked as "investigated". After that, the most stable of the "uninvestigated" saved conformers take part in the subsequent flapping/minimization process. Since only local deformation of the cyclic molecule takes place the subsequent energy minimization process is rapid. It is significant that in contrast to other systematic algorithms this approach does not produce a tree structure, since every branch is instantly pruned to retain the new conformer only. The search terminates when all the saved conformers are "investigated". Corner flapping sometimes leads to excessively strained structure when the flap movement shifts the corner atom into the ring. These weaknesses result since the environment of the corners being flipped are not into account.

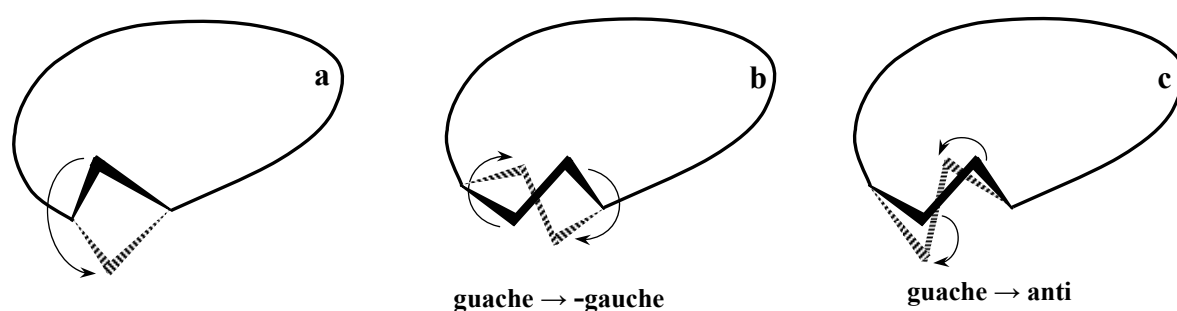
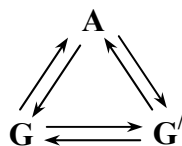


Figure 3-1. Illustration of the a) Corner flapping, b) and c) Edge flipping.

- *Edge flipping*³⁹² is a modification of Corner flapping (Figure 3-1). This new local perturbation mode flaps a pair of neighbouring corners together in opposite directions so that the bond between the two corners changes its rotation. There are six change possibilities, which are illustrated below:



The gauche bond (G) changes its sign (G') or becomes anti (A), while an anti bond (A) transforms into the gauche bond with both signs (G and G'). In action, to prevent the forming of too highly strained structures the movements of corner atoms in a ring structure are carried out according to the rotation patterns of the two neighbouring bonds.

- *Torsion flexing*³⁹³ is a term of a new stochastic (Monte Carlo) procedure that has been developed for the conformational search of cyclic molecules. It produces a local distortion of a ring bond in a cyclic molecule. Due to the fact that this approach does not evoke large atomic movements, following energy minimization processes are usually carried out rapidly even in cases where several bonds are distorted at the same time. The torsional flexing method is able to overcome potential energy barriers so that the subsequent energy minimization often yields another conformer of the cyclic molecule. It is significant that such structural perturbations are in most cases not arduous that avoids rapid minimization process. Figure 3-2 illustrates one example for torsional flexing. It is a local torsional rotation about one ring bond which does not change the atomic position of most of the ring atoms. To describe torsional flexing a torsion angle in a ring $X-A-B-C-D-Y$ are considered in which the ring bond $X-A$, as well as $D-Y$ are imagined to have been cleaved, resulting in two fragments. Both rotations are carried out by random angles. If the $B-C$ bond is a fusion bond between fused rings both rings and sets of atoms participate in rotation process then a rotation around the $B-C$ bond is performed. A (and its substituents) and B (and its substituents) are rotated clockwise while C (and its substituents) and D (and its substituents) are rotated anticlockwise.

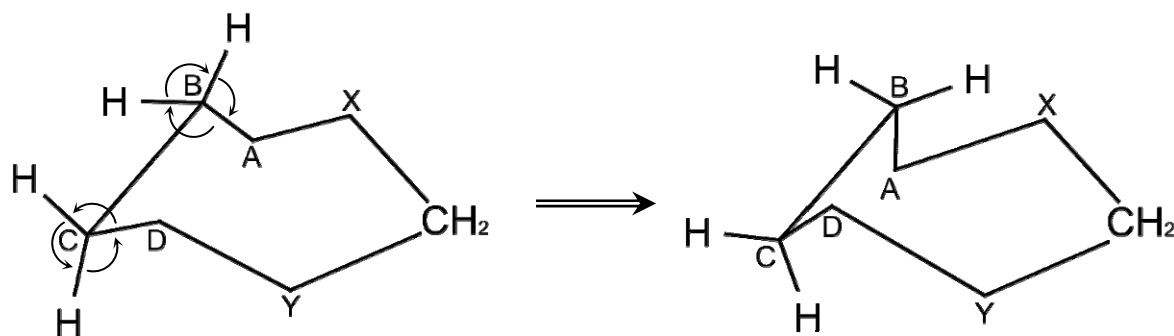


Figure 3-2. Torsional flexing, where the arrows show the direction of torsional rotation. Note that the right side of the resulting molecule structure is remained without changes after the torsional flexing procedure.

The structure is generated by reclosure of the ring. The resulting structure undergoes an energy minimization. During the single torsional flexing operation seven torsion angles are changed and the related bond lengths and bond angles are also modified. The number of torsion angles of torsional flexing operation is increased in the case of fused rings. Since bond length and angles may change during the energy minimization the torsional flexing is not completely a torsional coordinate method. However, the dominant structural modifications of a molecule take place in the torsional space. This approach is an efficient procedure for locating most of the low-energy conformers of a variety of cyclic molecules. Moreover, it is easy to implement and can easily be combined with existing molecular mechanics programs. Using of more than one conformational search procedure increases the guarantee of a complete search of the conformational space available to cyclic molecules. Nevertheless, torsional flexing is a method of choice for conformational search for cyclic molecules.

- *Low-Mode (LMOD) methods*^{309,394,395} are based on the eigenvector-following techniques, which were developed for locating saddle points on molecular potential energy surfaces. The saddle point search is initiated at or near a local minimum. The eigenvalues and the associated eigenvectors called the “normal modes” of the vibrations are determined by the diagonalization of the Hessian matrix. At the minimum the eigenvectors with the smallest eigenvalues point into directions of the potential energy surface with the modest ascent. Assuming that the lowest transition state lies in these directions the basic eigenvector-following (mode-following) idea is to select one or multiple low modes of the starting structure and follow the corresponding Hessian eigenvectors uphill till the energy barrier is crossed. The crossing point is reached if the next step is slightly lower in energy than the preceding one. A subsequent energy minimization will usually lead to the bottom of the neighbouring minimum-energy well. The LMOD procedure starts with an initial molecular model, which is energy minimized. The minimized structure is then subjected to normal mode analysis described above, and a user-specified number of low-frequency modes are stored. LMOD moves in both directions of the Hessian eigenvectors corresponding to the stored low-frequency modes until the algorithm finds a barrier crossing defined by the following criteria:

1. the end-structure of the trajectory is lower than the energy of the starting structure,
- or
2. the structure is at least lower than it was in the previous iteration step and the molecule has also moved further away from the starting structure.

The resulting perturbed initial structure is minimized afterwards. Naturally, it is possible to reach the minimum that is not connected to the initial minimum of the mode-following procedure as a result of energy minimization as well, but substantially, LMOD concentrates the search to the local neighbourhood of a minimum on the potential energy surface. Due to the fact that LMOD always follows the direction computed at the minimum, no re-evaluation of the Hessian has to be performed. A set of conformers, which are used as starting structures for structural perturbation along their low-frequency modes, is accumulated during the search process. In spite of the fact that the new minima found during an LMOD search allow to explore the complete potential energy surface (with the exception of the case of the high-energy conformers search that necessitates the higher frequency modes) LMOD is a systematic search procedure. But it is confined by the number of low-frequency modes considered. LMOD must be switched to a stochastic or Monte Carlo procedure when all possible systematic search directions are already explored for a particular minimum. LMOD was developed originally for macromolecules but it can be also successfully applied to flexible docking and to any kind of molecular systems including complexes. It carries out the search in conformation space of cyclic and acyclic molecules equally efficient. This approach runs neither in the torsion nor in the Cartesian space, nor in any other user-defined search space since it generates its own search low-dimensional search space, which is spanned by the low-frequency mode eigenvectors of the Hessian matrix. Moreover, there is no need for special handling of rings. The advantage of conformational searches performed with LMOD is that it generates structures automatically. Only the user-defined threshold for the low-frequency modes and the energy threshold for energy minimization have to be provided.

- *MOLS conformational search technique*³⁹⁶ is based on the technique of using Mutually Orthogonal Latin Squares (MOLS) to perform enhanced sampling of the conformational space. The main idea of this conformational search technique consists of putting all conformational variables of the molecule to a specific set of values and calculating the potential energy. Obtained values of the potential energy correspond to conformations systematically chosen to sample the entire conformational space. The torsion angles are usually chosen as conformational variables, but other variables are also possible. Each variable in the molecule is capable of taking up some different values in a range. Although this range could be restricted by various factors, in the case of torsion angle space it is taken to be 0–360°. Each cycle of the MOLS procedure consists of four steps as it is shown in Figure 3-3. The set of all possible values of the conformational variables define the complete conformational space of the molecule. All combinations of these values take

part in the sampling, each combination specifying one conformation of the molecule. MOLS is used to pick up such combinations. Each torsion angle is set to correspond to one Latin square to identify the conformations at which the potential energy calculations are to be carried out. It means that a set of MOLS forms by arrangement of the possible values of each torsion angle in the form of a Latin square. The energy hypersurface at these locations in torsion angle space is sampled on the second step. This is achieved by calculating the potential energy at each of the points. Calculation of average potential energy of the each value of the each parameter made on the third step is to recover the energy map of the conformational space. For this purposes one-dimensional representations of the variation of the potential along each of the torsion angles are produced. To estimate the efficacy of setting a particular torsion to a specific value, irrelative of the values of the other torsion angles, the average of potential over points in the MOLS is calculated, where a Boltzmann weighting function is used to identify the optimum value for each variable, and arrive directly at the best structure. The final fourth step is an examination of each one-dimensional representation. Boltzmann weighted averages of different sets of variables are taken to identify the optimum value for each parameter corresponding to the optimal conformation. This completes one cycle of calculations during which one low energy structure is identified. The structure obtained in one cycle of MOLS is either the global minimum energy structure, or one of the various low energy structures of the molecule. To locate another low energy structure the procedure is repeated using a different set of MOLS, however, it is not improbable that the search repeatedly achieves the previously obtained low energy structures. If the solution structure does not improve after a number of iterations it supposes that all the low energy structures of the molecule are identified. The MOLS method outlined above is thus a systematic sampling of the variable space to identify a library of low-energy three-dimensional structures. The search is unconstrained and easily parallelized. This method can be successfully applied to search low energy structures for a variety of small peptides at negligible computational time.

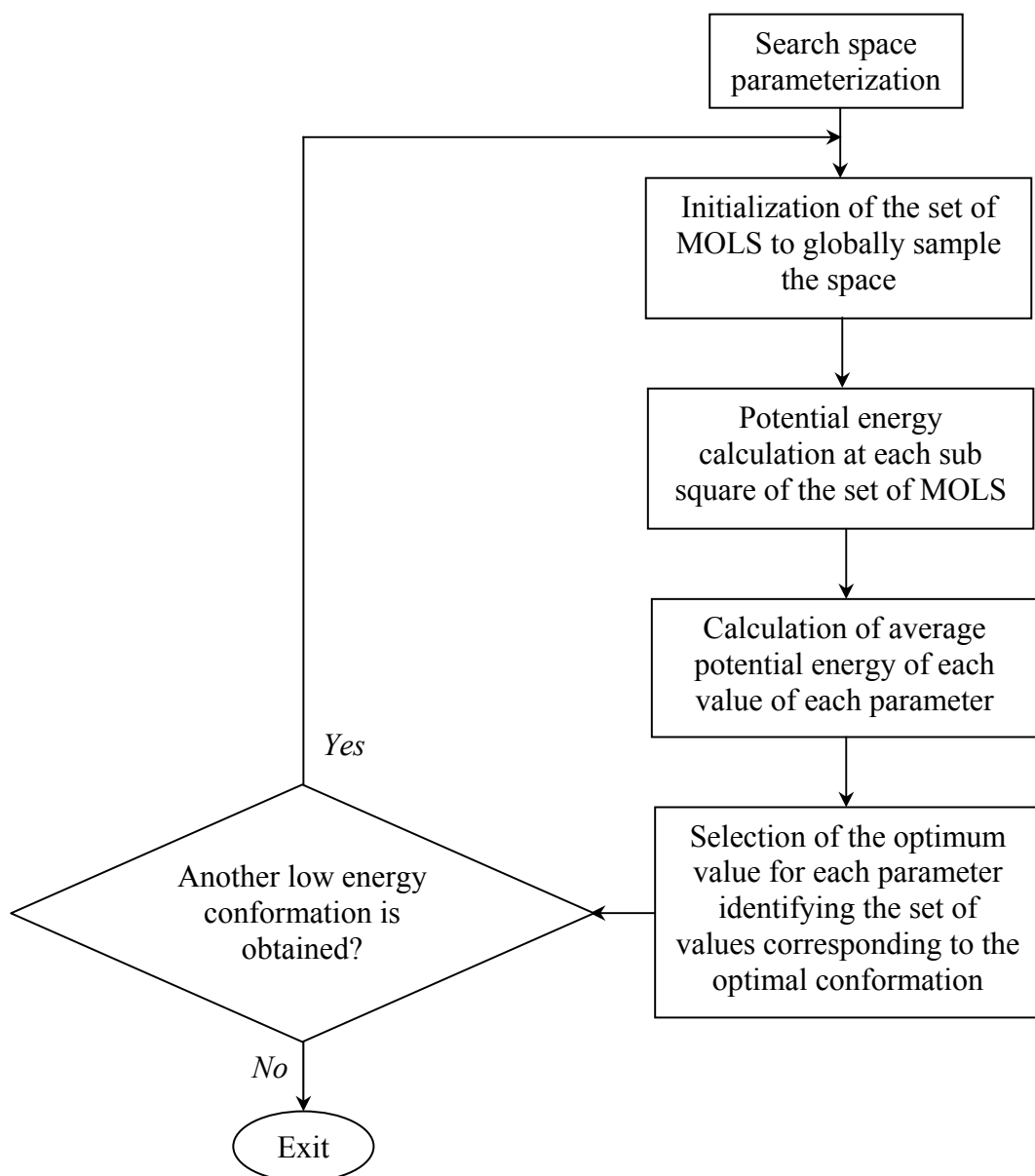


Figure 3-3. Flowchart of the MOLS procedure.

- *Ant Algorithm*⁷⁰ The original ant system, which was already described in the first part of the work, was developed for optimization in the continuous search space of conformational analysis. The conformational analysis problem is formulated as a multimodal function minimization problem in n dimensions, where n is the number of rotatable bonds in the flexible molecule. The aim is to find the global minimum of the function $e=f(\theta_1, \theta_2, \dots, \theta_n)$, where e is the energy of the molecule and $\theta_1, \theta_2, \dots, \theta_n$ are the torsion angles of the rotatable bonds. A roulette wheel selection algorithm is used to generate the torsion angles. The name of this selection method arises since a roulette wheel with slots sized according to the value of the probability function $p(\theta_i, t)=(1-$

$\beta)\tau(\theta_i,t)+\beta\eta(\theta_i)$ is used. $p(\theta_i,t)$ is associated with each torsion angle interval that gives the probability that an ant will choose the value for the i^{th} torsion angle in iteration t of the algorithm. β is a parameter between 0 and 1. $\tau(\theta_i,t)$ corresponds to the pheromone trail and is updated in each iteration of the algorithm. $\eta(\theta_i)$ corresponds to the visibility and keeps constant. Artificial ants move at n intervals between 0° and 360° , and the path each ant follows in one iteration of the algorithm determines the configuration of the molecule. The pheromone trail is updated in each iteration of the algorithm. At each iteration all ants provide data about conformations of the molecule, and for each ant the corresponding strain energy is calculated using the molecular mechanics. It was tuned and tested for the conformational analysis of flexible drug-like molecules, with the objective of obtaining an accurate estimate of the lowest possible conformational energy.

3.2 GOTS application

On the basis of our tests the GOTS turned out to be the most efficient of the developed algorithms. Hence, it was adapted for conformation searches. For this purpose some modifications of the algorithm and a smart definition of the variables used for the GOTS were necessary. These adaptations and first applications are described in the following.

3.2.1 Variable choice and numbering rules

The choice of the coordinates used to describe a molecule is critical for the efficiency of the optimization. Since gradients and Hessians are usually calculated in Cartesian space, Cartesian coordinates would be the most straightforward choice. But without the initial curvature information, optimization in Cartesian coordinates is extremely inefficient, especially for large systems³⁹⁷. Additionally, many studies have shown that Cartesian coordinates are generally outperformed by a well-chosen set of so-called “natural internal” coordinates³⁹⁸ - consisting of bond lengths, bond angles, and torsions. Internal coordinate systems can be classified as redundant or nonredundant. In a nonredundant internal coordinates system, the number of coordinates is equal to the number of internal degrees of freedom $3N-6$ (where N is the number of atoms in the system); in a redundant set the number of coordinates exceeds the number of internal degrees of freedom. So internal coordinates are frequently used to represent conformations of molecules. These coordinates, first introduced into geometry optimization by Pulay et al.³⁹⁹, involve the use of individual bond displacements as stretching coordinates and linear combinations of bond angles and torsions as deformational coordinates.

The Z-matrix coordinates are a typical example of nonredundant internal coordinates⁴⁰⁰. They describe the molecular geometry in terms of individual distances, angles, and torsions. Each line of a Z-matrix gives the internal coordinates for one of the atoms within the molecule. The Z-matrix format used in the Gaussian program package uses the following syntax: Element label, atom 1, bond length, atom 2, bond angle, atom 3, and dihedral angle. As an example the Z-matrix of n-Butanol is shown in Table 14. It provides a description of each atom in a molecule in terms of its atomic number, bond length, bond angle, and dihedral angle.

Table 14. A part of the internal coordinates in Z-matrix form of the n-Butanol molecule.

Element label	Atom 1	Bond length	Atom 2	Bond angle	Atom 3	Dihedral angle
c
c	3	cc4	2	ccc4	1	dih4
...
h	2	hc9	3	hcc9	4	dih9
h	2	hc10	3	hcc10	4	dih10
...

The illustration of the n-Butanol molecule is presented on Figure 3-4.

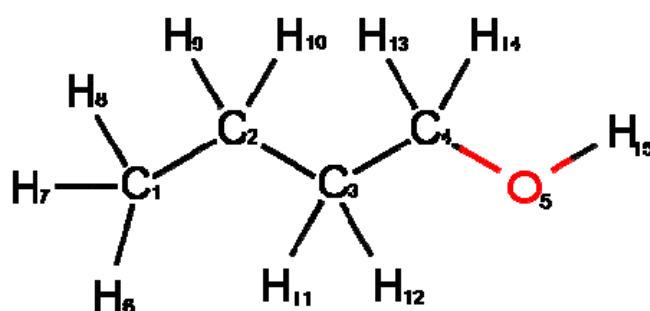


Figure 3-4. Atom numeration of the n-Butanol molecule.

In difference to other Z-matrix-like coordinates normally used in the literature^{401,402,403,404,405}, instead of altering of a bond length, a bond angle, and a dihedral angle of each atom, only dihedral angles are used in the present work since the bond length and angles of different conformers vary relatively little. Thus, they are defined to be stiff. If Z-matrix coordinates are used for optimizations, in particular for ring systems, it is difficult to find which an appropriate set of internal coordinates for the optimization process. Problems

occurs that various dihedral angles have to be changed together to perform reasonable rotation. To rotate the terminal methyl group consisting of atoms $C_1H_6H_7H_8$ around the C_1C_2 axis it is necessary to modify three dihedral angles ($C_3C_2C_1H_6$, $C_3C_2C_1H_7$, $C_3C_2C_1H_8$) although there is only one rotation. A crucial point is that these torsion angles must not be varied independently of each other. To demonstrate it on the example lets take a typical conformation of the molecule and attempt to move only one of dihedral angles at a time while keeping the rest of the variables constant. Moving $C_3C_2C_1H_6$ only while keeping $C_3C_2C_1H_7$, $C_3C_2C_1H_8$ constant distorts the internal structure of the methyl group. It is necessary to move various dihedral angles synchronously to achieve proper rotations around a freely rotatable single bond. The approximate separation of hard and soft movements of the system allows to create the structure of the molecule by setting the hard coordinate to constant or to particular dependence on the soft coordinates^{406,407,408}. For this one partition the angles in hard and soft dihedral angles is carried out. The soft ones are varied while the hard ones are changed accordingly or keep constant. The problem was recognized and the concept of “*related dihedrals*” was introduced by G. A. Chass⁴⁰⁹.

In this work the rules proposed by Echenique and Alonso⁴¹⁰ are used to account for this problem. To perform a rotation around a given bond only the “*main torsions*” are independently varied. The variations of the “*dependent torsions*” are not free but make sure that proper rotations take place. In Table 15 the Z-matrix from Table 14 is reorganized relative to the angle **(4,3,2,1)**. This angle is a “*main torsion*” while the other two are “*dependent torsions*”.

Table 15. Information from Z-matrix of the n-Butanol molecule.

Atom number	Atom name	Bond length	Bond angle	Dihedral angle
...
4	C	(4,3)	(4,3,2)	(4,3,2,1)
...
9	H	(9,2)	(9,2,3)	(9,2,3,4)
10	H	(10,2)	(10,2,3)	(10,2,3,4)
...

To ensure a proper behavior the following scheme is used:

1. Write all dihedral angles of the Z-matrix into the list called “*torsions*”. Empty the lists: “*main torsions*” and “*dependent torsions*”.
2. The first dihedral angle of the Z-matrix is chosen and written in the list “*main torsions*”.
3. Take the next dihedral angle from the “*torsions*” list.

4. Compare it with all previously found “*main torsions*”. If the second and third atoms of the dihedral angle under consideration agree to those of a “*main torsions*” it is a “*dependent torsion*”. It is related to the “*main torsion*”. If no agreement exists, it is a next “*main torsion*”, and is written to the main torsions list.
5. If the “*torsions*” list is empty, the selection is finished. Otherwise, the process go back to step 3.

It is significant that the vector direction does not count in this case (see Figure 3-5).

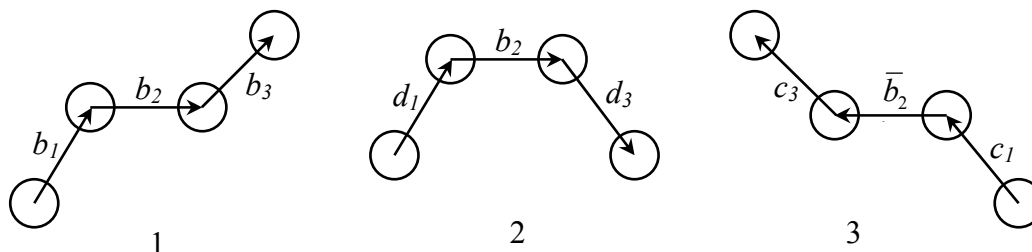


Figure 3-5. 1, 2 and 3 are related dihedral angles defined by three bond vectors connecting four atoms.

If the scheme is applied to the n- Butanol molecule (Table 14) Table 16 is obtained.

Table 16. “Main torsions” and “dependent torsions” of the n-Butanol molecule.

“Main torsions”	“Dependent torsions”
(4,3,2,1)	(9,2,3,4)
	(10,2,3,4)
	(11,3,2,1)
	(12,3,2,1)
(5,4,3,2)	(13,4,3,2)
	(14,4,3,2)
(6,1,2,3)	(7,1,2,3)
	(8,1,2,3)
(15,5,4,3)	-

Since only the “*main torsions*” are varied independently the number of degrees of freedom decreases.

3.2.2 Adaptation of the GOTS to the conformational search

To adopt the GOTS approach for the conformation search some modification were implemented. Its starts with the local minimization approach which is carried out as geometry optimization of the computational chemistry environment ChemShell using the Universal Force Field (UFF)⁴¹¹. To solve conformation search problems the variable choice is carried

out as indicated above. For the GOTS³³⁸ only function evaluations should be used to estimate step size and directions. For the first step of the modest ascent all functional values Fz_i^+ and Fz_i^- (Chapter 2 - Eq. 11) are computed using ChemShell single-point calculations to determine the general direction of the following moves. They are obtained by varying each dimension by a user defined step size Δx_i^0 . Step size and direction are estimated according to Chapter 2 - Eq. 14.

As described in the second chapter a ranking procedure of the internal degrees of freedom is used to find the optimal direction for the modest ascent. Applying this ranking procedure to the conformational search it turned out that the computed energy differences are too small. To obtain a reasonable differentiation between the various degrees, dynamical calculations of the ranking coefficients $rank_{max}$ and $rank_{min}$ are necessary. It means that $rank_{max}$ and $rank_{min}$ are not set to 0 and 1 but calculated for each case separately using the following approach (Eq. 16-19):

$$coefficient_{min} = \frac{Fz_{max} \cdot 100}{Fz_{min}} - 100 \quad (16)$$

$$rank_{min} = \frac{100 - coefficient}{100} \quad , \text{ if } 0 < coefficient_{min} < 100 \quad (17)$$

$$rank_{min} = 0.1 \text{ (or 0.0 - depends on problem)} \quad , \text{ otherwise}$$

$rank_{min} = 0.1$, value difference is large, therefore we have to redefine $rank_{max}$ and after that we use the formula (Chapter 2 - Eq. 14) to calculate $rank_i$
 $rank_{min} \neq 0.1$, value difference is small, we directly use the formula (Chapter 2 - Eq. 14) to calculate $rank_i$

To calculate $rank_{max}$ the same percentage analysis as for $rank_{min}$ is made but for each value:

$$coefficient_{max} = \frac{Fz_i \cdot 100}{Fz_{min}} - 100 \quad (18)$$

$$rank_{max} = \frac{100 - coefficient}{100} \quad , \text{ if } 0 < coefficient_{max} < 100 \quad (19)$$

$$rank_{max} = 1.0 \quad , \text{ otherwise}$$

The resulting modest ascent is followed until the next calculated function value is smaller than the previous one. This indicates that the barrier to the next valley is crossed. From this point the next local minimum is located using local minimization.

Other neighborhood strategies were also implemented and tested. In the first variant (see Chapter 2 - Eqs. 11-13) not only the preferential direction determined at the minimum is tested but at each step of this neighbourhood search all functional values Fz_i^+ and Fz_i^- are computed to determine the direction of the next move (Chapter 2 - Eq. 11, Figure 3-6). This is different to the standard approach in which the main promising moving area is determined at the local optimum and kept fixed during the modest ascent strategy. The moves are carried out with the step sizes for each variable that are estimated according to the linear ranking procedure. It turned out that the selected moves now often lead to areas close to already visited solutions. In consequence, the search starts to cycle and needs a considerable higher number of function evaluations to come to the next valley. Such ideas were rejected since this increases the computational costs without having many advantages.

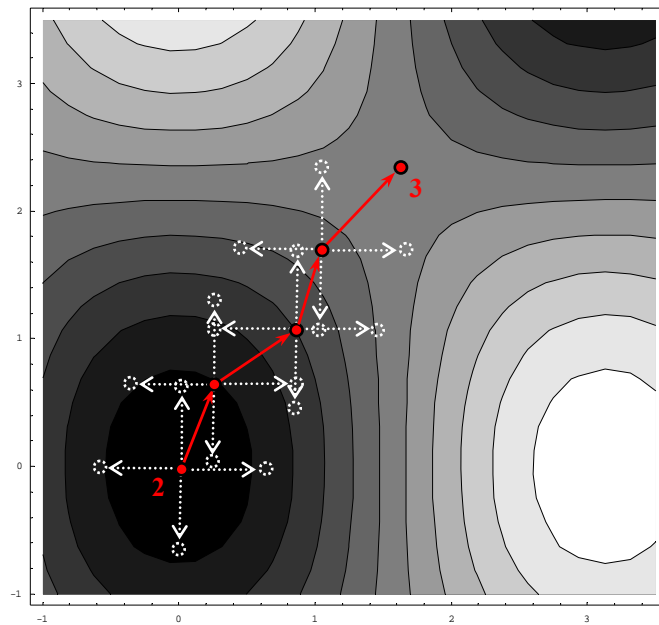


Figure 3-6. Illustration of neighbourhood strategy with testing all direction at each step.

Ranking and the neighbourhood strategies were also considered. In one version the ranking procedure was skipped, i.e. the move is carried out along the coordinate with the best function value (see Figure 3-7b). However, also in this case more steps are needed to escape from a valley. In another approach (Figure 3-7) a number of steps are tested for the selection of the most promising move. This included steps with ranking and steps without ranking. Furthermore, also small additional steps in positive and negative direction were performed after the step without ranking. Comparing the functional values of all these steps, the one from the step without ranking was often the best one. This means that the outcome of this version is closed to the previous one. Based on test results the original neighbourhood strategy turned out to possess the best cost-benefit ratio.

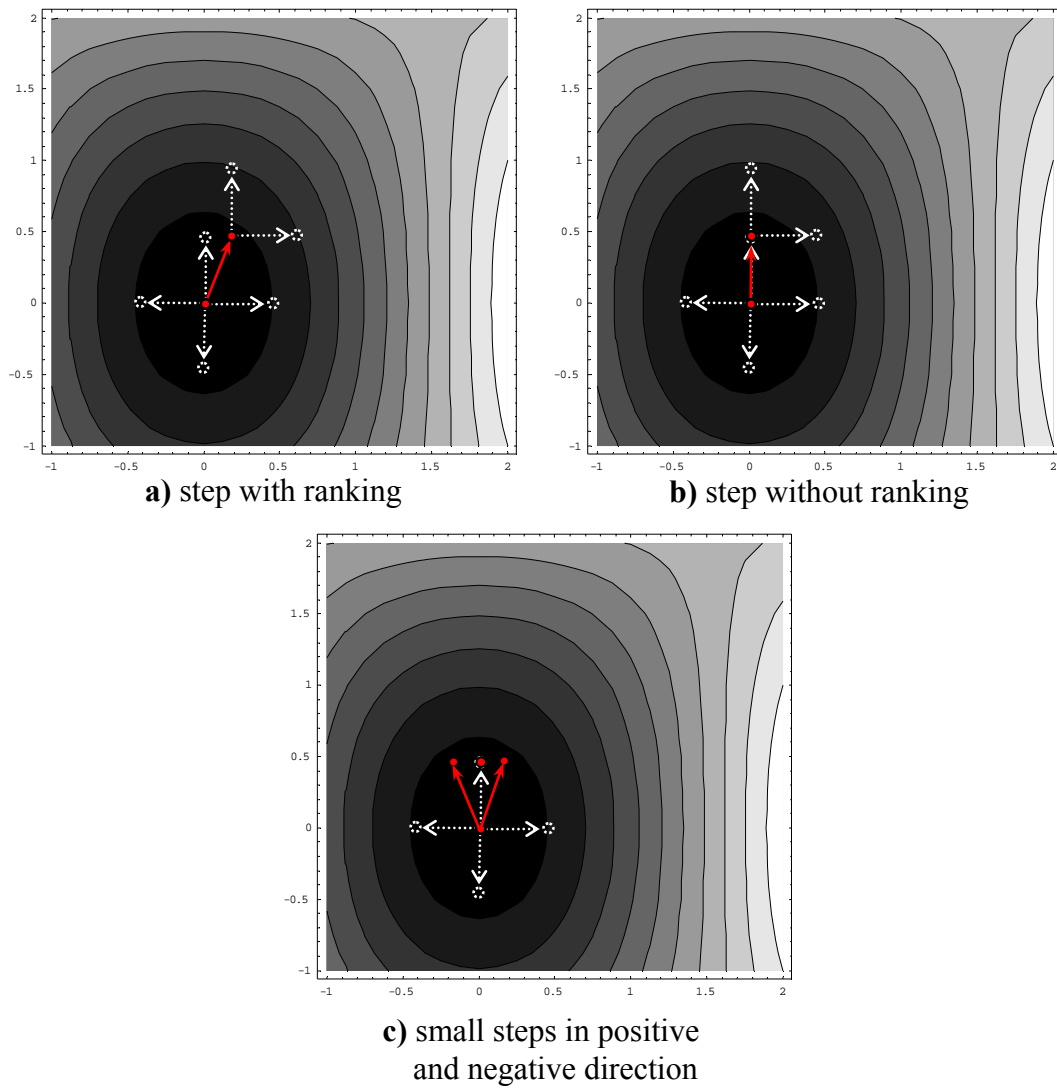


Figure 3-7. Illustration of the neighbourhood search with various kinds of steps.

The TS is a mathematical optimization method, which belongs to the class of local search techniques. To include more probability aspects, ideas from the Simulated Annealing (SA) method have been implemented. It shall help to focus on promising areas and to locate a good approximation to the global optimum of a given function in a large search space. In our approach, the SA ideas are exploited to determine if a new minimum is taken or not. If the new minimum is lower in energy it is always taken. If it is higher than the previous one the criteria used in the SA (Chapter 1 – Eq. 6) are taken to decide if it is taken as the new starting point or not. If the new minimum is not accepted the algorithm returns to the previous minimum and continues the search along the next modest ascent direction.

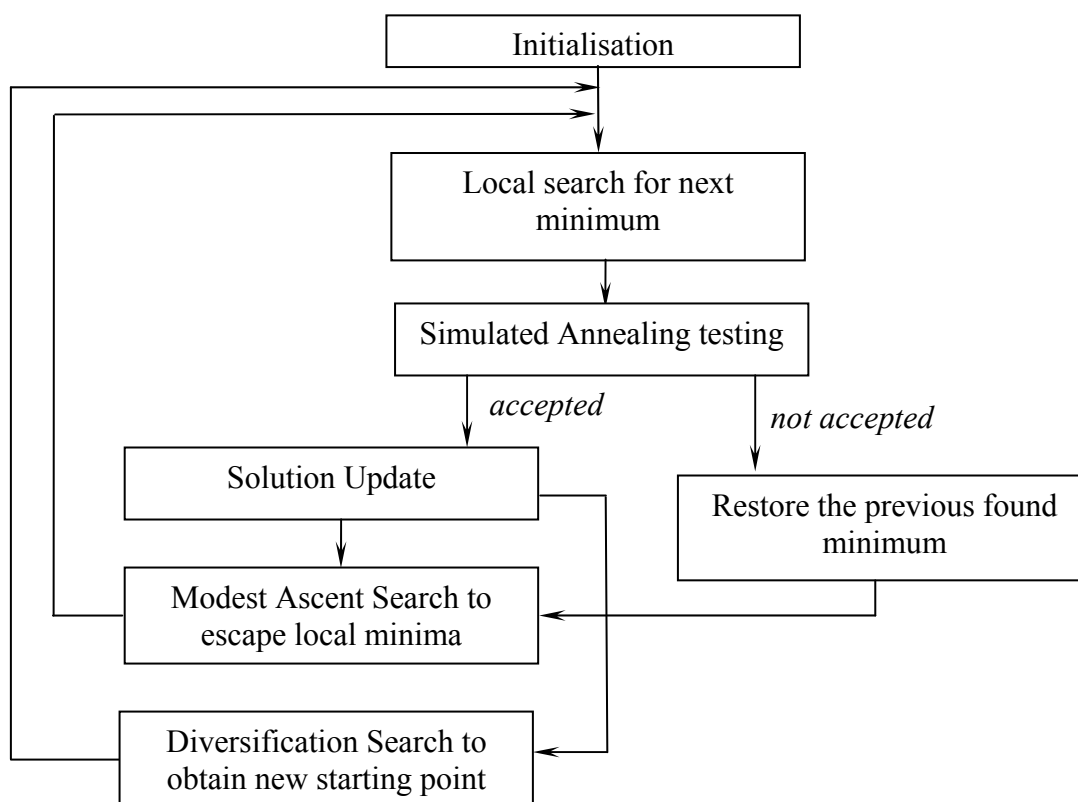


Figure 3-8. Flowchart of GOTS with SA elements.

To avoid reverse moves and cycles within the Tabu Search strategy a Tabu List (TL) is built up which memorizes the moves previously made. The elements of the TL are ordered according to the chronology of their appearance. If the TL is full the newest solution replaces the oldest one employing the FIFO principle. For approaches which move in small steps through the function $F(x_i)$ a simple storing of all visited points is sufficient to block already visited regions effectively. To overcome the problems resulting from making large steps Tabu Regions (TR) are used which were already successfully employed in the GTS and TSPA methods and which are related to ideas introduced by F. Glover. Tabu Direction (TD)³³⁸ turned out to be less efficient for the conformational search as a result of the circular motion of the the torsion angles from 0° to 360° . Examples of this are presented in Figure 3-9. The start point is 135° . The step size is 90° . The first movements are shown in part a) at the Figure 3-9. After these two steps the TL values are 135° , 225° , 315° . Vectors $\vec{2,1}$ and $\vec{3,2}$ are TDs. However, the next steps are already problematic if the TD approach is used. It is illustrated with the parts b) and c). In the case of selecting 360 as the number of degrees the next solution is 45° (see example b)). This unvisited solution is set “tabu” through the TD because the motion vector $\vec{3,4}$ coincides with the TD vector $\vec{3,2}$. Such approach leads to exclusion of

some unvisited solutions from the search. In the case of the unbounded number of degrees (see example c)) all subsequent solutions (405°, 495°, 585°, 675°, and etc.) are allowed because there are no such values in the TL. It results in cycling of the search process to the rotation of a single torsion angle. So, the solution is “tabu” if all of the variables are in the TR.

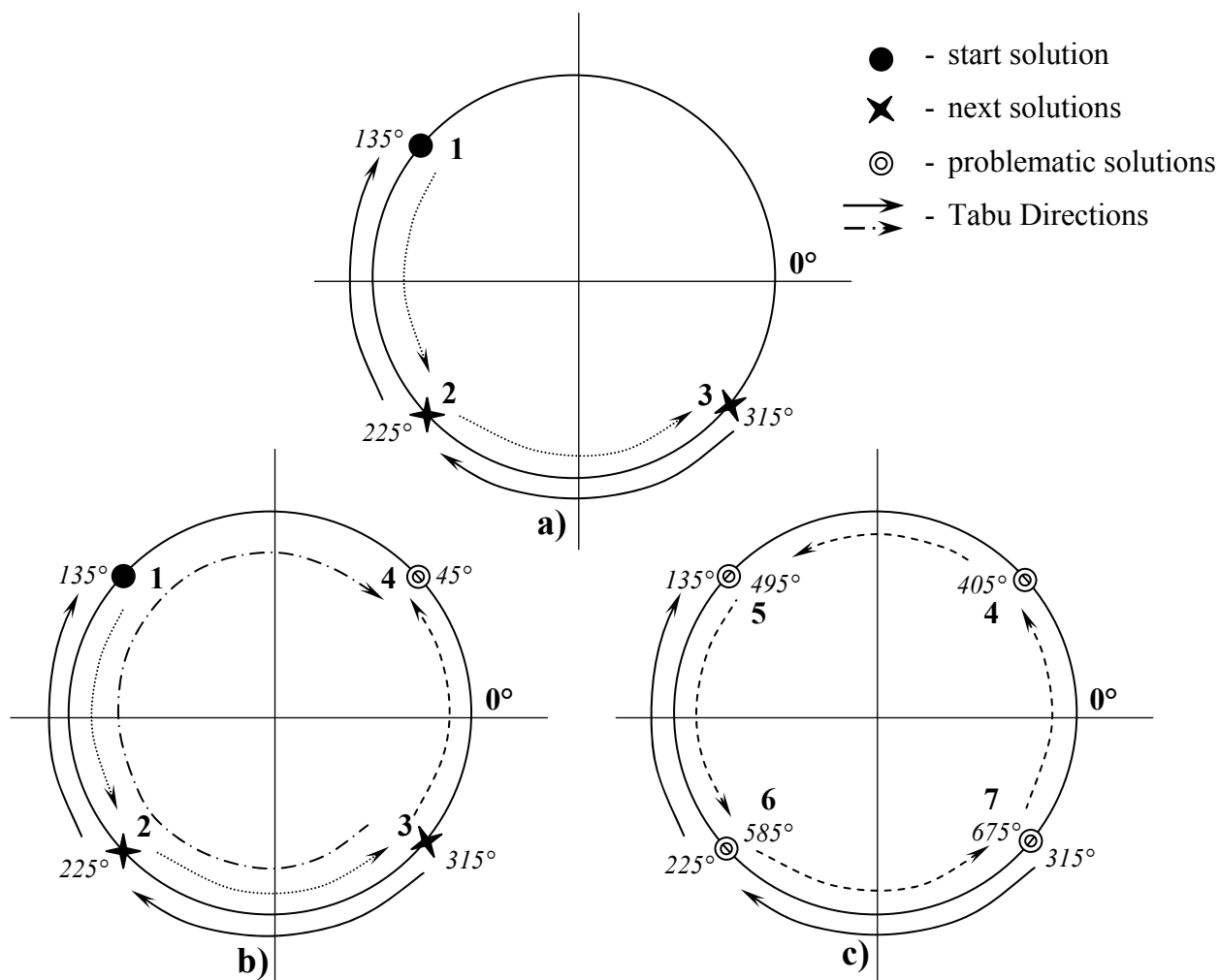


Figure 3-9. Rotation of the torsion angle.

The last element of the TS represents the diversification search (DS) which is used to select new starting points in the solution space. A DS becomes necessary if the solution does not improve after a number of iterations or if all neighborhood solutions are already set tabu. One way to enhance the convergence is to ensure that the search switches to regions that were not already investigated. In the present approach the following strategy is used. There are $N-3$ (where N is the number of atoms in the system) “main and dependent torsions”. “Dependent torsions” have to be changed together with the “main torsions”. To find a new starting point the “main torsions” are changed with steps of 30°, 60° or 120°. The rest of the variables are kept constant. To obtain reasonable geometries all “dependent torsions” are moved

accordingly. All points are excluded which already belong to Tabu Regions and the new search starts from the point with the lowest $F(x_i)$ value.

Figure 3-8 gives the flowchart of the GOTS method with SA elements. At first a start molecule structure in Z-matrix form is initialized. After that one starts with the search for improved minima. This part consists of local optimizations that are applied to obtain nearest local minima and the modest ascent searches used to escape to the next valley. The simulated annealing criterion is used to confirm or reject a new local minimum as the new starting point. In the case of rejection of a current minimum one returns to the previously found minimum solution and starts the search along the second promising direction based on the already calculated ranking coefficients. For the modest ascent search weighted function values can be used. In between an update of the solution vector and the Tabu List is performed. If the solution does not improve after a given number of iterations or if all neighborhood solutions are already set *tabu* the search for improved minima is aborted and the diversification search is performed to obtain new starting points.

3.2.3 Comparing the influence of different parameters

The proposed algorithm has several parameters. Some of them were already discussed in chapter 2. For the application of the method to conformational searches some of them are especially important for the fast convergence and efficiency of the optimization. The standard values have also to be adapted to the conformational search. The most important parameters are reported in Table 17.

Table 17. Description of the parameters of the approaches.

Parameter	Purpose	Recommended values
Δx_i	step size at the mildest ascent strategy	45°
<i>BADMAX</i>	number of not improved minima after which a DS is performed	5
N_{trial}	number of trial solutions for the diversification search	depend on number of “main torsions”
<i>Diversification step</i>	step size used during the diversification strategy	frequency 60°
$rank_{max}$	default maximum recency ranked value	1.0
$rank_{min}$	default minimum recency ranked value	0
<i>TR</i>	½ of the Tabu Region diameter	10°
<i>T</i>	control parameter (analog of temperature)	>> ΔE normally encountered

The parameter Δx_i gives the step size during the mildest ascent search. One could assume that a small step size increases the accuracy, but this is misleading. For small step sizes a wrong mildest ascent direction is taken. This is due to the fact that the energy differences between the directions are so small that the ranking procedure does not work anymore. Based on our experiences $\Delta x_i = 45^\circ$ is recommended as a standard value. If even larger step sizes are used fewer steps are necessary to leave a local minimum, but sometimes minima are missed.

The *BADMAX* value was set to 5. This parameter also plays an important role during the optimization process. Together with the Simulated Annealing approach the parameter *BADMAX* controls when the search switches to the diversification search, i.e. it aborts the search in the current region. If the solution does not improve after *BADMAX* local minima a diversification search is started to select new starting points in the solution space.

The global time-varying parameter T , with which a higher lying minimum is taken as the new starting point, influences the probability (see Eq. 5-6). T , called the temperature, is gradually decreased during the optimization process. To determine this parameter experimentation is required. The starting T value is set considerably larger than the largest ΔE normally encountered and decreased by 10 percent with each new found minimum. The dependency is such that a new local minimum solution is practically always accepted when T is large even if it is quite high with respect to the last one. If T goes to zero the probability that higher lying minima are accepted as new starting point decreases considerably.

During the diversification search steps equal to 60° were used. It means that a new start solution is obtained by changing all “*main torsions*” by 60° , 120° , 180° , 240° or 300° in random consecution. The “*dependent torsions*” are changed accordingly. The parameter N_{trial} is also connected with the diversification search. In our test cases N_{trial} was set to the number of “*main torsions*”.

The parameters $rank_{max}$ and $rank_{min}$ are ranking parameters from the linear ranking procedure (Eqs. 14-19). They are calculated dynamically in the range from 0.0 to 1.0 that means separately for each case.

The parameter TR controls the size of the tabu regions within the whole optimization process. One would expect that small TR do not efficiently block already visited regions so that the effort decreases with increasing TR . However, in most cases the effort increases for larger TR . This counter-intuitive behaviour may result because the tabu regions become so large that minima lying close by already visited points are overlooked or because the optimal path to the global minimum is blocked. A value of $TR = 10^\circ$, however, seems to be a good

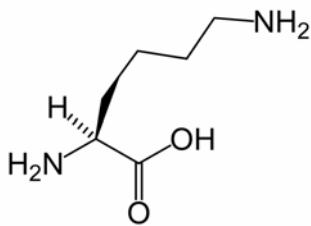
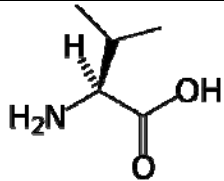
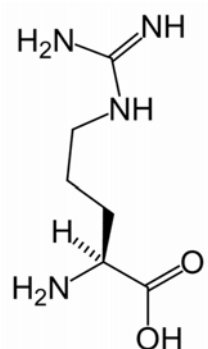
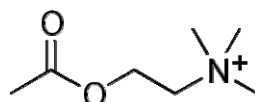
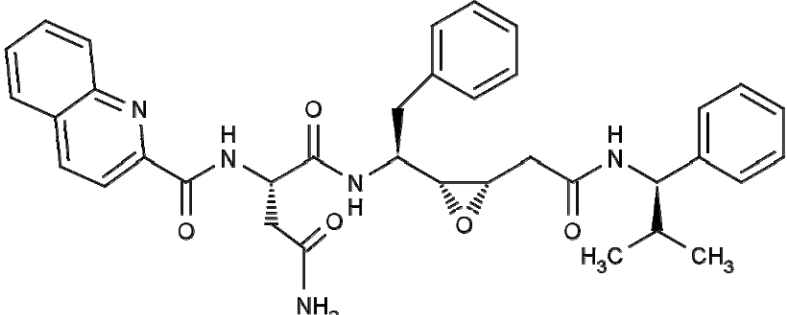
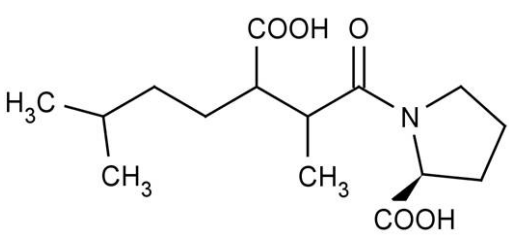
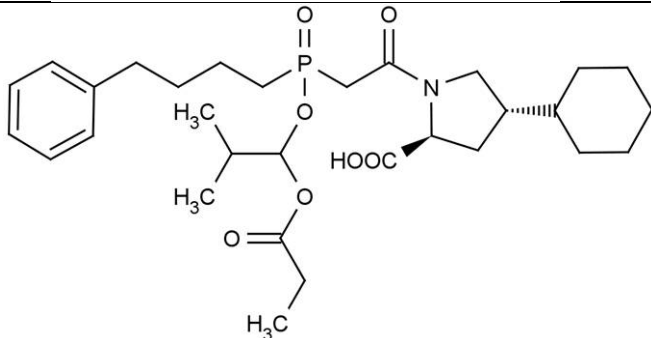
choice for a large variety of different problems. So, if the new solution variables differ from the variables of the already visited solution within $\pm 10^\circ$ it consider as a *tabu* solution.

3.3 Experimental results

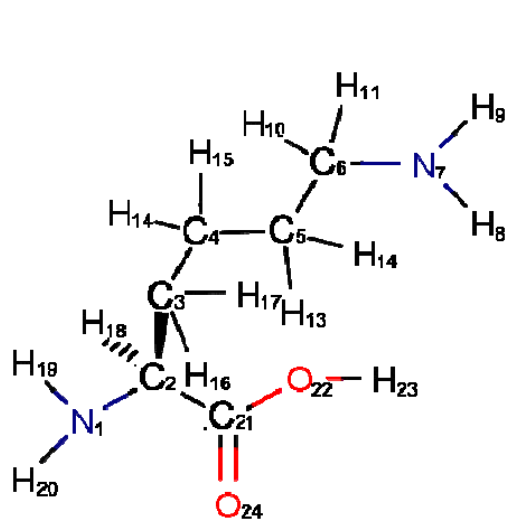
The developed GOTS was applied to detailed conformational studies of Lysine (LYS), Valine (VAL), Arginine (ARG), 2-acetoxy-N,N,N-trimethylethanaminium (Acetylcholine). In addition two ACE inhibitors and a HIV-1 protease inhibitor were included. All these molecules are presented in Table 18. LYS has 21 dihedral angles, ARG - 23, VAL - 16, Acetylcholine - 23, the first ACE inhibitor - 43, the second ACE inhibitor called Fosinopril - 82 and HIV-1 protease inhibitor - 82. According to our definition of “*main torsions*” and “*dependent torsions*” the number of variables decreases and there are only 8 “*main torsions*” in the case of LYS molecule, 11 for ARG, 6 for VAL, 8 for Acetylcholine, 17 for the first ACE inhibitor, 34 for Fosinopril and 41 for HIV-1 protease inhibitor to optimize.

As noted above, the program employs the computational chemistry environment ChemShell under Linux for the local geometry optimization. To determine function values the single-point calculations were carried out. The Universal Force Field (UFF)⁴¹¹, a molecular mechanics force field, was used in the GOTS procedure. The parameters used to generate the UFF include a set of hybridization dependent atomic bond radii, a set of hybridization angles, van der Waals parameters, torsional and inversion barriers, and a set of effective nuclear charges. The ability of UFF to correctly reproduce the experimental rotational barriers and conformational energies of a variety of organic molecules was investigated by Casewit, Colwell, and Rappé⁴¹². Good agreement with experiment was observed when UFF was applied to the conformational equilibrium of compounds for which charge evidently does not play an important role.

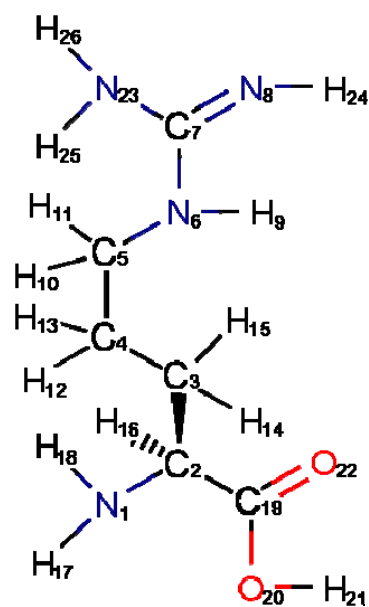
Table 18. List of the test molecules.

Lysine (LYS)	
Valine (VAL)	
Arginine (ARG)	
Acetylcholine	
Peptidomimetic HIV-1 protease inhibitor	
Inhibitor of angiotensin converting enzyme (ACE)	
Fosinopril	

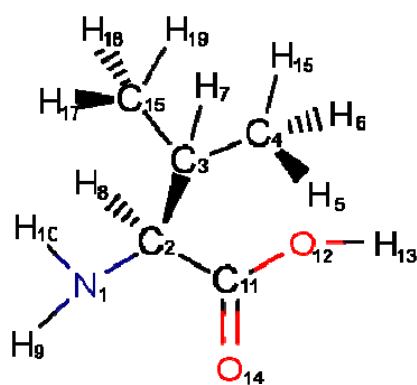
The atomic numberings used in our test runs are shown in Figure 3-10.



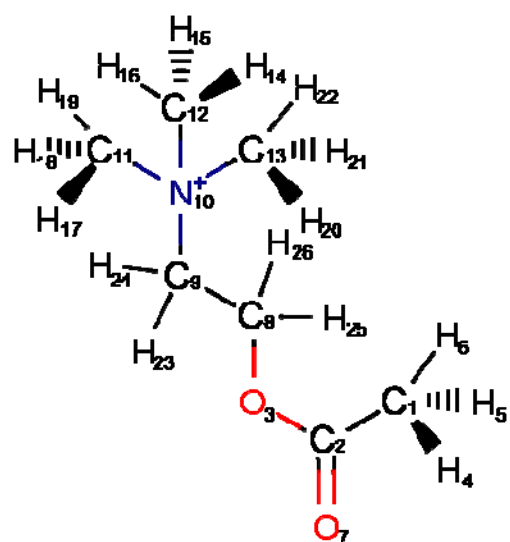
LYS



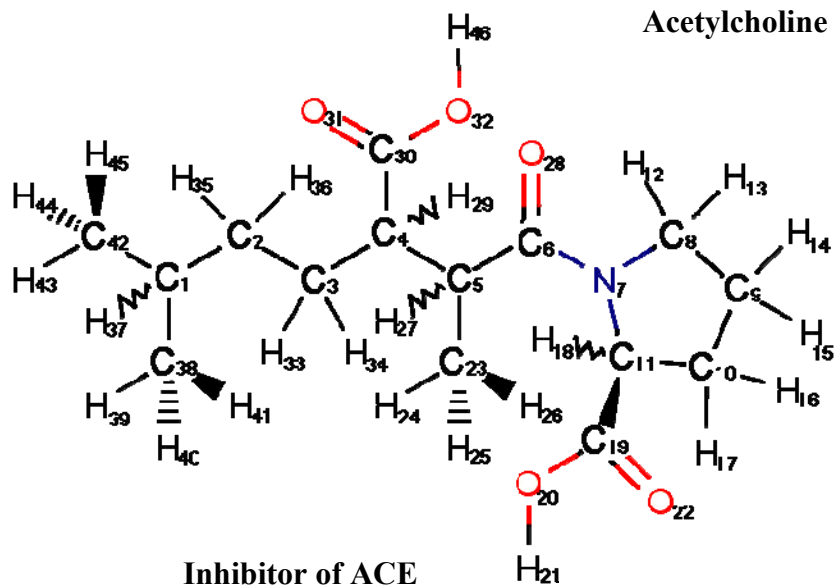
ARG



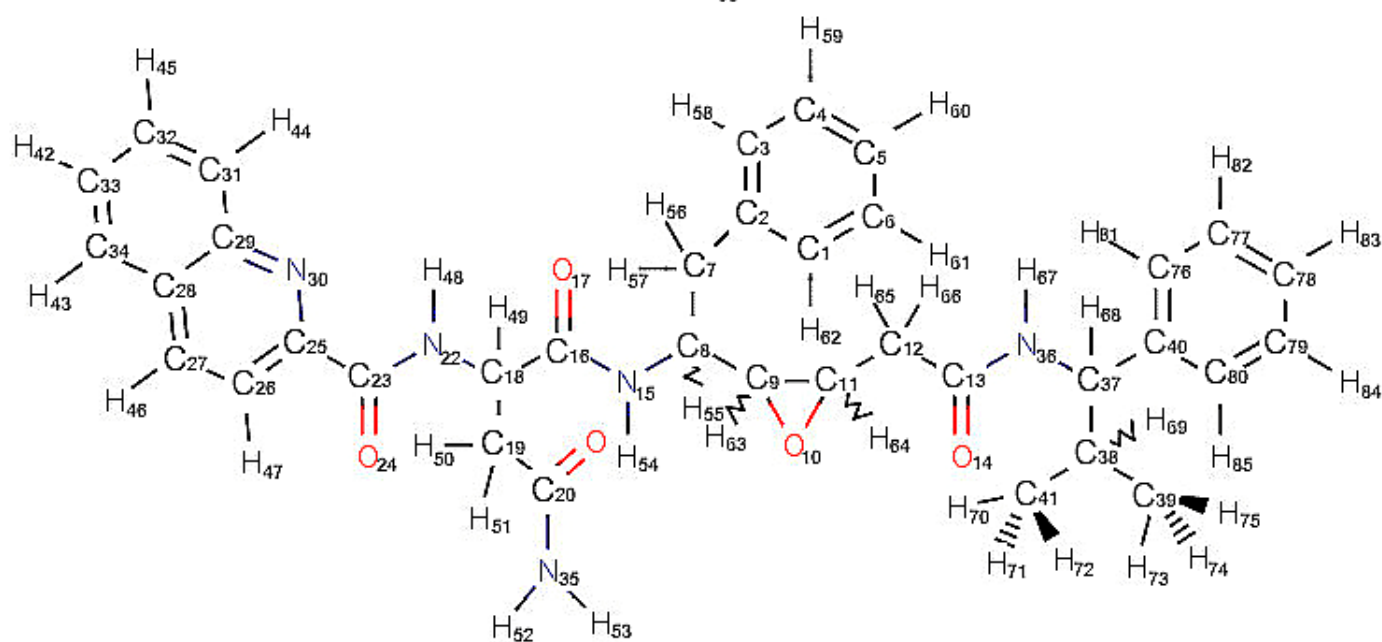
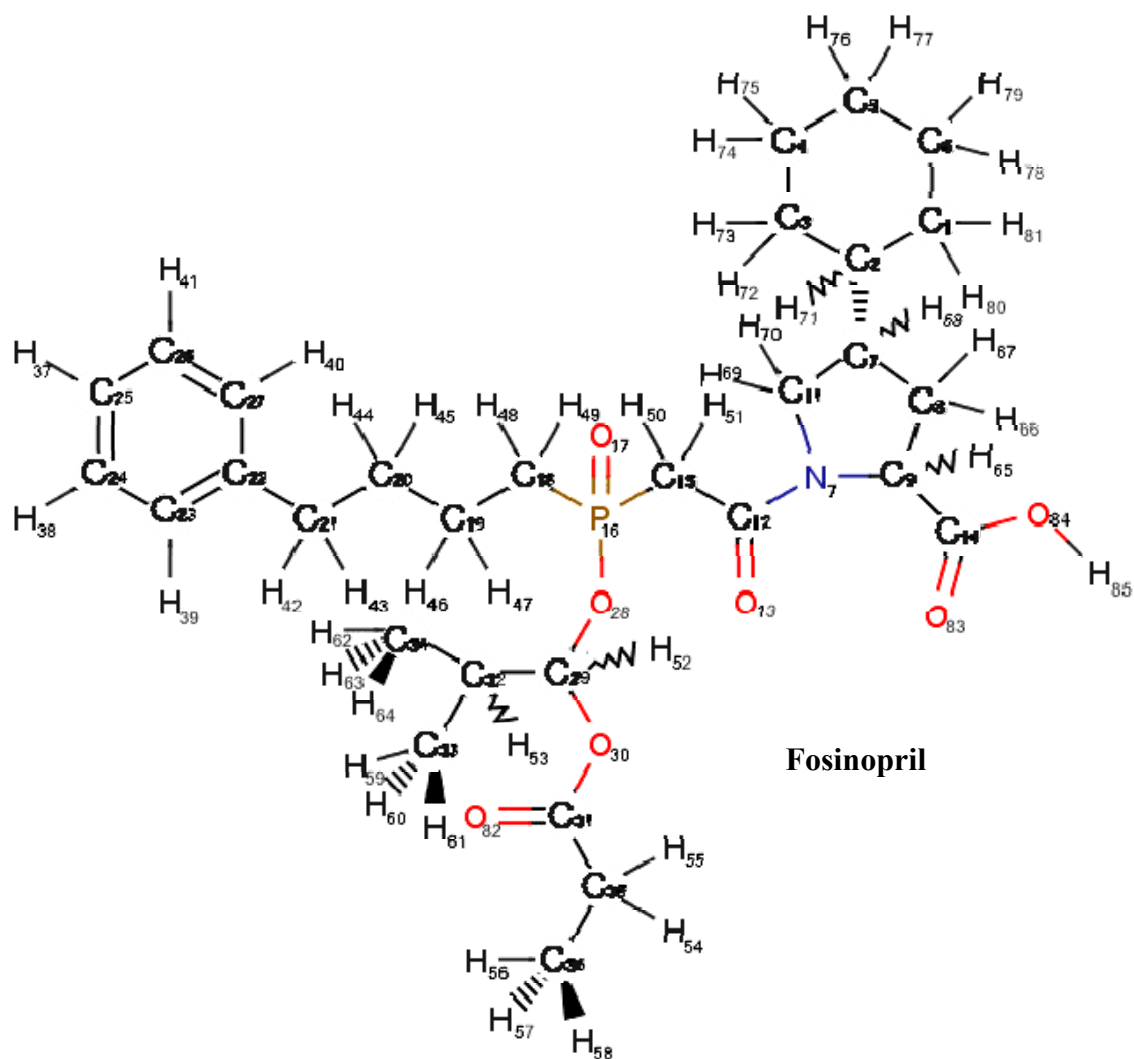
VAL



Acetylcholine



Inhibitor of ACE



Peptidomimetic HIV-I protease inhibitor

Figure 3-10. Used numbering systems.

3.3.1 Conformational studies of amino acids

In the structure shown in the Figure 3-11, **R** represents a side chain specific to each amino acid. The central carbon atom is a chiral carbon atom (except for glycine) to which the two termini and the **R**-group are attached.

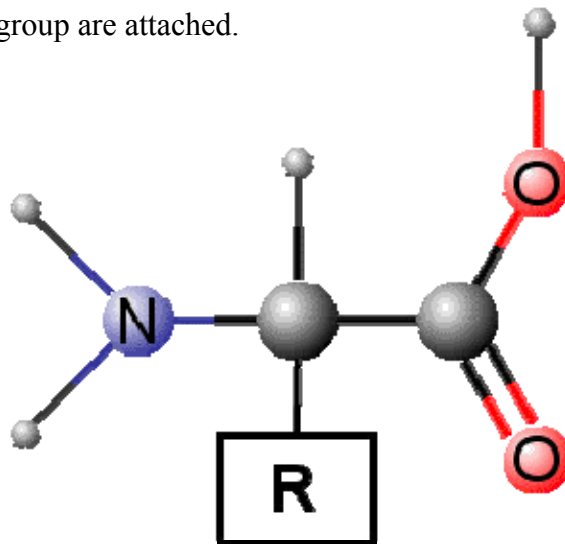


Figure 3-11. The general structure of a α -amino acid, with the amino group on the left and the carboxyl group on the right.

Peptides, and their building blocks, the amino acids, possess many conformers each of them differs in the location and type of bend in the backbone, and in the conformational variability space in the side chains. The similarity of all amino acids, they differ only in the **R**-group, can be used to simplify the conformational search for these molecules. This can be realized by the division of the molecule structures into parts. One of the parts is common for all amino acids while another one includes the rest **R**. The divisions for molecules LYS, ARG, and VAL are indicated in Figure 3-12.

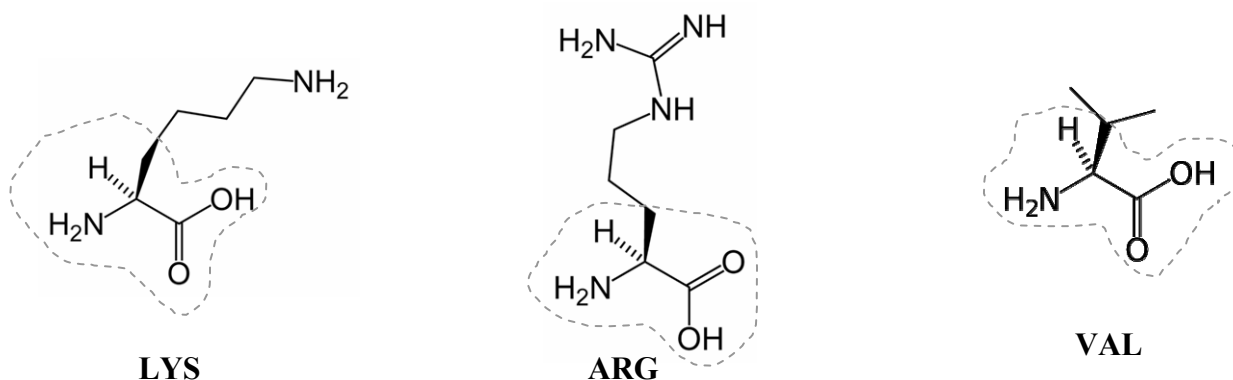


Figure 3-12. Division used in the conformational search of amino acids.

The conformational search is started for the common part. The best solution of this is kept constant within the modest ascent part of the search. During the local search for the next minimum all variables take part, i.e. local optimizations are carried out without any restriction. Variable dihedral angles for the first and second part of the whole optimization process and for the optimization without parting are demonstrated with the example of LYS (see Figure 3-13, where all unmarked torsions keep constant during modest ascent search).

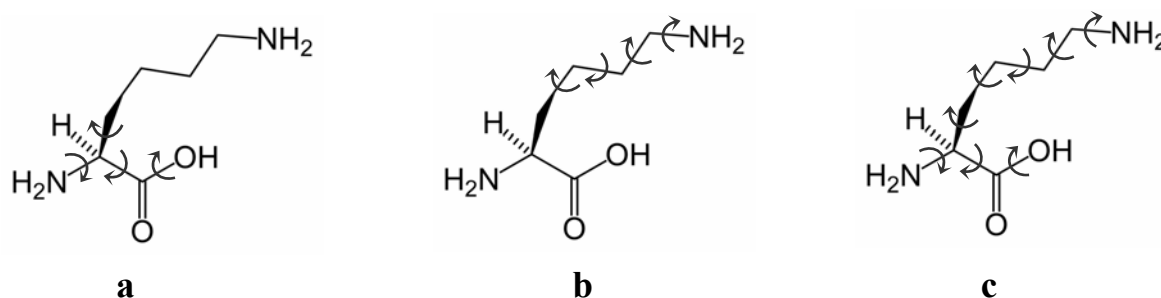


Figure 3-13. Variable torsions a) of the first part b) of the second part c) employed in local search.

To proof the stability and efficiency of the GOTS method for conformational search tasks, test runs were made from four different starting structures (with the exception of VAL because of its size). Dihedral angles were modified to obtain the various start structures of the molecules. The results are given in Table 19. It convincingly demonstrates that partings of amino acids increase the efficiency of the optimization search. Additionally, better optimal structures are obtained since less number of variables participates in modest ascent strategy.

Table 19. Test results for LYS, ARG, and VAL.

Molecule	Start structure	First minimum (Hartree)	Best minimum (Hartree)	Best minimum
				without constants (Hartree)
LYS	1	0.0170	0.0120	0.0121
	2	0.0170	0.0120	0.0123
	3	0.0148	0.0120	0.0120
	4	0.0151	0.0120	0.0120
ARG	1	0.0391	0.0168	0.0179
	2	0.0353	0.0170	0.0192
	3	0.0336	0.0168	0.0189
	4	0.0355	0.0175	0.0203
VAL	1	0.0153	0.0130	0.0131

Figure 3-14 - Figure 3-16 superimpose the first, best, and worst found energy minimum structures for the LYS, the ARG, and the VAL.

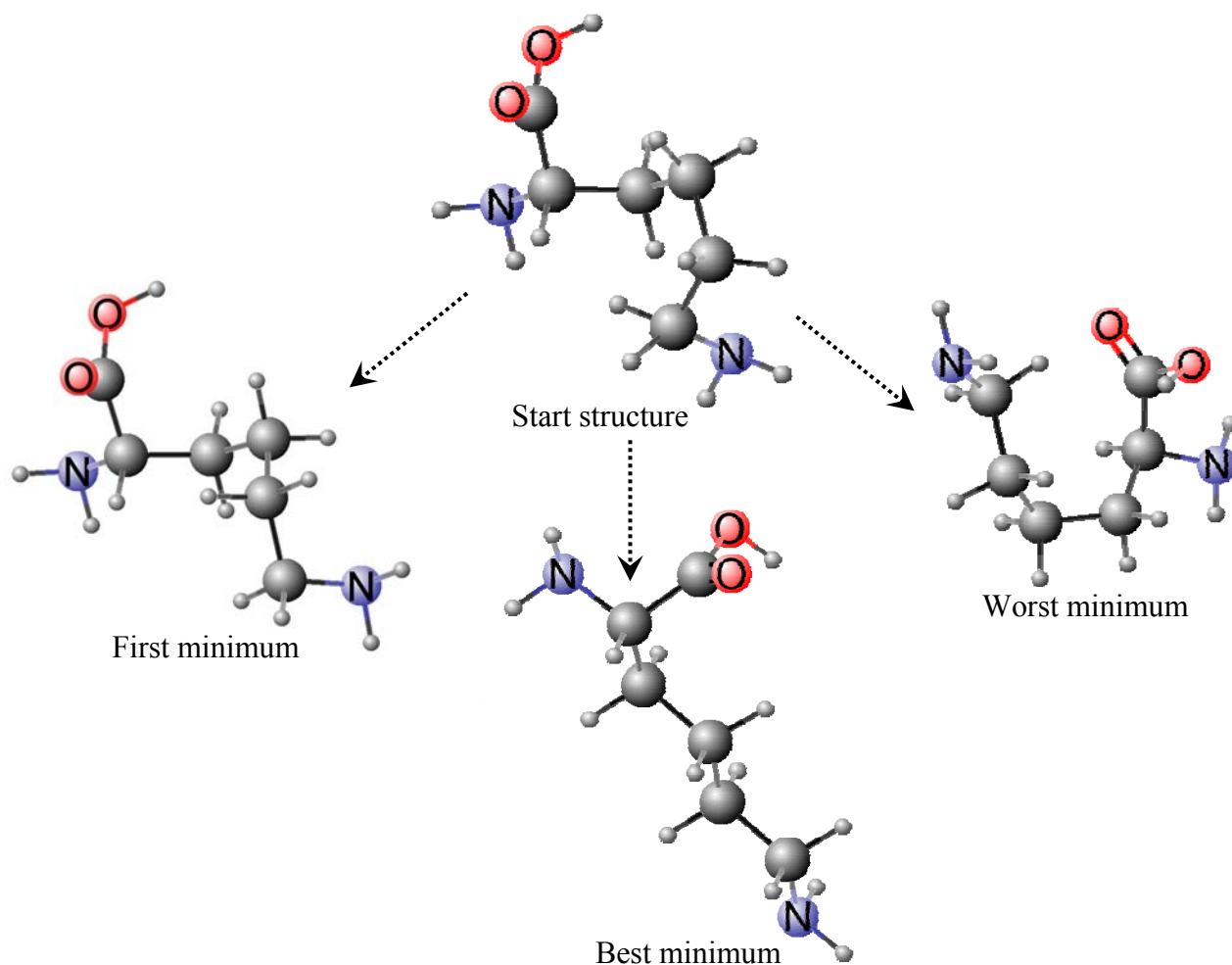


Figure 3-14. First, best, and worst minimum structures of the LYS.

To demonstrate the behaviour of the search optimization process Figure 3-17 monitors the local minima found along the search starting from the first test structure. The minimum ordinal number is given on the abscissa while the corresponding energy value is given on the ordinate.

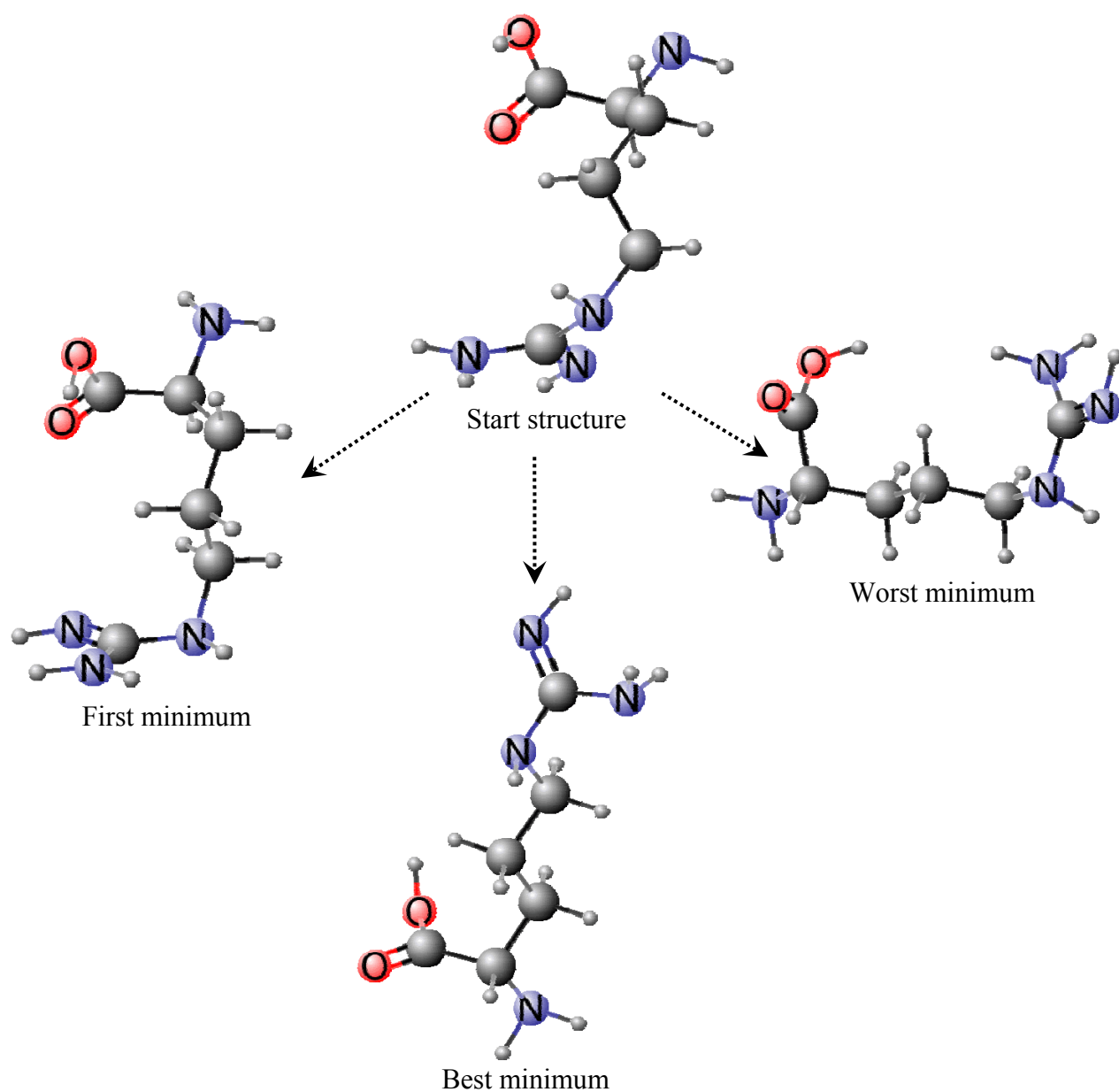


Figure 3-15. First, best, and worst minimum structures of the ARG.

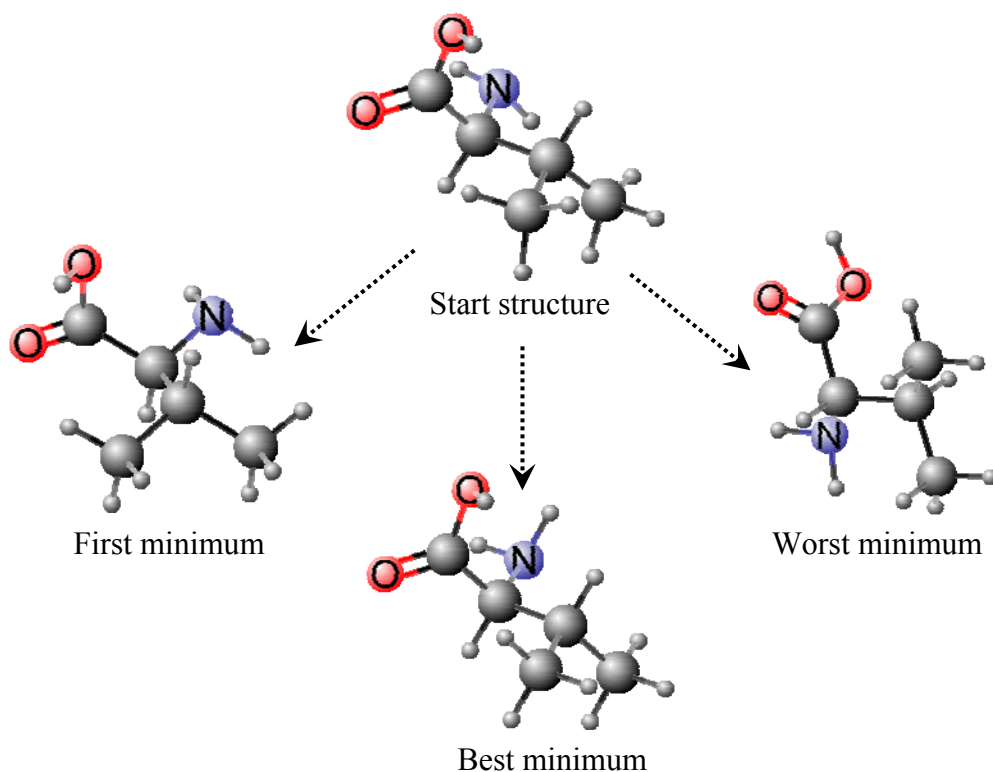


Figure 3-16. First, best, and worst minimum structures of the VAL.

Obviously the course of the search depends strongly on the molecule under consideration. For example, in the case of the VAL molecule, there are only two “*main torsions*” in the second part of the optimization process. Hence, one frequently returns to conformations with the similar energy values. However, it is important to note that equal energy values can characterize different molecule structures. In such cases a comparison of all optimized structures is necessary to decide if a new structure is found. The course found for Valine nicely shows how our procedure can move to a complete different region (steps 27-35) but returns to the most favourable one. The most simple case is the LYS molecule. However, the curve for the ARG molecule structure shows that to reach the best minimum the DS and SA were successfully applied to improve the search outcome.

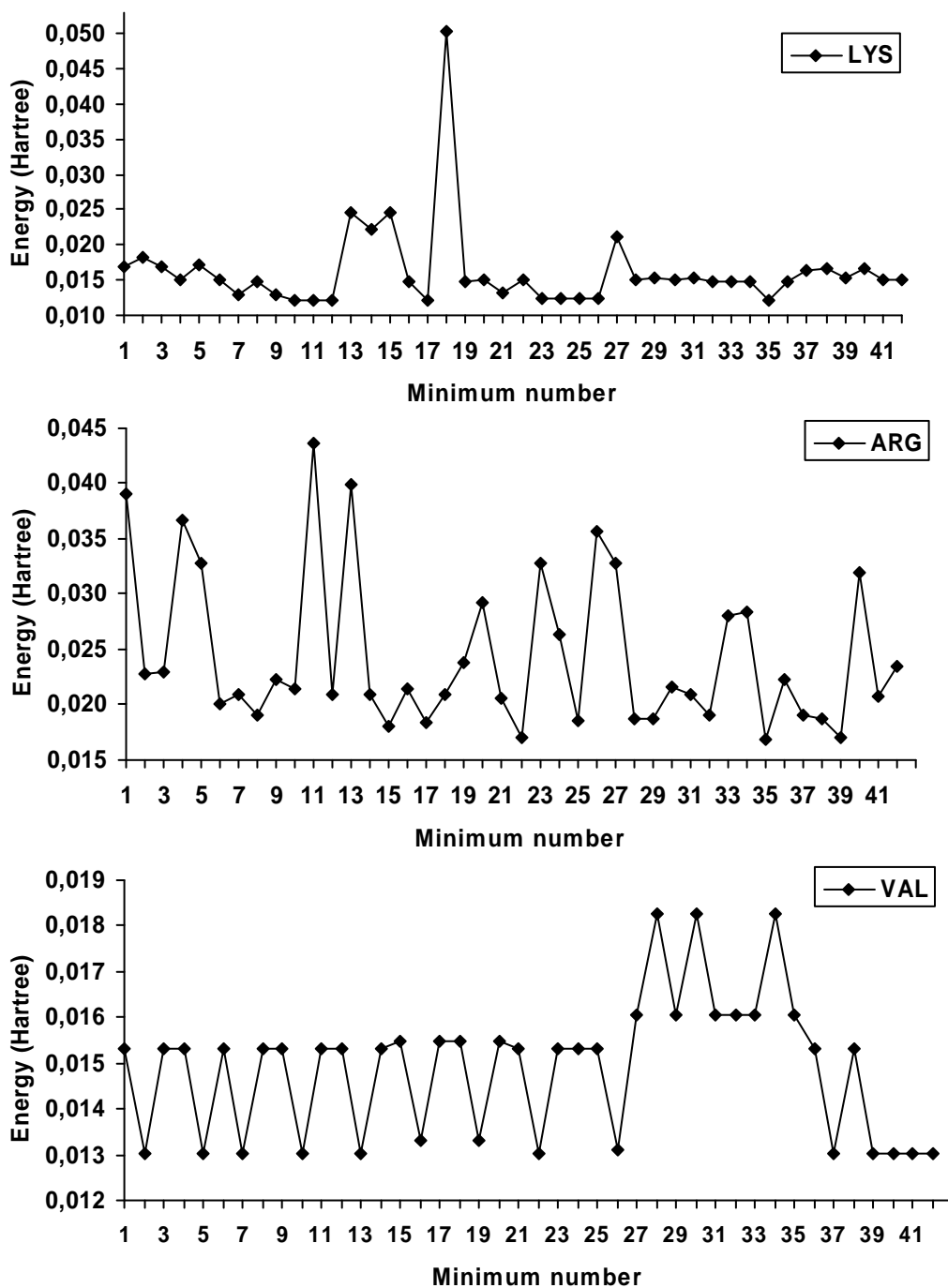


Figure 3-17. Illustration of the optimization process for LYS, ARG, and VAL molecules.

3.3.2 Conformational studies of acetylcholine

The Acetylcholine molecule was treated analogous to the amino acids. The molecule was optimized with and without a division of the molecule. The small part was optimized first and again kept constant during the following modest ascent searches. The partitioning used for the latter and the dihedral angles of both parts of Acetylcholine molecule are presented on the Figure 3-18.

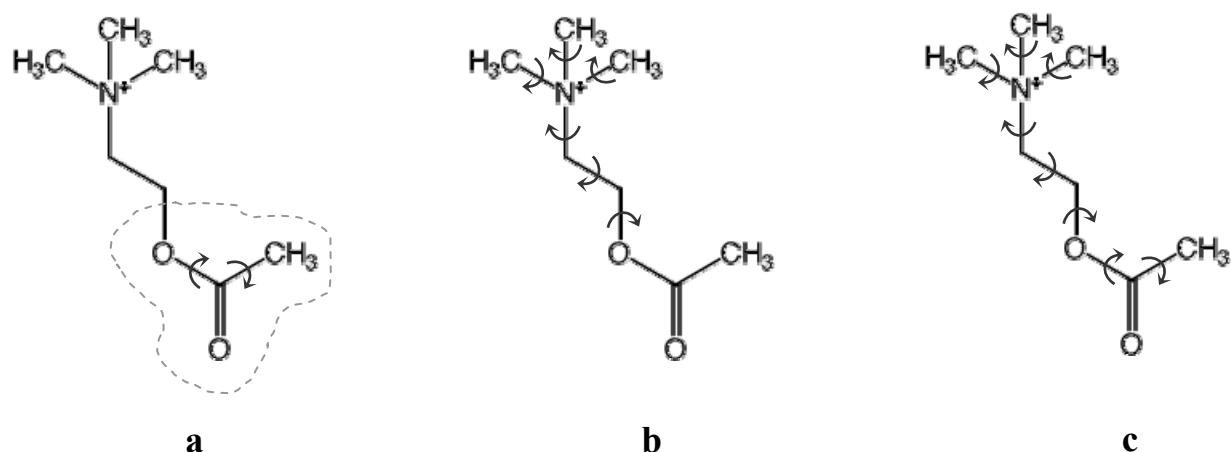


Table 20. Test results for Acetylcholine.

Three different start structures were created to test the abilities of the method. In all cases, our method converged to the same minimum. It is depicted in the Figure 3-19 together with the first and worst found minima. The energies are collected at the Table 20.

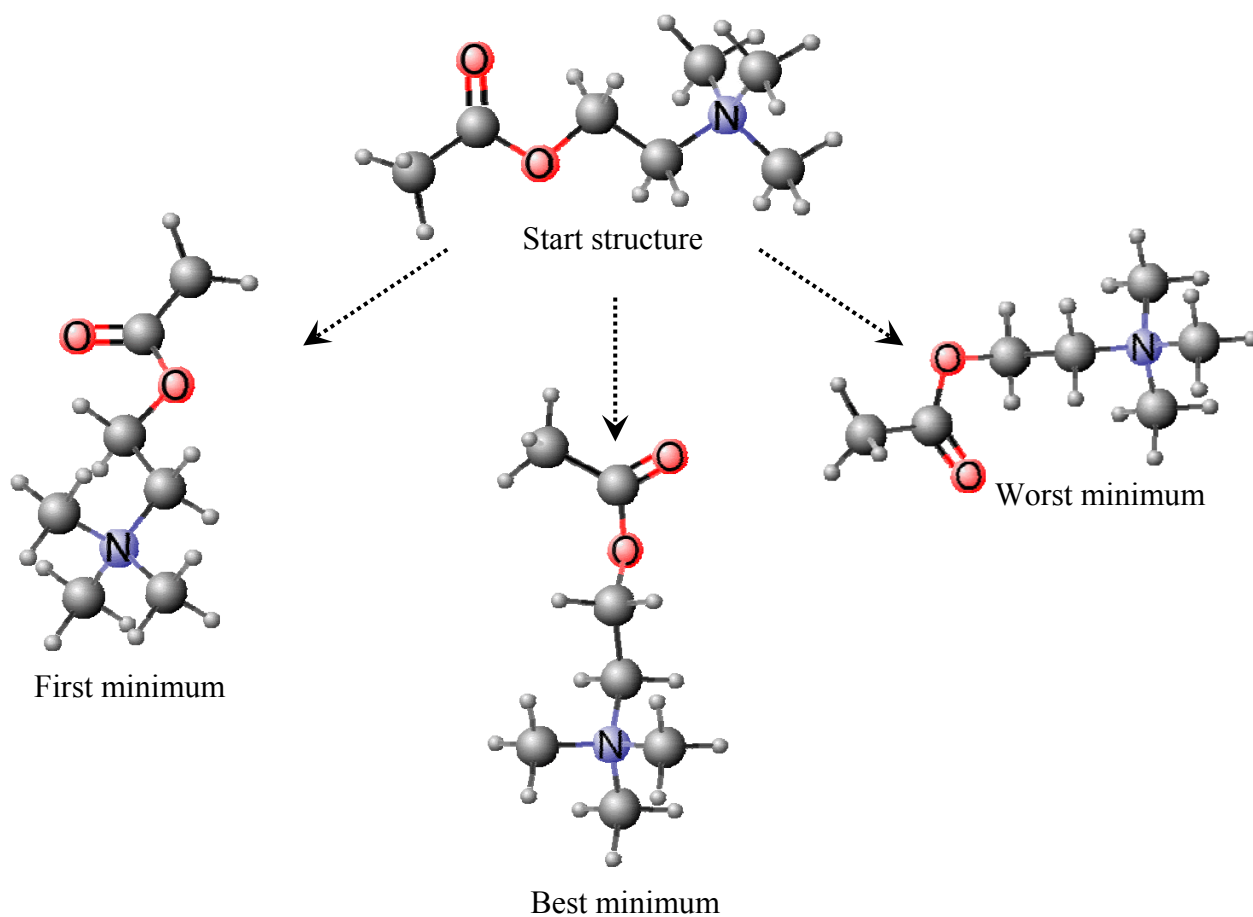


Figure 3-19. Start structure, first, best, and worst minimum structures of Acetylcholine.

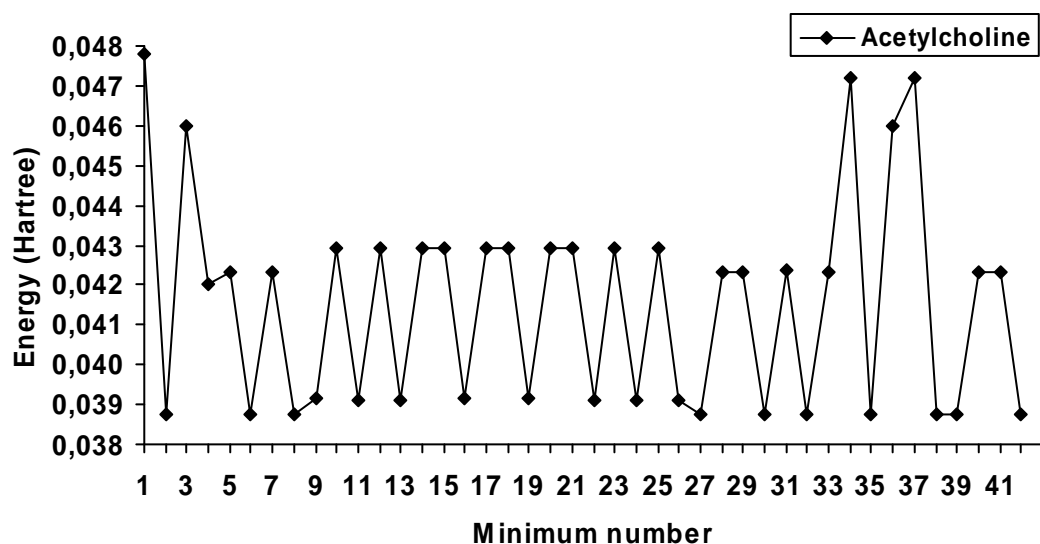


Figure 3-20. Illustration of the optimization process for Acetylcholine molecule.

3.3.3 Conformational studies of ACE and HIV-1 protease inhibitors.

To test our approach for the optimizations of larger molecules conformational searches were carried out for two angiotensin converting enzyme (ACE) inhibitors and HIV-1

protease inhibitor. ACE inhibitors are used primarily in treatment of congestive heart failure and hypertension whereas protease inhibitors are applied to treat or prevent infection by viruses, including HIV and Hepatitis C.

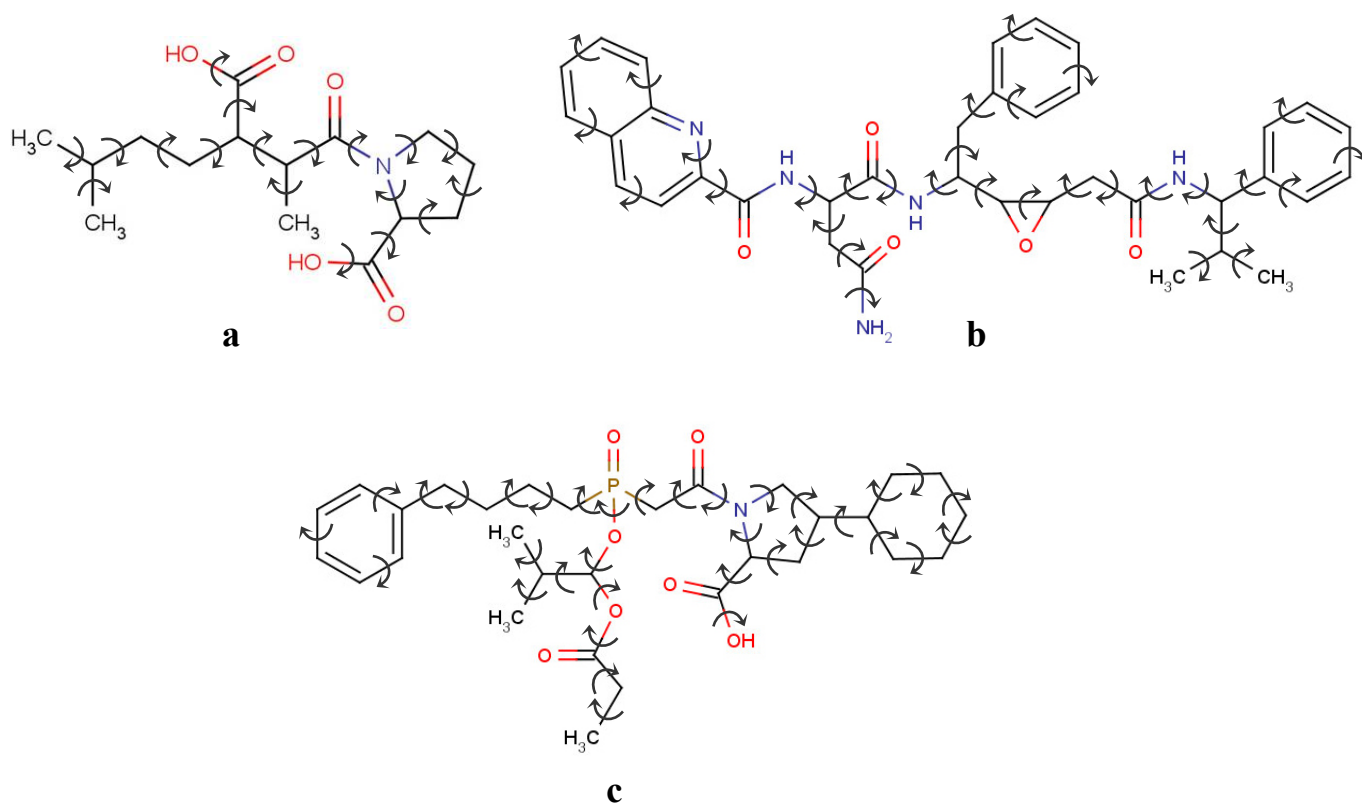


Figure 3-21. Variable torsions. a)ACE inhibitor b) Fosinopril c) HIV-1 protease inhibitor.

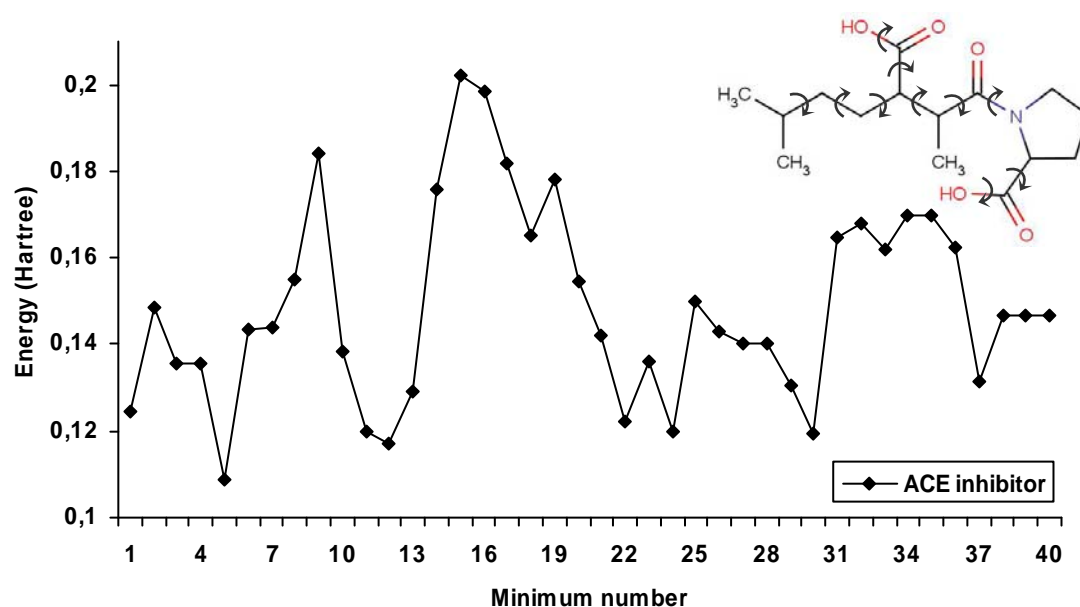


Figure 3-22. Illustration of the optimization process for ACE inhibitor molecule.

For these molecules the parting approach was not employed. The Figure 3-21 illustrates the dihedral angles that are varied during the optimization process. The behaviour of the optimization process for the ACE inhibitor is given in Figure 3-22. During this process only the indicated dihedral angles were varied. The ordinal number of the minima is on the abscissa and the corresponding energy value is on the ordinate. The results, collected in Table 21, show that the new approach can achieve low lying conformers. Figure 3-23 - Figure 3-25 show the first, best and worst found energy minimum structures for ACE and HIV-1 protease inhibitors.

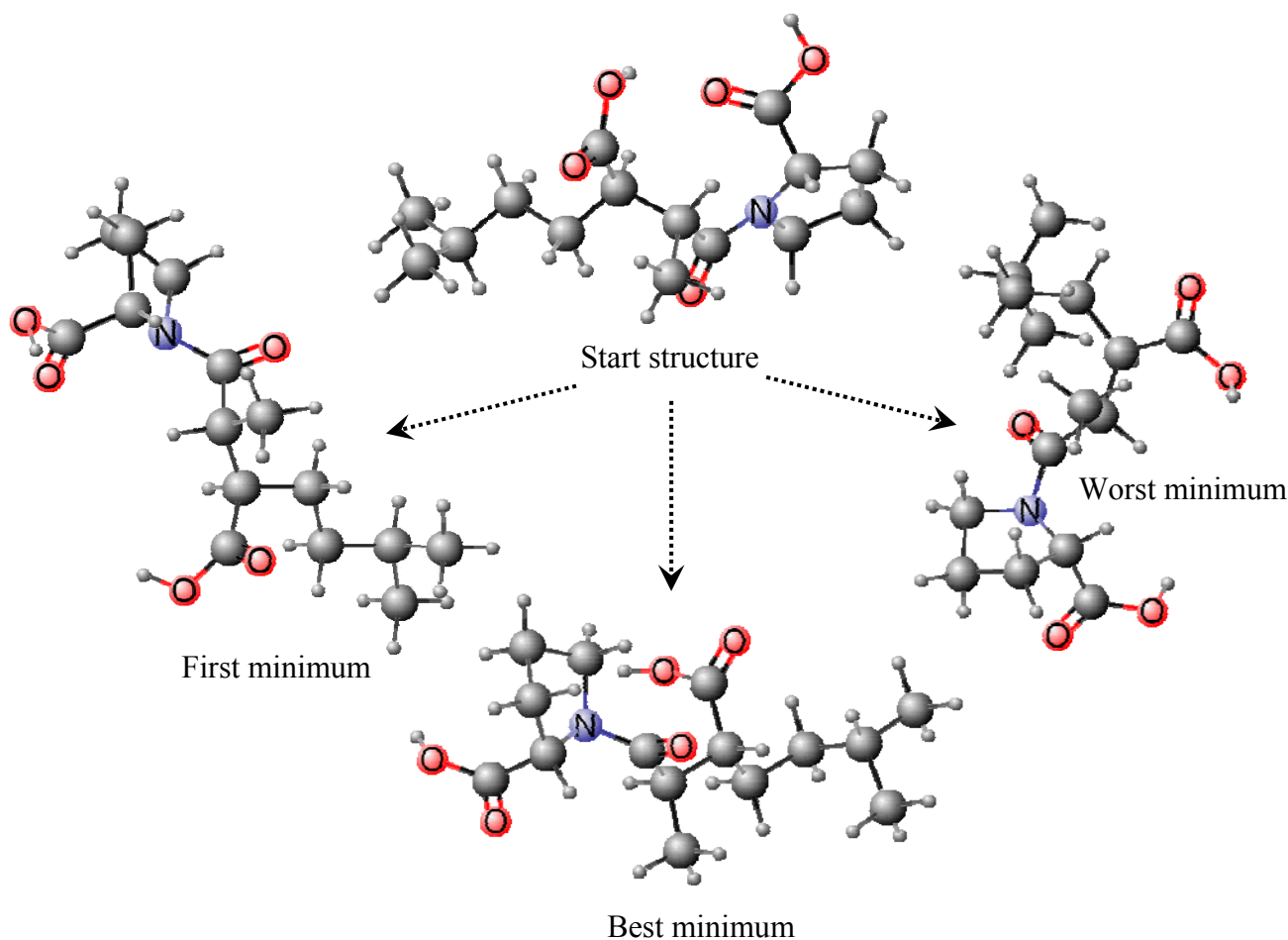


Figure 3-23. Start structure, first, best, and worst minimum structures of ACE inhibitor.

Table 21. Test results for ACE and HIV-1 protease inhibitors.

Molecule	First minimum (kcal/mol)	Best minimum without constants (kcal/mol)
Peptidomimetic HIV-1 protease inhibitor	0.757	0.729
Inhibitor of ACE	0.124	0.113
Fosinopril	0.201	0.183

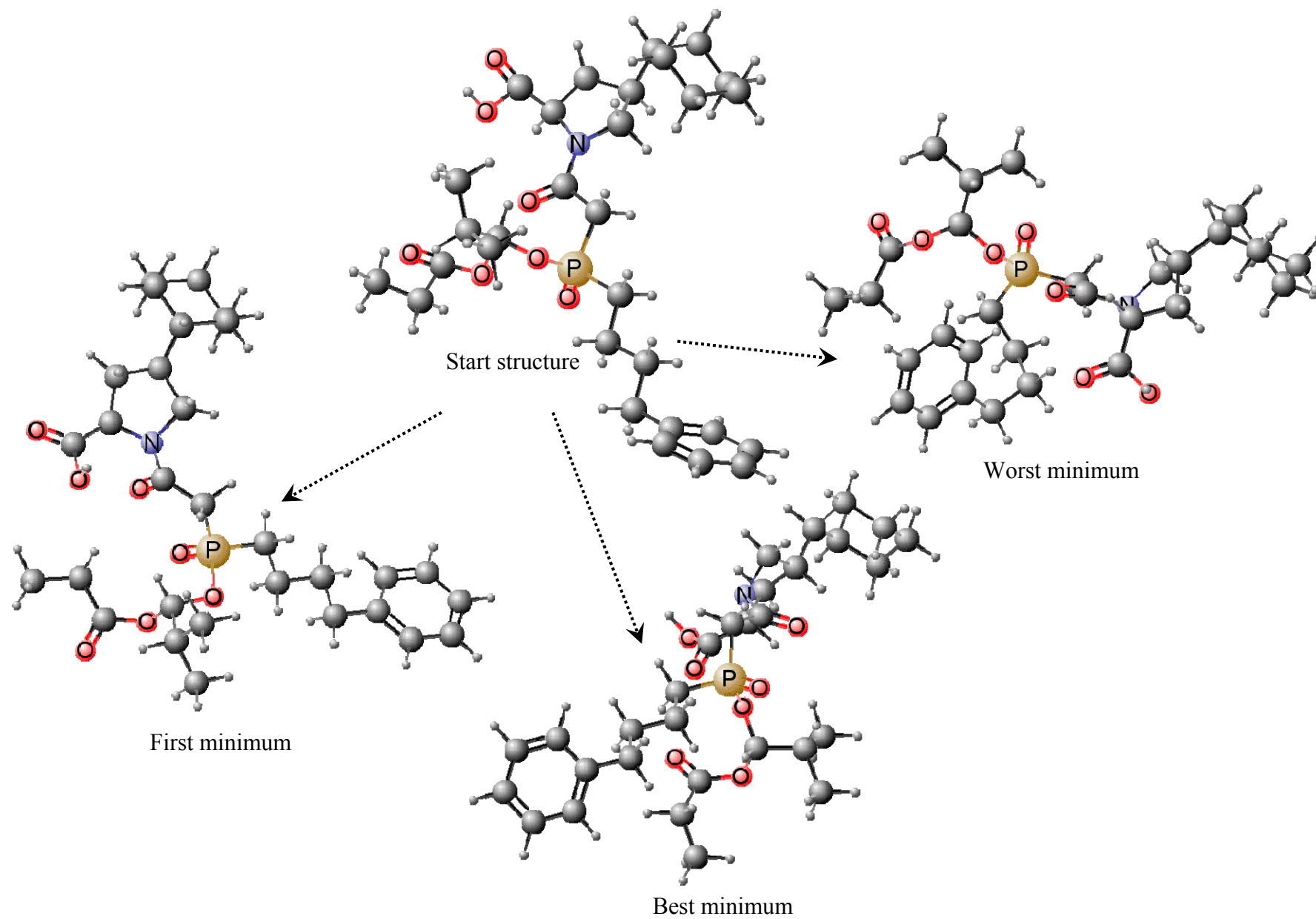


Figure 3-24. Start structure, first, best, and worst minimum structures of Fosinopril.

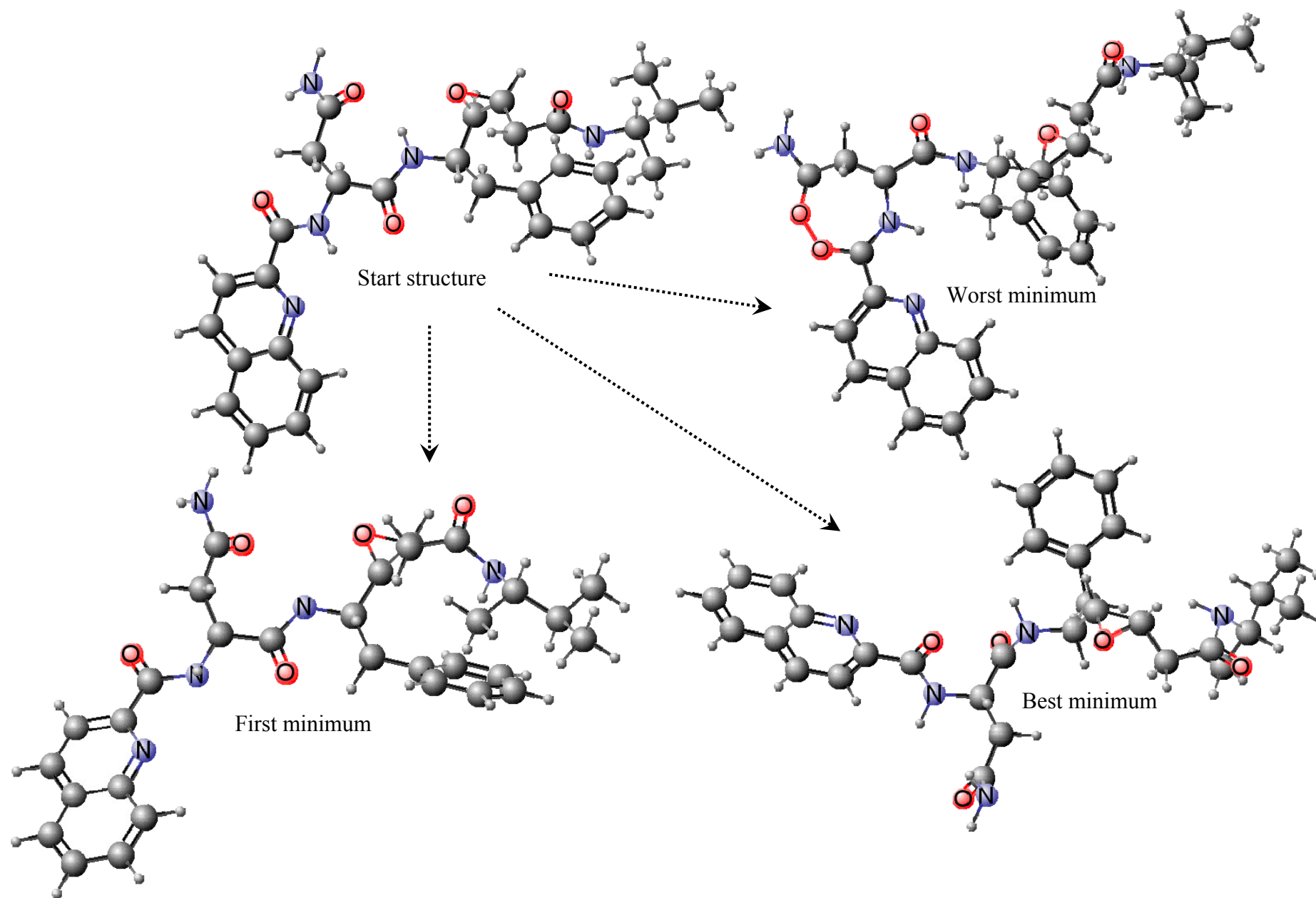


Figure 3-25. Start structure, first, best, and worst minimum structures of HIV-1 protease inhibitor.

3.4 Conclusions

In this chapter the global optimization routine Gradient only Tabu-Search (GOTS) was adapted and applied to conformational search problems. It provides the same global minimum in the tested molecules independently which starting structure was taken. Finally, the method is quite general and can work with more sophisticated geometry optimization and single-point calculation methods of ChemShell. The Z-matrix form is used for the definition of a molecular structure and to manipulate the chemical structures. Implementation of Simulated Annealing ideas gives a possibility to delocalize the search, to investigate more promising areas and to locate a good approximation to the global optimum of a given function within a large search space.

The GOTS algorithm was tested for several molecular structures. It can be also applied to cyclic molecules under condition that some of torsions are kept constant to save a ring structures. The test calculations indicate that the GOTS method is an efficient approach for conformational searches.

Chapter 4 Summary

The optimization is a part of mathematics. The optimization aim is to find a discrete mathematical object that maximizes or minimizes an objective arbitrary function specified by the user. The nature of such mathematical objects and the search space are usually problem-specific. The optimized system is typically only optimal in one application. Many activities can be formulated as optimization problems. Because of the computational complexity of these problems, exact optimization techniques of operations research like linear programming or dynamic programming are for the most part unfeasible for large-scale problems. Therefore, researchers, engineers, and scientists have used metaheuristic search techniques to find near optimal, good-enough optimal or even the global solutions. Of course, the ultimate aim is to apply metaheuristic search techniques like GAs, SA, and TS to solve real-world problems.

This work encompasses three parts. The first part provides a concise review of the most prominent metaheuristic concepts currently available and gives essential preliminaries together with definition of the combinatorial optimization problems. It substantiates the choice of the investigation direction and basis idea of the developed methods.

In the second part the new nonlinear global optimization routines based on the TS strategy are described. The new approaches are the Gradient Tabu Search (GTS), the Gradient Only Tabu Search (GOTS), and the Tabu Search with Powell's Algorithm (TSPA). All try to determine the global minimum of a function by the *steepest descent - mildest ascent* strategy. The steepest descent is followed to find the next local minimum. To escape a local minimum the algorithm moves along the modest ascent till a next valley is reached. Then again the steepest ascent strategy is used to determine the minimum of this valley and so on. GTS and GOTS use the combination of the Steepest Descent and the Quasi-Newton methods for the steepest descent part. To realize the modest ascent strategy the GTS evaluate the diagonal elements of the Hessian while the GOTS scans the neighbourhood by function evaluations. Also the TSPA uses the latter method for the modest ascent strategy. For the steepest descent part, however, the Direction Set Method with Powell's Algorithm is employed. To ensure an efficient blocking of already visited regions despite the lower number of steps these approaches introduce Tabu Regions and Tabu Directions as new elements for the Tabu List. The new algorithms are compared with other approaches using various well-known test

functions with varying dimensionality. These test computations with up to 50 variables show that the new approaches outperform most previous approaches on the basis of the number of function evaluations. The surfaces of the test function are shown on the Figure 4-1. Comparison with the differential evolution ansatz proves the efficiency of the present approaches on the basis of wall clock timings especially for a high number of variables. The influence of user-defined parameters on the efficiency of the new approaches is also investigated. Moreover, the proposed approaches could be enhanced even further if more sophisticated descent method and diversification strategies are applied instead.

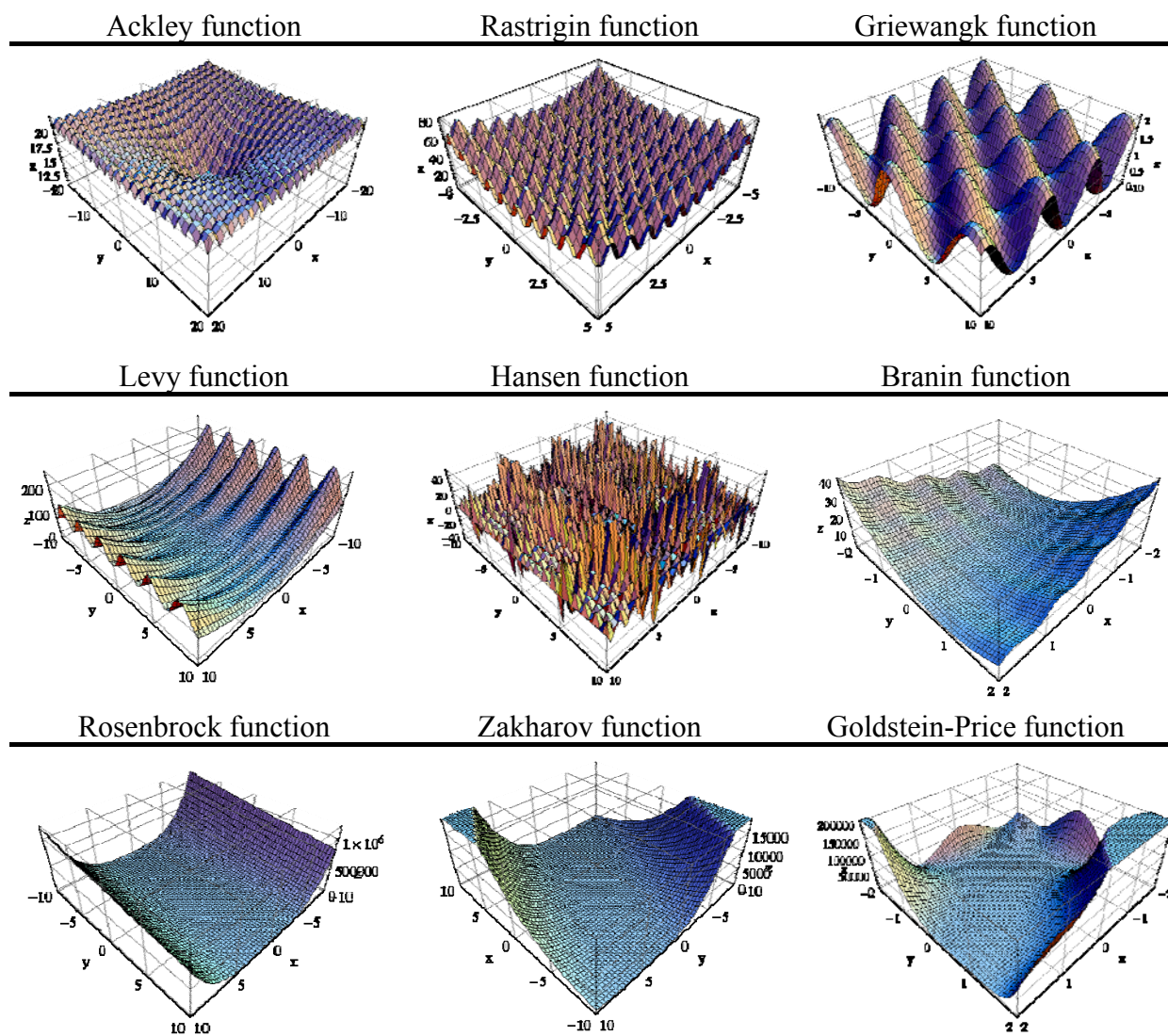


Figure 4-1. Test function.

In order to test the appropriateness of the GOTS method for molecular geometry problems, the optimization of a function with a functional form similar to general potential energy functions (see Eq. 20) was carried out.

$$E = \sum_{i=1}^n (1 + \cos 3\omega_i) + \frac{(-1)^i}{\sqrt{10.60099896 - 4.141720682(\cos \omega_i)}}, \quad (20)$$

where ω_i is the torsion angle $\omega_{i(i+3)}$,

$n+3$ is the number of atoms or beads in the given system

Illustration of the surface of this function is presented on the Figure 4-2. It is significant that the global minimum of the function is known. The number of its local minima increases exponentially with the size of the problem, which characterizes the principal complexity of the minimization of the molecular potential energy functions.

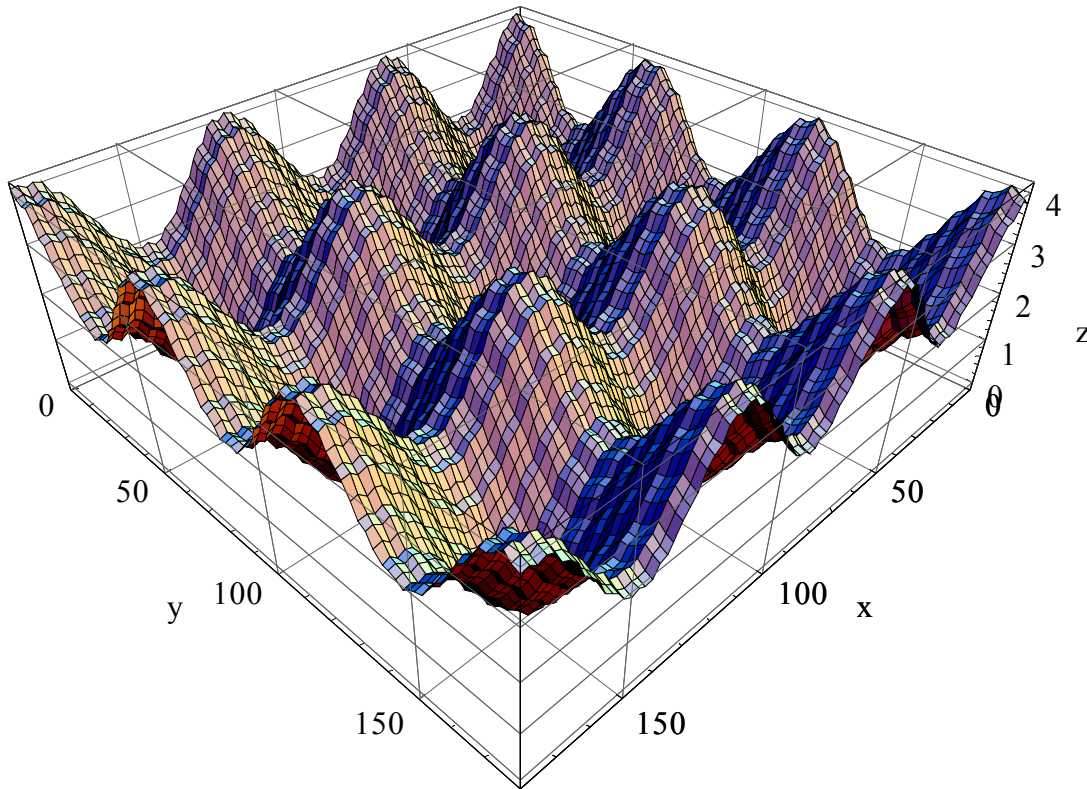


Figure 4-2. Surface of the function with functional form similar to general PES function.

The GOTS was successfully applied to this the n -chain problem ($n = 7, \dots, 20$), i.e. the global minimum was always found. It was much efficient than the Deterministic algorithm that uses Branch and Bound scheme with techniques of interval analysis to provide the lower bounds. Figure 4-3 demonstrates the efficiency of the GOTS method in comparison to the Deterministic algorithm on the basis of number of function evaluation that introduced in the diagram form for function of 7, 10, and 20 variables. The amount of function evaluations required determining the global minima increases by a factor of about 11 if the dimension increases from 10 to 20 whereas this factor for Deterministic algorithm is 352.

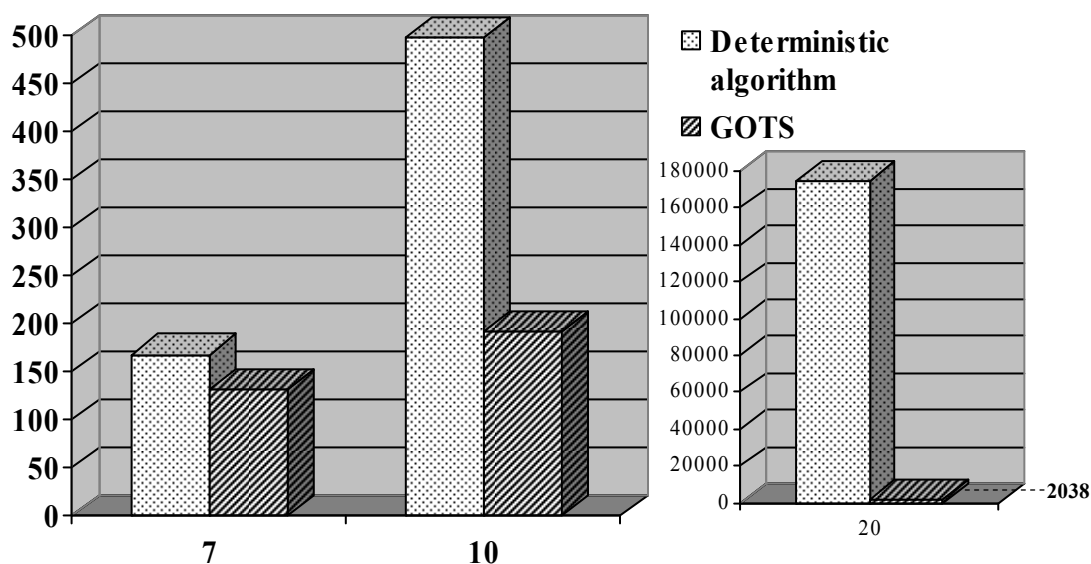


Figure 4-3. Diagrams for function of 7, 10, and 20 variables.

GOTS provides excellent solutions for hard problems with minimal computational effort. The algorithm can be suitably modified to incorporate more rigorous mechanisms to handle very large-scale problems. Because of problem-specific knowledge GOTS deals with variables and constraints effectively. However, the trade-off between effectiveness and generality requires careful consideration, and the junction with other techniques is necessary in order to build even more general methodology. In this connection the recommended default parameters were given at this work to be destined with several test cases. The large-scale test cases show that the initial parameter settings for the GOTS may be overly cautious; however, in most cases, the global minimum is found independently which parameter values are used. Nevertheless, these calculations also show that parameters settings are problem dependent. The optimal parameter values may require some adjustment to starting values indicated in this work. Strategies to estimate optimal parameter values are discussed.

An efficient search for the global minimum of a highly dimensional function with many local minima is central for the solution of many problems in computational chemistry. Well known examples for such global optimization problems are the conformational search for molecules with a high number of freely rotatable bonds or the optimization of parameters of a force field. Conformational analysis is the study of a molecule conformations and the influence of them on the molecule properties. The basis of a conformational analysis is the conformational search which has the goal to identify the energetically lowest lying conformers of the molecule. This usually requires the location of the global energy minimum of the PES. These structures are very important because they determine most of the properties of the molecule. Over the past several years, a multitude of conformational search techniques have been developed for this purpose.

In the last part of the work the GOTS is applied for such chemical optimization problems. The chapter provides a systematic approach how the variables are chosen and the adjustable parameters are set. As test cases the global minimum energy conformation of some amino acids, of two angiotensin converting enzyme (ACE) inhibitors, of 2-acetoxy-N,N,N-trimethylethanaminium, and of a HIV-1 protease inhibitor is determined.

The major advantages of the exploitation of GOTS method are that it provides the achievement of the global minimum without depending on the starting structure. The concept of *main* and *dependent torsions* was used to allow easy implementation of the internal rotations. It decreases the number of the variables of the optimization process and gives the possibility to speed up the optimization. The mildest ascent strategy does not require the computation of gradients, thereby leading to an additional saving in computational time. Finally, the method is quite general and can work with more sophisticated geometry optimization and single-point calculation methods of ChemShell. Some alterations were made at the original version of the method. The diversification strategy was modified. In the original GOTS the diversification strategy becomes necessary if the solution does not improve after a number of iterations that means that three found local minima at hand are worse than the current solution or if all neighbourhood solutions are already set tabu. Implementation of Simulated Annealing ideas concerning found local minima gives a possibility to delocalize the search, to investigate more promising areas, and to locate a good approximation to the global optimum of a given function in a large search space. If the new minimum is not accepted according to the so-called Metropolis criterion the algorithm returns to the previous minimum and continues the search along the next modest ascent direction. This increases the efficiency of the GOTS method.

The GOTS algorithm can be applied to cyclic molecules under condition that some of torsions keep constant to save a wholeness of a structure. So, the test calculations indicate that the GOTS is an efficient approach for conformational searches but also provides widespread fields of action.

Chapter 5 Zusammenfassung

Die Optimierung ist ein Teilgebiet der Mathematik. Das Optimierungsziel ist das globale Maximum oder Minimum einer vom Benutzer bestimmte Zielfunktion. Der Suchraum und die Natur solcher Objekte sind problemspezifisch und die mathematische Behandlung meist recht kompliziert. In der Regel sind zur Lösung entwickelte mathematische Modelle zudem nur begrenzt verwendbar, d.h. schon kleine Veränderungen in den realen Fragestellungen erfordern recht unterschiedliche Modelle. Viele Fragestellungen können als Optimierungsprobleme formuliert werden. Wegen des hohen Rechenaufwandes sind die genauen Optimierungsmethoden der Operationsforschung, wie z. B. lineare oder dynamische Programmierung, zum größten Teil jedoch undurchführbar. Deshalb wurden Metaheuristik-Suchmethoden entwickelt, um möglichst gute Näherungslösungen zu finden. Die Metaheuristik-Suchmethoden wie genetische Algorithmen, das Simulated Annealing oder die Tabu-Suche haben sich zu wichtigen Hilfsmitteln in den Naturwissenschaften und der Wirtschaftsmathematik entwickelt.

Die Arbeit umfasst drei Kapitel. Das erste Kapitel stellt eine kurze Zusammenfassung über die bekanntesten, zurzeit verwendeten Metaheuristischen-Konzepte dar und gibt notwendige Einleitungen zusammen mit der Definition der kombinatorischen Optimierungsprobleme. Das Kapitel begründet die Wahl des Tabu-Ansatzes und diskutiert die Basisideen der entwickelten Methoden.

Im zweiten Kapitel werden die neuen entwickelten, nichtlinearen Optimierungsroutinen beschrieben, die auf Tabu-Suchstrategien beruhen. Die neuen Algorithmen sind Gradient Tabu Search (GTS), Gradient Only Tabu Search (GOTS) und Tabu Search with Powell's Algorithm (TSPA). Bei diesen Methoden wird das globale Minimum einer Funktion durch die „*steilsten Abstieg - schwächste Aufstieg*“-Strategie bestimmt. Dem *steilste Abstieg* folgt man, um den nächsten lokalen Minimum zu finden. Um von einem lokalen Minimum zum nächsten zu kommen bewegt sich der Algorithmus entlang der *geringsten Aufstiegs* bis das nächste Tal erreicht ist. Dann wird wieder die *steilste Abstieg* Strategie verwendet, um das Minimum dieses Tales zu finden und so weiter. GTS und GOTS verwenden der Kombination die Steepest-Descent- und die Quasi-Newton-Methoden für den *steilste-Abstieg-Bereich*. Der GTS wertet die diagonalen Elemente des Hessians zur

Realisierung der *schwächste-Aufstieg-Strategie* aus während die GOTS Methode die Nachbarschaft mit Funktionswertberechnungen abscannt. Auch der TSPA Ansatz verwendet die letztere Methode für die *schwächste Aufstieg* Strategie. Für den *steilsten Abstieg* wird aber die *Direction Set Method with Powell's Algorithm* verwendet. Um eine effiziente Sperrung der bereits besuchten Regionen trotz der geringeren Anzahl von Schritte zu erreichen, führen alle Konzepte Tabu-Regionen und Tabu-Richtungen als Elemente der Tabu-Liste ein. Sie blockieren bereits besuchte Regionen sehr effizient und sind flexibel und leicht zu verwenden. Anschließend folgt eine Diskussion über Effizienz der Algorithmen. Hierbei werden die Ergebnisse mit anderen Ansätzen verglichen. Der Vergleich wird an Hand gut bekannten Testfunktionen mit unterschiedlichen Dimensionalität durchgeführt. Untersucht werden hierzu Funktionen, die von 2, 10, 30 oder 50 Variablen abhängen. Berechnungen mit häufig verwendeten Test-Funktionen (Abbildung 5-1) zeigen, dass die neuen Methoden effizienter sind als bisherige Ansätze. Insbesondere bei hochdimensionalen Problemen ergeben sich erhebliche Vorteile. Ein Vergleich mit dem Differential-Evolution-Ansatz beweist ebenfalls die Effizienz der neuen Methoden auf der Basis von „wall clock timings“. Ferner beinhaltet das Kapitel eine Untersuchung des Einflusses benutzerdefinierter Parameter auf die Effizienz der neuen Ansätze. Weiterentwicklung wären durch anspruchsvollere Methode für das „*steilsten Abstieg*“ - Verfahren oder Diversifikationsstrategien möglich.

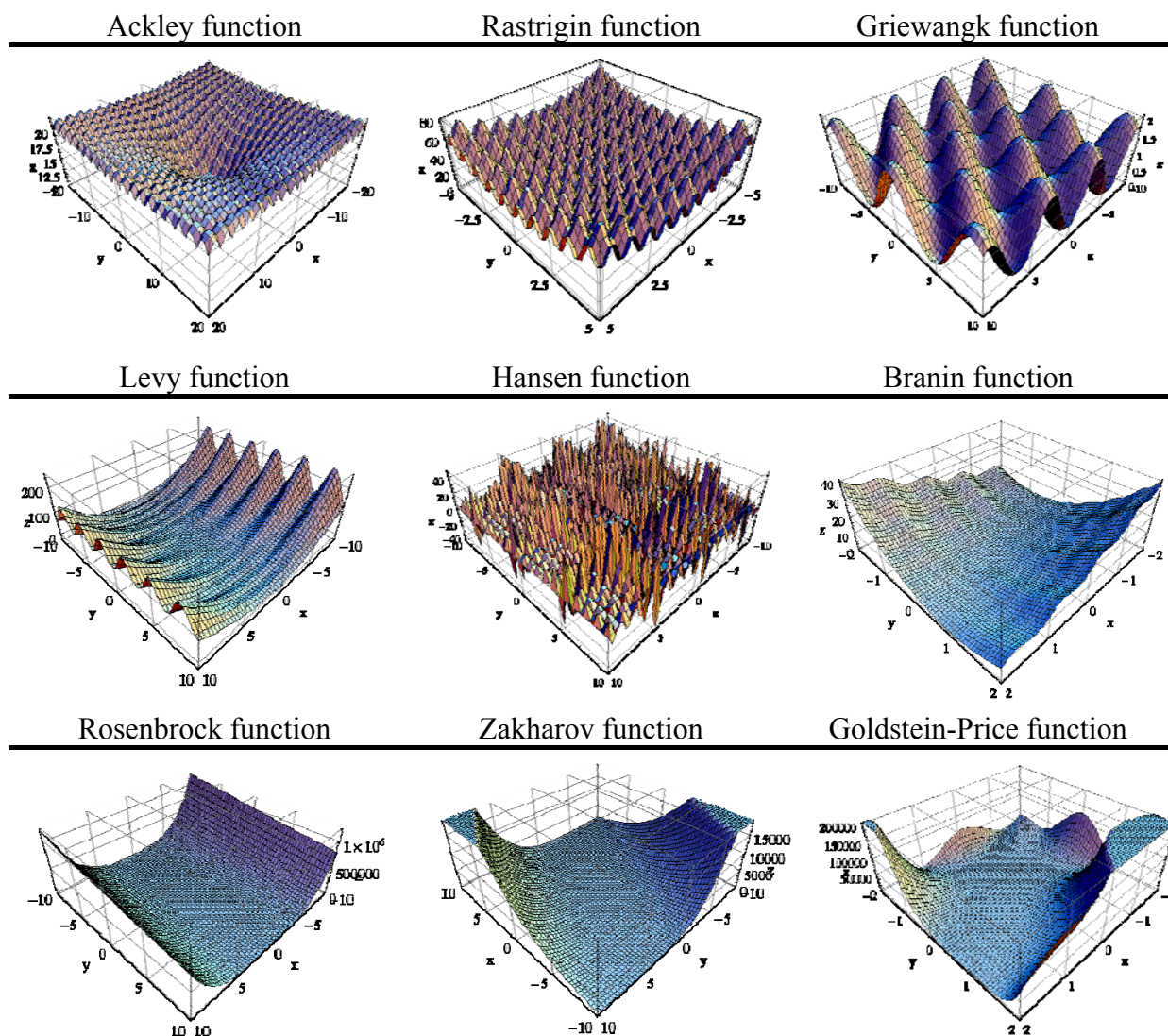


Abbildung 5-1. Test Funktionen.

Um die Eignung der GOTS-Methode für Problemen der Moleküloptimierung zu prüfen, wurde auch die Optimierung einer Funktion durchgeführt, die den in der Chemie auftretenden Potentialhyperflächen ähnelt (siehe Gl. 20).

$$E = \sum_{i=1}^n (1 + \cos 3\omega_i) + \frac{(-1)^i}{\sqrt{10.60099896 - 4.141720682(\cos \omega_i)}}, \quad (20)$$

wobei ω_i der Verdrehungswinkel $\omega_{i(i+3)}$,

$n+3$ die Zahl von Atomen oder Teilchen im gegebenen System ist.

Abbildung 5-2 illustriert den Funktionsgraphen des Funktion von 2 Variablen. Die Zahl ihrer lokalen Minima nimmt exponential mit der Größe des Problems zu, was der Komplexität der Funktionen der molekularen potentiellen Energie entspricht.

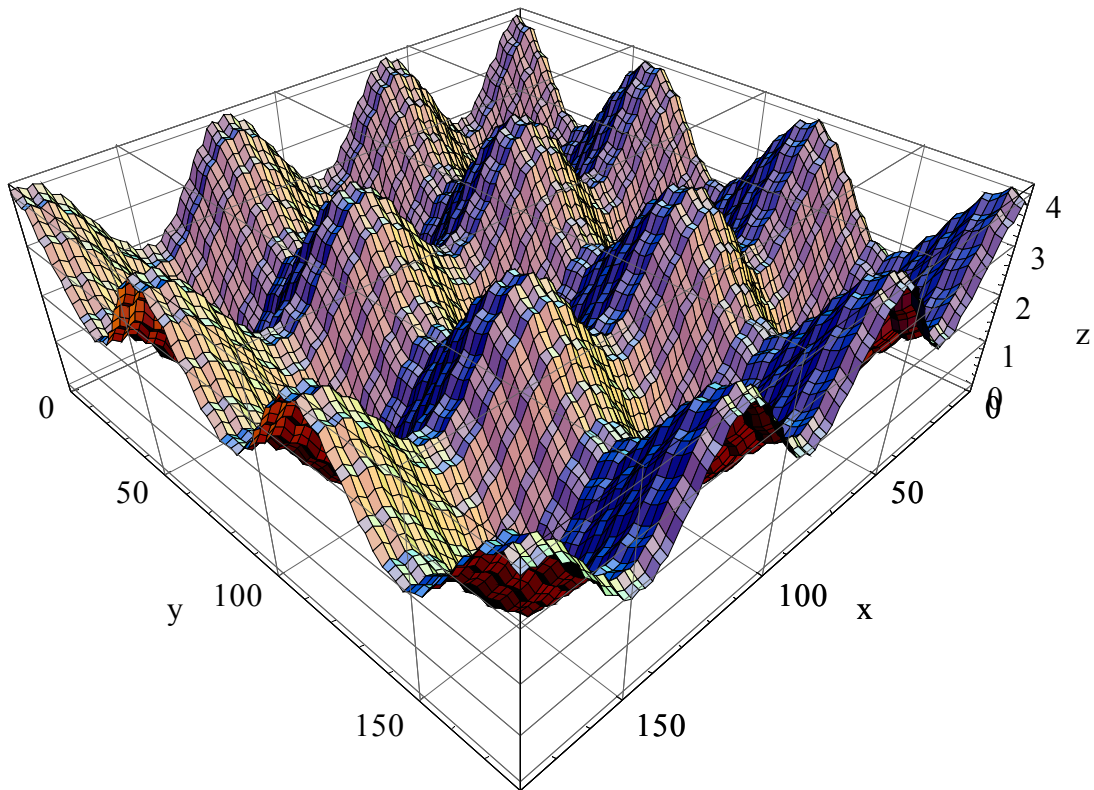


Abbildung 5-2. Oberfläche der Funktion mit funktionalen Form ähnlich allgemeinen potentielle Energie Oberfläche Funktion.

In den Tests für dieses so genannte n -Kette-Problemen ($n = 7, \dots, 20$) findet die GOTS Methode immer das globale Minimum, allerdings treten auch die Probleme, die bei der Suche des globalen Minimums solcher Funktionen entstehen, klar hervor. Ein Vergleich mit dem Deterministischen Algorithmus der die „Branch-and-Bound“-Schema mit „Intervall-Analyse“-Methoden verwendet, um die niedrigsten Grenzen zu erreichen zeigt, dass die GOTS Methode auch für diese Anwendung sehr effizient ist. Abbildung 5-3 vergleicht den für beide Methoden notwendigen Aufwand auf Basis der zur Bestimmung des globalen Minimums erforderliche Funktionswertberechnungen. Untersucht wurden Funktionen, die von 7, 10 und 20 Variablen abhängen. In allen Fällen wurde das benannte globale Minimum gefunden. Dabei steigt die Zahl der notwendigen Funktionswertberechnungen bei der GOTS Methode um einen Faktor von etwa 11, wenn die Dimension von 10 auf 20 vergrößert wird. Für den Deterministischen Algorithmus beträgt der Faktor 352.

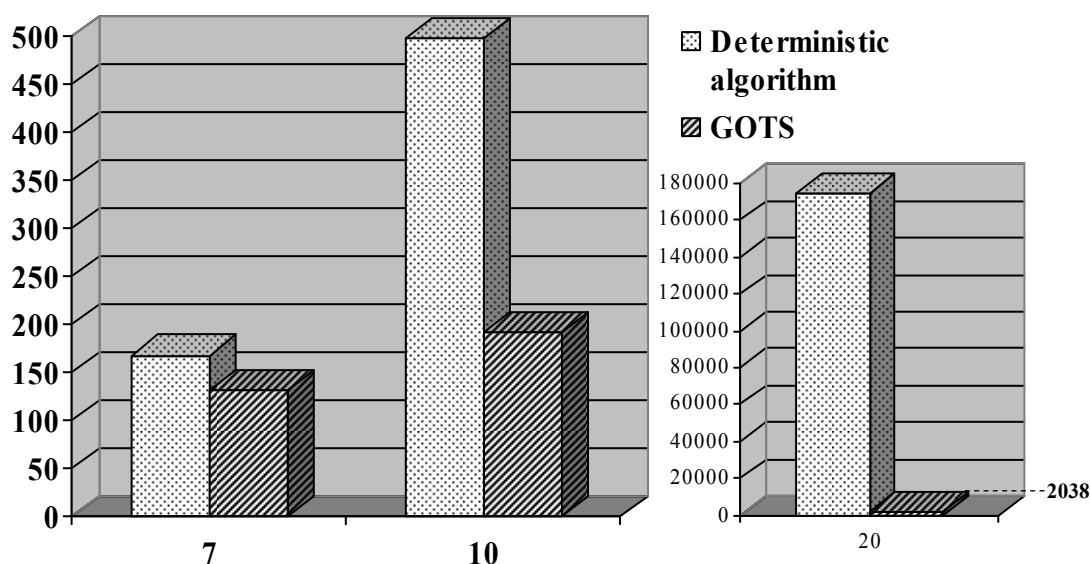


Abbildung 5-3. Anzahl der Funktionswertberechnungen, die zur Bestimmung des globalen Minimums der in Gl. 20 definierten Funktion notwendig waren.

Alle unsere Untersuchungen zeigen, dass der GOTS die Minima komplexer Probleme sehr effizient findet. Allerdings wird die Effizienz von dem Parameter des Algorithmus beeinflusst. Aufbauend auf unseren Tests werden daher Werte für diese Parameter vorgeschlagen. Unabhängig von der Wahl der Parameter werden die globalen Optima aber in den weitaus meisten Fällen gefunden. Die Tests zeigen, dass die optimalen Größen der Parameter problemabhängig sind. Daher werden ebenfalls Wege diskutiert, wie man optimale Werte abschätzen kann.

Eine effiziente Suche nach dem globalen Minimum einer hochdimensionalen Funktion mit mehreren lokalen Minima ist auch von zentraler Bedeutung für viele Probleme der Computerchemie. Bekannte Beispiele für solche globalen Optimierungsprobleme sind die Konformationsanalyse bei Molekülen mit einer hohen Zahl frei drehbarer Bindungen oder die Optimierung der Parameter eines Kraftfeldes. Die Grundlage einer Konformationsanalyse ist die Suche nach dem energetisch tiefstliegenden Konformer des Moleküls. Mathematisch ist die Suche gleich bedeutend mit der Suche des globalen Energieminimums der Energiehyperfläche des Moleküls. Diese Konformeren sind sehr wichtig, da sie die Eigenschaften des Moleküls bestimmen. In den letzten Jahren ist eine Vielzahl von konformativen Suchmethoden entwickelt worden.

Das letzte Kapitel der Arbeit beschreibt die Anwendung der GOTS Methode auf dieses Problem. Diskutiert werden die Auswahl der Variablen und die Einstellung der justierbaren Parameter. Die Effizienz der GOTS Methode wird an Hand einiger Aminosäuren,

zwei Angiotensin-Derivaten (ACE-Hemmer), des Acetylcholin und eines HIV-1-Protease-Hemmstoff gezeigt.

Die wichtigsten Vorteile der Verwendung der GOTS Methode bestehen darin, dass das Erreichen des globalen Minimums nicht von der Ausgangsstruktur abhängt. Das Konzept, von Haupt- und abhängigen Torsionen wurde verwendet, um interne Rotation um Einfachbindungen zu ermöglichen. Hierdurch wird ebenfalls die Anzahl Variablen reduziert, was die Optimierung effizienter macht. Die „*mildeste Aufstieg*“-Strategie verlangt keine Berechnung der Gradienten, was ebenfalls Zeit spart. Der neue Ansatz wurde mit dem ChemShell-Programmpaket verknüpft. Zur Anpassung an die neue Aufgabe waren einige Modifizierungen notwendig. Insbesondere die Diversifikationsstrategie wurde angepasst. Im ursprünglichen GOTS wurde nach einem neuem Startpunkt gesucht (Diversifikation), wenn nacheinander drei höher liegende lokale Minima gefunden wurden. Zur Konformationsanalyse wurde eingebaut, dass man mit einer gewissen Wahrscheinlichkeit zum tiefer liegenden Minimum zurückgeht, um dort die Suche nach neuen Minima fortzuführen. Gesucht wird dann in die Richtung, die den zweitniedrigsten Anstieg besitzt. Dies vergrößert die Effizienz der GOTS Methode, da vielversprechende Bereiche intensiver untersucht werden.

Die beschriebene Methode kann auch für zyklische Moleküle verwendet werden. Allerdings müssen einige der Torsionswinkel konstant gehalten werden, damit die Ringstrukturen nicht aufgebrochen werden. Die Testberechnungen deuten darauf hin, dass der GOTS nicht nur ein effizientes Konzept für Konformationsanalyse ist, sondern weit verbreitete Anwendungsbereiche finden kann.

Appendix A Unconstrained Test Problems

A.1 Branin function (BR)

Definition: $BR(x) = (2x_2 + \frac{1}{20} \sin(4\pi x_2) - x_1)^2 + (x_2 - \frac{1}{2} \sin(2\pi x_1))^2$.

Feasible search space: $-10.0 \leq x_i \leq 10.0$, $i = 1, 2$.

Global minimum: $x^* = (0, 0)$; $BR(x^*) = 0$.

A.2 Goldstein and Price function (GP)

Definition: $GP(x) = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2))$
 $(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$.

Feasible search space: $-2.0 \leq x_i \leq 2.0$, $i = 1, 2$.

Global minimum: $x^* = (0, -1)$; $GP(x^*) = 3$.

A.3 Hansen function (H)

Definition: $H(x) = \sum_{i=1}^5 i \cos((i-1)x_1 + i) \sum_{j=1}^5 j \cos((j+1)x_2 + j)$.

Feasible search space: $-10.0 \leq x_i \leq 10.0$, $i = 1, \dots, n$.

Global minimum: $x^* = \left\{ \begin{array}{l} (-7.589, -7.708) \\ (-7.589, -1.425) \\ (-7.589, 4.858) \\ (-1.307, -7.708) \\ (-1.307, -1.425) \\ (-1.307, 4.858) \\ (4.976, -7.708) \\ (4.976, -1.425) \\ (4.976, 4.858) \end{array} \right\}$; $H(x^*) = -176.542$.

A.4 Ackley function (AK_n)

Definition: $AK_n(x) = 20 + e - 20e^{-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)}$.

Feasible search space: $-30.0 \leq x_i \leq 30.0$, $i = 1, \dots, n$.

Global minimum: $x^* = (0, \dots, 0)$; $AK_n(x^*) = 0$.

A.5 Rastrigin function (R_n)

Definition: $R_n = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$.

Feasible search space: $-5.12 \leq x_i \leq 5.12$, $i = 1, \dots, n$.

Global minimum: $x^* = (0, \dots, 0)$; $R_n(x^*) = 0$.

A.6 Griewangk function (G_n)

Definition: $G_n(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$.

Feasible search space: $-600.0 \leq x_i \leq 600.0$, $i = 1, \dots, n$.

Global minimum: $x^* = (0, \dots, 0)$; $G_n(x^*) = 0$.

A.7 Levy function (L_n)

Definition: $L_n(x) = \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) + (x_n - 1)(1 + \sin^2(2\pi x_n))$.

Feasible search space: $-10.0 \leq x_i \leq 10.0$, $i = 1, \dots, 4$,

$$-5.0 \leq x_i \leq 5.0, \quad i = 1, \dots, 7.$$

Global minimum: for $i=4$ $x^* = (1, 1, 1, -9.752)$; $L_n(x^*) = -21.502$;

for $i=7$ $x^* = (1, \dots, 1, -4.754)$; $L_n(x^*) = -11.504$.

A.8 Rosenbrock function (Rb_n)

Definition: $Rb_n(x) = \sum_{i=1}^{n/2} 100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2$

Feasible search space: $-10.0 \leq x_i \leq 10.0$, $i = 1, \dots, n$

Global minimum: $x^* = (1, \dots, 1)$; $Rb_n(x^*) = 0$

A.9 **Zakharov function** (Z_n)

Definition: $Z_n(x) = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5 \cdot i \cdot x_i)^2 + (\sum_{i=1}^n 0.5 \cdot i \cdot x_i)^4$

Feasible search space: $-5.0 \leq x_i \leq 10.0, i = 1, \dots, n$

Global minimum: $x^* = (0, \dots, 0); Z_n(x^*) = 0$

A.10 **DeJong function** (DJ)

Definition: $DJ(x) = x_1^2 + x_2^2 + x_3^2$

Feasible search space: $-2.56 \leq x_i \leq 5.12, i = 1, 2, 3$

Global minimum: $x^* = (0, \dots, 0); DJ(x^*) = 0$

A.11 **Trid function** (Tr_n)

Definition: $Tr_n(x) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$

Feasible search space: $-n^2 \leq x_i \leq n^2, i = 1, \dots, n$

Global minimum: for $i=5$ $Tr_n(x^*) = -30$

for $i=10$ $x_i = i \cdot (11 - i); Tr_n(x^*) = -210$

for $i=20$ $Tr_n(x^*) = -1520$

Appendix B Notation

TSP	travelling salesman problem
GO	global optimization
ACE	angiotensin converting enzyme
$F(x_i)$	continuous objective function
x_i	decision variable vector
x_l	lower bound
x_r	upper bound
D	feasible set
ACO	Ant colony optimization
PSO	Particle swarm optimization
GBAS	Graph-based Ant System
S-ACO	Simulation based Ant Colony Optimization
\Re^n	multidimensional space
$x(p)$	particle position in multidimensional space
$v(p)$	velocity of the particle in multidimensional space
$best(p)$	“particle best” particle position
$best(N(p))$	“local (neighbourhood) best” particle position
$best(X_{pop})$	“global best” particle position
QSAR	quantitative structure-activity relationship
GA	Genetic Algorithm
CGA	Continuous Genetic Algorithm
RNA	Ribonucleic Acid
SS	Scatter Search
MA	Memetic Algorithm
SA	Simulated Annealing
T	main control parameter in the cooling schedule (temperature)
ΔF	function difference
$F(pos_{i+1})$	function value of the resulting solution

$F(pos_i)$	function value of the current solution
ρ	probability
JSS	job shop scheduling problem
VNS	Variable Neighbourhood Search
N_i	set of neighbourhood structures
$N_i(x)$	set of solutions in the i^{th} neighbourhood of solution x
VND	Variable Neighbourhood Descent
RVNS	Reduced Variable Neighbourhood Search
CPU	Central Processing Unit
VNDS	Variable Neighbourhood Decomposition Search
SVNS	Skewed Variable Neighbourhood Search
PVNS	Parallel Variable Neighbourhood Search
GVNS	General Variable Neighbourhood Search
GRASP	Greedy Randomized Adaptive Search Procedure
TS	Tabu Search
TL	Tabu List
FIFO	First In First Out
GTS	Gradient Tabu Search
TD	Tabu Direction
TR	Tabu Region
TDV	Tabu Direction Vector
NMDV	New Move Direction Vector
R_{TR}	Radius of the Tabu Region
R_{up}	upper bound for each variable
R_{low}	lower bound for each variable
$coeff$	coefficient used to compute the radius R_{TR}
Δx_i	step size at the mildest ascent strategy
α	Tabu Direction coefficient
$alam$	first step size at the line search
N_{trial}	number of trial points at the diversification search
$LSIZE$	number of elements in tabu list
$rank_i$	recency ranked value
$rank_{max}$	maximum recency ranked value
$rank_{min}$	minimum recency ranked value

$NDIM$	dimension
$Iter_{main}$	
$Iter_{worst}$	loop termination numbers
$Iter_{loc}$	
$Iter_{MAS}$	
a_{min}^i	visited minima
b_{start}^i	starting points of the optimization from which the respective minimum
c_{new}^i	next trial point
BFGS	Broyden Fletcher Goldfarb and Shanno
LMOD	Low Mode search
DS	Diversification Search
AK_n	n -dimensional Ackley function
BR	Branin function
GP	Goldstein-Price function
G_n	n -dimensional Griewangk function
R_n	n -dimensional Rastrigin function
H	Hansen function
L_n	n -dimensional Levy function
Rb_n	n -dimensional Rosenbrock function
DJ	DeJoung function
Z_n	n -dimensional Zakharov function
Tr_n	n -dimensional Trid function
TRUST	Terminal Repeller Unconstrained Subenergy Tunneling
GOTS	Gradient Only Tabu Search
TSPA	Tabu Search with Powell's Algorithm
Fz_i^+	function value of the position with increasing
Fz_i^-	function value of the position with decreasing
D_i	direction for each variable
Fz_i	functional value needed for the linear ranking procedure
Fz_{max}	maximal Fz_i value
Fz_{min}	minimal Fz_i value
N	number of parameters
ECTS	Enhanced Continuous Tabu Search

ESA	Extended Simulated Annealing
CHA	Correlation Height Analysis
CTSS	Continuous Tabu Simplex Search
PES	Potential Energy Surface
BF	Boltzmann Factor
E	Energy
RGN	Random Generated Number
d	number of possible values of each torsion angle
d^N	number of conformations to be generated and minimized
MOLS	Mutually Orthogonal Latin Squares
ANN	Artificial Neural Networks
UFF	Universal Force Field
$coefficient_{min}$	minimal function values percentage
$coefficient_{max}$	maximal function values percentage
LYS	Lysine
VAL	Valine
ARG	Arginine
ACE	Angiotensin Converting Enzyme
TR	$\frac{1}{2}$ of the Tabu Region diameter
$BADMAX$	number of local minima which did not improved the solution
ω_i	torsion angle

References

- ¹ Neumaier A., Global optimization and constraint satisfaction., In: Bomze I., Emiris I., Neumaier A., Wolsey L. (eds.), *Proceedings of GICOLAG workshop* (of the research project Global Optimization, Integrating Convexity, Optimization, Logic Programming and Computational Algebraic Geometry), **2006**.
- ² Gendreau, M.; Laporte G.; Semet, F., A tabu search heuristic for the undirected selective travelling salesman problem, *European Journal of Operational Research*, **1998**, 106, pp. 539-545.
- ³ Sexton, R. S.; Alidaee, B.; Dorsey R. E.; Johnson, J. D., Global optimization for artificial neural networks: A tabu search application, *European Journal of Operational Research*, **1998**, 106, pp. 570-584.
- ⁴ Costamagna, E.; Fanni A.; Giacinto, G., A Tabu Search algorithm for the optimisation of telecommunication networks, *European Journal of Operational Research*, **1998**, 106, pp. 357-372.
- ⁵ Teh, Y. S.; Rangaiah, G. P., Tabu search for global optimization of continuous functions with application to phase equilibrium calculations., *Computers and Chemical Engineering*, **2003**, 27, pp. 1665-1679.
- ⁶ Lin, B. ; Miller, D. C., Tabu search algorithm for chemical process optimization, *Comp. Chem. Engineering*, **2004**, 22, 2287-2306.
- ⁷ Horst R. and Pardalos P. M.(eds), Handbook of Global Optimization, Kluwer Academic Publishers, Boston, MA, **1995**.
- ⁸ Horst R., Pardalos P.M. and Thoai N.V., Introduction to Global Optimization, Second Edition, Kluwer Academic Publishers, Boston, MA, **2000**.
- ⁹ Pardalos P. M. and Resende M. G. C. (eds.), Handbook of Applied Optimization, Oxford University Press, Oxford, **2002**.
- ¹⁰ Pardalos P.M. and Romeijn H. E (eds.), Handbook of Global Optimization, Kluwer Academic Publishers, Boston, MA, **2002**.
- ¹¹ Pardalos P. M., Romeijn H. E and Tuy H., Recent developments and trends in global optimization, *J. Comput. Appl. Math.*, **2000**, 124, pp. 209–228.
- ¹² Ribeiro C.C. and Hansen P. (eds.), Essays and Surveys in Metaheuristics, Kluwer Academic Publishers, Boston, MA, **2002**.
- ¹³ Voß S. Intelligent Search TH Darmstadt Habilitationsschrift, Darmstadt **1993**.
- ¹⁴ Osman I. H. and Kelly J. P., Meta-Heuristics: Theory and Applications, Kluwer Academic Publishers, Boston, MA, **1996**.
- ¹⁵ Glover F. and Kochenberger G. (eds.), Handbook of Metaheuristics, Kluwer Academic Publishers, Boston, MA, **2002**.
- ¹⁶ LaValle, S. M.; Finn, P. W.; Kavradi, L. E.; Latombe, J.-C. *Journal of Computational Chemistry*, **2000**, 21, pp. 731-747.
- ¹⁷ Seonggu R.; Baek S.-G.; Bogman L.; Chihyo P.; Nakyen C.; Chang S.L.; Young C.S.; Hoil C.; Jong S.K.; Heungsik Y.; Sung C.K.; Jong H.O. *Bioorganic and Medicinal Chemistry Letters*, **1998**, 8, pp. 2423-2426.
- ¹⁸ Gavrillets S., Fitness Landscapes and the Origin of Species. Number MPB-41 in Monographs in Population Biology. Princeton University Press, **2004**.
- ¹⁹ Renze J., Weisstein E. W., Extreme Value Theorem, MathWorld, A Wolfram Web Resource. <http://mathworld.wolfram.com/ExtremeValueTheorem.html>
- ²⁰ Weise T., Global Optimization Algorithms: Theory and Application, **2008** (e-book)
- ²¹ Müller-Merbach H., Heuristic and their design: a survey., *European Journal of Operational Research*, **1981**, 8, pp. 1-23.
- ²² Pearl J., Heuristic: Intelligent Search Techniques for Computer Problem Solving. Addison Wesley, Reading, **1984**.
- ²³ Zanakis, S. H., Evans, J. R., Vazacopoulos A. A., Heuristic methods and applications: a categorized survey., *European Journal of Operational Research*, **1989**, 415, pp. 298-311.
- ²⁴ Reeves C. R., Modern Heuristic Techniques for Combinatorial Problems. Blackwell, Oxford, **1993**.
- ²⁵ Polya G., How to solve it. Princeton University Press, Princeton, **1945**.
- ²⁶ Glover F., Future paths for integer programming and links to artificial intelligence, *Computers and Operations Research*, **1986**, 13, pp. 533–549.
- ²⁷ Goldberg D. E., Genetic Algorithms in Search, Optimization & Machine Learning, Reading: Addison-Wesley, **1989**.

- ²⁸ Aarts, E. H. L., Lenstra J.K. (eds.), Local Search in Combinatorial Optimization. Chichester: Wiley, **1997**.
- ²⁹ Corne D., Dorigo M., Glover F. (eds.), New Ideas in Optimization, London: McGraw-Hill, **1999**.
- ³⁰ Michalewicz Z., Genetic Algorithms+Data Structures=Evolution Programs. Third Edition, Berlin: Springer, **1996**.
- ³¹ Laporte G., Osman I.H. (eds.), Metaheuristics in Combinatorial Optimization., Annals of Operations Research **63**, **1996**.
- ³² Voss S., Martello S., Osman I.H., Roucairol C. (eds.), Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Boston: Kluwer, **1999**.
- ³³ Gendreau M., Guertin F., Potvin J.-Y., Taillard E.D., Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching., *Transportation Science*, **1999**, 33, pp. 381–390.
- ³⁴ Gendreau M., Laporte G., Semet F., A Dynamic Model and Parallel Tabu Search Heuristic for Real-Time Ambulance Relocation, *Parallel Computing*, **2001**, 27, pp. 1641–1653.
- ³⁵ Glover F., Laguna, M., Tabu Search Kluwer, Academic Publishers, MA, USA, **1997**.
- ³⁶ Taillard E. D., Gambardella L.M., Gendreau M., Potvin J.-Y., Adaptive memory programming: A unified view of metaheuristics., *European Journal of Operational Research*, **2001**, 135, pp. 1-16.
- ³⁷ Hopfield J. J., Tank D. W., Neural Computation of Decisions in Optimization Problems, *Biological Cybernetics*, **1985**, 52, pp. 141-152.
- ³⁸ Van den Bout D. E., Miller T. K., III Graph partitioning using annealed neural networks., *IEEE Transactions on Neural Networks*, **1990**, 1 , pp.192-203.
- ³⁹ Peterson C., Anderson J., Neural Networks and NP-complete optimization problems: a performance study on the graph bisection problem, *Complex Systems*, **1988**, 2, pp.59-89.
- ⁴⁰ Lee H.-M., Hsu C.-C., Neural network processing through energy minimization with learning ability to the multiconstraint zero-one knapsack problem., *Proceedings Tools for AI Conference*, Fairfax, Virginia, **1990**, pp. 548-555.
- ⁴¹ Tagliarini G. A., Page E.W., Solving constraint satisfaction problems with neural networks, *Proceedings IEEE International Conference on Neural Networks*, **1987**, 3, pp.741-747.
- ⁴² Kennedy M., Chua L., Neural networks for linear and nonlinear programming., *IEEE Transactions on Circuits and Systems*, **1988**, 35, pp.554-562.
- ⁴³ Looi C.-K., Neural network methods in combinatorial optimization, *Computers and Operations Research*, **1992**, 19, pp. 191-208
- ⁴⁴ Dorigo M., Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy, **1992**.
- ⁴⁵ Dorigo, M., Maniezzo, V., Colomi, A., The Ant System: An autocatalytic optimization process, Technical Report, Dept. of Electronics, Politecnico di Milano, Italy, **1991**.
- ⁴⁶ Dorigo M., Maniezzo V., Colomi A., The Ant System: Optimization by a colony of cooperating agents, *IEEE Trans. on Systems, Man, and Cybernetics*, **1996**, 26, pp. 1-13.
- ⁴⁷ Colomi A., Dorigo M., Maniezzo V., Distributed Optimization by Ant Colonies, *Proceedings of the First European Conference on Artificial Life*, Varela F.J., Bourgine P. (eds.), MIT Press, Cambridge, MA, **1992**, pp. 134-142.
- ⁴⁸ Deneubourg J.L., Goss S., Collective patterns and decision-making., *Ethology, Ecology & Evolution*, **1989**, 1, pp. 295-311.
- ⁴⁹ Goss S., Beckers R., Deneubourg J.L., Aron S., Pasteels J.M., How Trail Laying and Trail Following Can Solve Foraging Problems for Ant Colonies., In Hughes R.N. (eds.), NATO ASI Series, Vol. G 20, Behavioural Mechanisms of Food Selection, Springer-Verlag, Berlin, **1990**.
- ⁵⁰ Gambardella L.M., Taillard E.D., Dorigo M., Ant colonies for the QAP. Technical Report 4-97, IDSIA, Lugano, Switzerland, **1997**.
- ⁵¹ Maniezzo V., Colomi A., The ant system applied to the quadratic assignment problem., *IEEE Trans. Knowledge and Data Engineering*, **1999**, pp.769 – 778.
- ⁵² Dorigo M., Gambardella L. M., Ant colonies for the travelling salesman problem., *BioSystems*, **1997**, 43, pp. 73-81.
- ⁵³ Bullnheimer B., Hartl R. F., Strauss C., A new rank-based version of the ant system: a computational study., *Central European Journal for Operations Research and Economics*, **1999**, 7, pp. 25-38.
- ⁵⁴ Stützle T., Hoos H., The MAX-MIN ant system and local search for the travelling salesman problem., In: Baeck T., Michalewicz Z., Yao, X. (eds.), *IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference*, IEEE Press, **1997**, pp.309-314.
- ⁵⁵ Dorigo M., Gambardella L. M., Ant colony system: A cooperative learning approach to the travelling salesman problem., *IEEE Transactions on Evolutionary Computation*, **1997**, 1, pp. 53-66.
- ⁵⁶ Gambardella L. M., Dorigo M., Ant-Q: A reinforcement learning to the travelling salesman problem., In: *Proceeding of the Twelfth International Conference on Machine Learning*, Palo Alto, CA: Morgan Kaufmann, **1995**, pp.252-260.
- ⁵⁷ Gambardella L. M., Dorigo M., Solving symmetric and asymmetric TSPs by ant colonies., In: *Proceeding of the IEEE Conference on Evolutionary Computation*, ICEC96, **1996**, pp. 622-627.

- ⁵⁸ Merkle D., Middendorf M., Schmeck H., Ant colony optimization for resource-constrained project scheduling., In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, **2000**, pp 893–900.
- ⁵⁹ Bonabeau E., Henaux F., Guerin S., Snyers D., Kuntz P., Therulaz G., Routing in telecommunication networks with “Smart” ant-like agents., *Workshop on Intelligent Agents for Telecommunication Applications*, Springer Verlag, **1998**.
- ⁶⁰ Schoonderwoerd R., Holland O., Bruten J., Rothkrantz L., Ant-based load balancing in telecommunications networks., *Adaptive Behaviour*, **1996**, 5, pp. 169-207.
- ⁶¹ White T., Pagurek B., Oppacher F., Connection management using adaptive mobile agents. In: Arabnia H.R. (eds.), *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, CSREA Press, **1998**, pp. 802-809.
- ⁶² Di Caro G., Dorigo M., AntNet: Distributed stigmergetic control for communications networks., *Journal of artificial Intelligence Research*, **1998**, 9, pp.317-365
- ⁶³ Saleh M., Ghani A. A. A., Adaptive Routing in Packet-Switched Networks Using Agents Updating Methods. *Malaysian Journal of Computer Science*, **2003**, 16, pp. 1-10.
- ⁶⁴ Heusse M., Snyers D., Guerin S., Kuntz P., Adaptive agent-driven routing and load balancing in communication networks, *Adv. Complex Syst.*, **1998**, 1, pp. 237-254.
- ⁶⁵ Bullnheimer B., Hartl R.F., Strauss C., An improved Ant System algorithm for the Vehicle Routing Problem., *Annals of Operations Research*, **1999**, 89, pp. 319-328.
- ⁶⁶ Bullnheimer B., Strauss C., Tourenplanung mit dem Ant System., Technical Report 6, Institut für Betriebswirtschaftslehre, Universität Wien, **1996**.
- ⁶⁷ Gambardella L. M., Dorigo M., HAS-SOP: An hybrid ant system for the sequential ordering problem., Technical Report 11-97, IDSIA, Lugano, CH, **1997**.
- ⁶⁸ Costa D., Hertz A., Ants can colour graph., *Journal of Operational Research Society*, **1997**, 48, pp. 295-305.
- ⁶⁹ Michel R., Middendorf M., An island model based ant system look ahead for the shortest supersequence problem., In: Eiben A.E., Back T., Schoenauer M., Schwefel H.-P. (eds.), *Proceedings of PPSN-V, Fifth International Conference on Parallel Problem Solving from Nature*, Springer-Verlag, **1998**, pp.692-701.
- ⁷⁰ Daeyaert F., De Jonge M., Koymans L., Vinkers M., An Ant Algorithm for the Conformational Analysis of Flexible Molecules., *J. Comput. Chem.*, **2007**, 28, pp. 890–898.
- ⁷¹ Dorigo M., Blum C., Ant colony optimization theory: a survey., *Theoretical Computer Science*, **2005**, 344, pp. 243-278.
- ⁷² Gutjahr W.J., A graph-based ant system and its convergence., *Future Generation Computer Systems*, **2000**, 16, pp. 873-888.
- ⁷³ Gutjahr W.J., ACO algorithms with guaranteed convergence to the optimal solution., *Information Processing Letters*, **2002**, 82, pp. 145-153.
- ⁷⁴ Gutjahr W.J., A generalized convergence result for the Graph-based Ant System., *Probability in the Engineering and Informational Sciences*, **2003**, 17, pp. 545-569.
- ⁷⁵ Stützle T., Dorigo M., A short convergence proof for a class of ACO algorithms., *IEEE Trans. Evol. Comput.*, **2002**, 6, pp. 358-365.
- ⁷⁶ Sebastiani G., Torrisi G.L., An extended ant colony algorithm and its convergence analysis., *Methodology and Computing in Applied Probability*, **2005**, 7, pp. 249-263.
- ⁷⁷ Merkle D., Middendorf M., Modelling the Dynamics of Ant Colony Optimization., *Evolutionary Computation*, **2002**, 10, pp. 235-262.
- ⁷⁸ Gutjahr W.J., On the finite-time dynamics of ant colony optimization., *Methodology and Computing in Applied Probability*, **2006**, 8, pp. 105-133.
- ⁷⁹ Gutjahr W.J., First steps to the runtime complexity analysis of ant colony optimization., *Computers and Operations Research*, **2008**, 35, pp. 2711-2727.
- ⁸⁰ Gutjahr W.J., A converging ACO algorithm for stochastic combinatorial optimization., *Proc. SAGA 2003 (Stochastic Algorithms: Foundations and Applications)*, Springer LNCS 2827, **2003**, pp. 10-25
- ⁸¹ Gutjahr W.J., S-ACO: An ant-based approach to combinatorial optimization under uncertainty, *Proc. ANTS 2004 (4th International Workshop on Ant Colony Optimization and Swarm Intelligence)*, **2004**, Springer LNCS 3172, pp. 238-249.
- ⁸² Rauner M., Brailsford S.C., Gutjahr W.J., Zeppelzauer W., Optimal screening policies for diabetic retinopathy using a combined discrete-event simulation and ant colony optimization approach., *Proc. Int. Conference on Health Sciences Simulation*, Western MultiConference '05, eds.: Anderson J.G., Katzper M., **2005**, pp. 147-152.
- ⁸³ Brailsford S.C., Gutjahr W.J., Rauner M., Zeppelzauer W., Combined Discrete-event Simulation and Ant Colony Optimisation Approach for Selecting Optimal Screening Policies for Diabetic Retinopathy., *Computational Management Science*, **2007**, 4, pp. 59-83.
- ⁸⁴ Korošec P., Šilc J., Real-parameter optimization using stigmergy., In: *Proceedings of the Second International Conference on Bioinspired Optimization Methods and their Application*, **2006**, pp 73–84.

- ⁸⁵ Russel C. E., Kennedy J., A new optimizer using particle swarm theory.. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, **1995**, pp 39–43.
- ⁸⁶ Kennedy J., Russel C. E., Particle swarm optimization., In: *Proceedings of IEEE International Conference on Neural Networks*, **1995**, 4, pp. 1942–1948.
- ⁸⁷ Venter G., Sobieszcanski-Sobieski J., Particle swarm optimization., *AIAA Journal*, **2003**, 41, pp. 1583–1589.
- ⁸⁸ Cai T., Pan F., Chen J., Adaptive particle swarm optimization algorithm., In: *Proceedings of Fifth World Congress on Intelligent Control and Automation*, **2004**, 3, pp. 2245–2247,
- ⁸⁹ Russel C. E., Yuhui Shi, Comparison between genetic algorithms and particle swarm optimization., In: *Proceedings of the 7th International Conference on Evolutionary Programming*, **1998**, pp. 611–616.
- ⁹⁰ Angeline P. J., Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences., In: *Proceedings of the 7th International Conference on Evolutionary Programming VII*, **1998**, pp. 601–610.
- ⁹¹ Kennedy J., Spears W. M., Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator., *IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, **1998**.
- ⁹² Shi, Y., Eberhart R. C., A Modified Particle Swarm Optimizer., *IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, **1998**.
- ⁹³ Yuhui Shi, Russel C. E., Parameter Selection in Particle Swarm Optimization, *Evolutionary Programming VII, Lecture Notes in Computer Science*, **1998**, pp. 591–600.
- ⁹⁴ Clerc M., The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization., *IEEE Congress on Evolutionary Computation*, Washington D.C., **1999**, pp. 1951–1957.
- ⁹⁵ Kennedy J., Mendes R., Population structure and particle swarm performance., *IEEE Congress on Evolutionary Computation (CEC)*, **2002**.
- ⁹⁶ Kennedy J., Mendes R., Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms., *IEEE International Workshop on Soft Computing in Industrial Applications*, **2003**, pp. 45–50.
- ⁹⁷ Watts D. J., Small Worlds: The Dynamics of Networks Between Order and Randomness., Princeton University Press, **1999**.
- ⁹⁸ Watts D. J., Strogatz S. H., Collective dynamics of small-world networks., *Nature*, **1998**, 393, pp. 440–442.
- ⁹⁹ Suganthan P. N., Particle swarm optimiser with neighbourhood operator., *IEEE Congress on Evolutionary Computation (CEC)*, **1999**, pp. 1958–1962.
- ¹⁰⁰ Li X., Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization., *Lecture Notes on Computer Science*, **2004**, 3102, pp. 105–116.
- ¹⁰¹ Fourie P. C., Groenwold A. A., Particle Swarms in Size and Shape Optimization., *Proceedings of the International Workshop on Multidisciplinary Design Optimization*, Pretoria, South Africa, **2000**, pp. 97–106.
- ¹⁰² Fourie P. C., Groenwold A. A., Particle Swarms in Topology Optimization., *Extended Abstracts of the Fourth World Congress of Structural and Multidisciplinary Optimization*, Dalian, China, **2001**, pp. 52–53.
- ¹⁰³ Meissner M., Schmuker M., Schneider G., Optimized particle swarm optimization (opso) and its application to artificial neural network training., *BMC Bioinformatics*, **2006**, 7, 125.
- ¹⁰⁴ Rasmussen T.K., Krink T., Improved hidden markov model training for multiple sequence alignment by a particle swarm optimization-evolutionary algorithm hybrid., *BioSystems*, **2003**, 72, pp. 5–17.
- ¹⁰⁵ Parsopoulos K. E., Vrahatis M. N., Recent approaches to global optimization problems through particle swarm optimization., *Natural Computing*, **2002**, 1, pp. 235–306.
- ¹⁰⁶ Wu G., Hansen V., Kreysa E., Gemünd H.-P., Optimierung von fss-bandpassfiltern mit hilfe der schwarmintelligenz (particle swarm optimization)., *Advances in Radio Science*, **2006**, 4, pp. 65–71.
- ¹⁰⁷ Baltar A. M., Fontane D. G., A generalized multiobjective particle swarm optimization solver for spreadsheet models: application to water quality., In: *Proceedings of AGU Hydrology Days 2006*, Fort Collins, Colorado, **2006**, pp. 1–12.
- ¹⁰⁸ Cedeño W., Agrafiotis D. K., Using particle swarms for the development of qsar models based on k-nearest neighbour and kernel regression., *Journal of Computer-Aided Molecular Design*, **2003**, 17, pp. 255–263.
- ¹⁰⁹ Shen Q., Jiang J.-H., Jiao C.-X., Huan S.-Y., li Shen G., Yu R.-Q., Optimized partition of minimum spanning tree for piecewise modeling by particle swarm algorithm. qsar studies of antagonism of angiotensin ii antagonists., *Journal of Chemical Information and Modeling / J. Chem. Inf. Comput. Sci.*, **2004**, 44, pp. 2027–2031.
- ¹¹⁰ Bremermann H. J., Optimization through evolution and recombination., *Self-Organizing systems*, **1962**, pp. 93–10.
- ¹¹¹ Woodrow “Woody” Wilson Bledsoe, The evolutionary method in hill climbing: Convergence rates., Technical report, Panoramic Research Inc., Palo Alto, California, USA, **1962**.
- ¹¹² Woodrow “Woody” Wilson Bledsoe, An analysis of genetic populations., Technical report, Panoramic Research Inc., Palo Alto, California, USA, **1962**.
- ¹¹³ Bagley J. D., The Behaviour of Adaptive Systems which employ Genetic and Correlation Algorithms., PhD thesis, The University of Michigan, College of Literature, Science, and the Arts, Computer and Communication Sciences Department, Ann Arbor, MI, USA, **1967**.

- ¹¹⁴ Cavicchio D. J. Jr., Adaptive Search using Simulated Evolution., PhD thesis, The University of Michigan, College of Literature, Science, and the Arts, Computer and Communication Sciences Department, Ann Arbor, Michigan, USA, **1970**.
- ¹¹⁵ Frantz D. R., Nonlinearities in Genetic Adaptive Search., PhD thesis, The University of Michigan, Ann Arbor, MI, USA, **1972**.
- ¹¹⁶ Holland J., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, **1975**.
- ¹¹⁷ De Jong, K.A., An Analysis of the Behavior of a Class of Genetic Adaptive Systems., University of Michigan, Ann Arbor, MI, Ph.D. Thesis, **1975**.
- ¹¹⁸ Baker, J.E., Reducing Bias and Inefficiency in the Selection Algorithm., In: *Proceedings of the Second International Conference on Genetic Algorithms and their Applications*, New Jersey, USA, **1985**.
- ¹¹⁹ Mühlenbein H., Schlierkamp-Voosen D., Analysis of Selection, Mutation and Recombination in Genetic Algorithms., Technical Report 93/94, GMD, **1993**.
- ¹²⁰ Chipperfield A.J., Fleming P.J., Pohlheim H., Fonseca C.M., Genetic Algorithm Toolbox User's Guide., ACSE Research Report No. 512, University of Sheffield, **1994**.
- ¹²¹ Berthiau G., Siarry P., A Genetic Algorithm for Globally Minimizing Functions of Several Continuous Variables., In: *Second International Conference on Metaheuristics*, Sophia-Antipolis (France), **1997**.
- ¹²² Bäck T., Fogel D.B., Michalewicz Z. (eds.), Evolutionary Computation 1: Basic Algorithms and Operators., Institute of Physics Publishing, **2000**.
- ¹²³ Bäck T., Fogel D.B., Michalewicz Z. (eds.), Evolutionary Computation 2: Advanced Algorithms and Operators., Institute of Physics Publishing, **2000**.
- ¹²⁴ Holland J. H., Adaptation in Natural and Artificial Systems., The University of Michigan Press, Ann Arbor, **1975**.
- ¹²⁵ Quagliarella D., Periaux J., Poloni C., Winter G., Genetic Algorithms and Evolution Strategy in Engineering and Computer Science: Recent Advances and Industrial Applications., John Wiley & Sons Ltd, **1998**.
- ¹²⁶ Chelouah R., Siarry P., A Continuous Genetic Algorithm Designed for the Global Optimization of Multimodal Functions., *Journal of Heuristics*, **2000**, 6, pp. 191–213.
- ¹²⁷ Sekhon J. S., Mebane W. R., Genetic Optimization Using Derivatives., Draft, **1998**.
- ¹²⁸ Levine D., Application of a hybrid genetic algorithm to airline crew scheduling., *Computers & Operations Research*, **1996**, 23, pp. 547–558.
- ¹²⁹ Cleveland G. A., Smith S. F., Using genetic algorithms to schedule flow shop releases., In: *ICGA, Proceedings of the third International Conference on Genetic Algorithms*, **1989**, pp. 160–169.
- ¹³⁰ Kwan R. S. K., Kwan A. S. K., Wren A., Driver scheduling using genetic algorithms with embedded combinatorial traits., In: *Proceedings of 7th International Conference on Computer-Aided Scheduling of Public Transport*, Cambridge/Boston, MA, USA, **1997**, 471, pp. 81–102.
- ¹³¹ Beaty S. J., Genetic algorithms for instruction sequencing and scheduling., In: *Workshop on Computer Architecture Technology and Formalism for Computer Science Research and Applications*, **1992**.
- ¹³² Xiao Y. L., Williams D. E., Game: Genetic algorithm for minimization of energy, an interactive program for three-dimensional intermolecular interactions., *Computers & Chemistry*, **1994**, 18, pp. 199–201.
- ¹³³ Lopes da Cunha A. G., A multi-objective evolutionary algorithm for solving traveling salesman problems: Application to the design of polymer extruders., In: *Proceedings of the 7th International Conference on Adaptive and Natural Computing Algorithms*, **2005**, pp. 189–193.
- ¹³⁴ Deaven D. M., Ho K.M., Molecular geometry optimization with a genetic algorithm., *Physical Review Letters*, **1995**, 75, pp. 288–291.
- ¹³⁵ Cagnoni A., Dobrzeniecki A., Poli R., Yanch J., Genetic algorithm-based interactive segmentation of 3D medical images., *Image and Vision Computing*, **1999**, 17, pp. 881–895.
- ¹³⁶ Smigrodzki R., Goertzel B., Pennachin C., Coelho L., Prosdociimi F., Parker W. D. Jr., Genetic algorithm for analysis of mutations in parkinson's disease., *Artificial Intelligence in Medicine*, **2005**, 35, pp. 227–241.
- ¹³⁷ Yan H., Jiang Y., Zheng J., Peng C., Xiao S., Discovering critical diagnostic features for heart diseases with a hybrid genetic algorithm., In: Valafar F., Valafar H. (eds.), *Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, Las Vegas, Nevada, USA, **2003**, pp. 406–409.
- ¹³⁸ Vinterbo S. A., Ohno-Machado L., A genetic algorithm approach to multi-disorder diagnosis., *Artificial Intelligence in Medicine*, **2000**, 18, pp. 117–132.
- ¹³⁹ Min H., Smolinski T. G., Boratyn G. M., A genetic algorithm-based data mining approach to profiling the adopters and non-adopters of e-purchasing., In: Smari W. W. (eds.), *Information Reuse and Integration*, Third International Conference, **2001**, pp. 1–6.
- ¹⁴⁰ Kamrani A., Rong W., Gonzalez R., A genetic algorithm methodology for data mining and intelligent knowledge acquisition., *Computers & Industrial Engineering*, **2001**, 40, pp. 361–377.
- ¹⁴¹ Gopalan J., Alhajj R., Barker K., Discovering accurate and interesting classification rules using genetic algorithm., In: Crone S. F., Lessmann S., Stahlbock R. (eds.), *Proceedings of the 2006 International Conference on Data Mining*, Las Vegas, Nevada, USA, **2006**, pp. 389–395.

- ¹⁴² Szpiro G. G., A search for hidden relationships: Data mining with genetic algorithms., *Computational Economics*, **1997**, 10, pp. 267–277.
- ¹⁴³ Cordon O., Herrera F., Sánchez L., Evolutionary learning processes for data analysis in electrical engineering applications., In: Quagliarella D., Périaux J., Poloni C., Winter G. (eds.), *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science*, **1998**, pp. 205–224.
- ¹⁴⁴ Chai J., Ma S., Robust epipolar geometry estimation using genetic algorithm., *Pattern Recognition Letters*, **1998**, 19, pp. 829–838.
- ¹⁴⁵ Jedidi A., Caminada A., Finke G., 2-objective optimization of cells overlap and geometry with evolutionary algorithms., In: *Applications of Evolutionary Computing*, EvoWorkshops, **2004**, pp. 130–139.
- ¹⁴⁶ Hsieh I., Chen K.-C., Wang C. A., A genetic algorithm for the minimum tetrahedralization of a convex polyhedron., In: *Proceedings of the 15th Canadian Conference on Computational Geometry*, Halifax, Canada, **2003**, pp. 115–119.
- ¹⁴⁷ Kundu S., Seto K., Sugino S., Genetic algorithm based design of passive elements for vibration control., In: *Proceedings of 4th MOVIC Conference*, **1998**, 3, pp. 1183–1188.
- ¹⁴⁸ Kundu S., Seto K., Sugino S., Genetic algorithm application to vibration control of tall flexible structures., In: *Proceedings of The First IEEE International Workshop on Electronic Design, Test and Applications*, Christchurch, New Zealand, **2002**, pp. 333–337.
- ¹⁴⁹ Yuret D., Maza M., A genetic algorithm system for predicting the OEX., *Technical Analysis of Stocks & Commodities*, **1994**, pp. 58–64.
- ¹⁵⁰ El-Fakihy K., Yamaguchiz H., v. Bochmann G., A method and a genetic algorithm for deriving protocols for distributed applications with minimum communication cost., In: *Proceedings of Eleventh IASTED International Conference on Parallel and Distributed Computing and Systems*, Boston, USA, **1999**.
- ¹⁵¹ Yamamoto L. A. R., Tschudin C., Genetic evolution of protocol implementations and configurations., In: *IFIP/IEEE International workshop on Self-Managed Systems and Services*, Nice, France, **2005**.
- ¹⁵² Hill T, Lundgren A, Fredriksson R, Schiöth H. B., Genetic algorithm for large-scale maximum parsimony phylogenetic analysis of proteins., *Biochimica et Biophysica Acta*, **2005**, 1725, pp. 19-29.
- ¹⁵³ To C. C., Vohradsky J., A parallel genetic algorithm for single class pattern classification and its application for gene expression profiling in *Streptomyces coelicolor*., *BMC Genomics*, **2007**, 8, 49.
- ¹⁵⁴ Herrera F., López E., Rodríguez M. A., A linguistic decision model for promotion mix management solved with genetic algorithms., *Fuzzy Sets and Systems*, **2002**, 131, pp. 47 – 61.
- ¹⁵⁵ Green M.L., Miller R., Evolutionary molecular structure determination using Grid-enabled data mining., *Cluster Computing and the Grid*, **2004**, pp. 328 – 335.
- ¹⁵⁶ Wehrens R., Pretsch E., Buydens L. M. C., Quality Criteria of Genetic Algorithms for Structure Optimization., *J. Chem. Inf. Comput. Sci.*, **1998**, 38, pp. 151-157.
- ¹⁵⁷ Gondro C., Kinghorn B.P., A simple genetic algorithm for multiple sequence alignment., *Genetics and Molecular Research*, **2007**, 6, pp. 964-982.
- ¹⁵⁸ Wang S., Wang Y., Du W., Sun F., Wang X., Zhou C., Liang Y., A multi-approaches-guided genetic algorithm with application to operon prediction., *Artificial Intelligence in Medicine*, **2007**, 41, pp. 151-159.
- ¹⁵⁹ Willett P., Genetic algorithms in molecular recognition and design., *Trends in Biotechnology*, **1995**, 13, pp. 516-521.
- ¹⁶⁰ van Batenburg F. H., Gulyaev A. P., Pleij C. W., An APL-programmed genetic algorithm for the prediction of RNA secondary structure., *Journal of Theoretical Biology*, **1995**, 174, pp. 269-280.
- ¹⁶¹ Colomi A., Dorigo M., Maniezzo V., A genetic algorithm to solve the timetable problem., Tech. rep. 90-060 revised, Politecnico di Milano, Italy, **1992**.
- ¹⁶² Iba H., Akiba S., Higuchi T., Sato T., BUGS: A Bug-Based Search Strategy using Genetic Algorithms., *Journal of Japanese Society for Artificial Intelligence*, **1992**, 8, pp. 797-809.
- ¹⁶³ Ibrahim W., Amer H., An Adaptive Genetic Algorithm for VLSI Test Vector Selection., *Applied Simulation and Modelling*, **2006**, 522, pp. 86-92.
- ¹⁶⁴ Glover F., Heuristics for integer programming using surrogate constraints., *Decision Sciences* **1977**, 8, pp. 156–166.
- ¹⁶⁵ Glover F., A template for scatter search and path relinking., In: Hao J.-K., Lutton E., Ronald E., Schoenauer M., Snyers D. (eds.), *Artificial Evolution, Lecture Notes in Computer Science*, **1998**, 1363, pp. 13–54.
- ¹⁶⁶ Laguna M., Martí R., Scatter Search. Methodology and Implementations in C., Kluwer Academic Publishers, **2003**.
- ¹⁶⁷ Martí R., Laguna M., Glover F., Principles of scatter search. Special issue on scatter search and path relinking., *European Journal of Operational Research*, **2006**, 169, pp. 359–372.
- ¹⁶⁸ Glover F., Genetic algorithms and scatter search: Unsuspected potentials., *Statistics and Computing*, **1994**, 4, pp. 131– 140.
- ¹⁶⁹ De Jong K.A., Spears W.M., A formal analysis of the role of multi-point crossover in genetic algorithms., *Annals of Mathematics and Artificial Intelligence*, **1992**, 5, pp. 1–26.
- ¹⁷⁰ Kita H., A comparison study of self-adaptation in evolution strategies and real-coded genetic algorithms., *Evolutionary Computation Journal*, **2001**, 9, pp. 223–241.

- ¹⁷¹ Glover F., Genetic algorithms, evolutionary algorithms and scatter search: Changing tides and untapped potentials., *INFORMS Computer Science Technical Section Newsletter*, **1998**, 19, pp. 7–14.
- ¹⁷² Hansen P., Mladenovic N., Variable neighborhood search: Principles and applications., *European Journal of Operational Research*, **2001**, 130, pp. 449–467.
- ¹⁷³ Moscato P.A., On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms., Technical Report Caltech Concurrent Computation Program Report 826, Caltech, Caltech, Pasadena, CA, **1989**.
- ¹⁷⁴ Rego C., Alidaee B., Metaheuristic Optimization via memory and evolution: Tabu search and Scatter Search., Kluwer Academic Publishers Boston/Dordrecht/London, **2005**.
- ¹⁷⁵ Herrera F., Lozano M., Molina D., Continuous scatter search: An analysis of the integration of some combination methods and improvement strategies., *European Journal of Operational Research*, **2006**, 169, pp. 450–476.
- ¹⁷⁶ Perez J. A. M., Batista B. M., Laguna M., Scatter Search Based Metaheuristic for Robust Optimization of the Deploying of “DWDM” Technology on Optical Networks with Survivability., *Yugoslav Journal of Operations Research*, **2005**, 1, pp. 65–77.
- ¹⁷⁷ Hamiez J.-P., Hao J.-K., Scatter Search for Graph Coloring., *Artificial Evolution*, **2002**, pp. 195 – 213.
- ¹⁷⁸ García del Amo I. J., López F. G., Torres M. G., Batista B. M., Pérez J. A. M., Vega J. M. M., Martn R. R., Data Mining with Scatter Search., *Computer Aided Systems Theory*, **2005**, pp. 199–204.
- ¹⁷⁹ Alvarez A. M., González-Velarde J. L., De-Alba K., Scatter Search for Network Design Problem., *Annals of Operations Research*, **2005**, 138.
- ¹⁸⁰ Sousa S.H.G., Maschio C., Schiozer D.J., Scatter Search Metaheuristic Applied to the History-Matching Problem., Society of Petroleum Engineers 102975, **2006**.
- ¹⁸¹ Egea J. A., Rodríguez-Fernández M., Bangal J. R., Martí R., Scatter search for chemical and bio-process optimization., *Journal of Global Optimization*, **2007**, 37, pp. 481–503.
- ¹⁸² Cordon O., Damas S., Santamaria J., A Scatter search algorithm for the 3D image registration problem., *Parallel Problem Solving from Nature VIII*, **2004**, pp. 471–480.
- ¹⁸³ Laguna, M., Marti R., Neural Network Prediction in a System for Optimizing Simulations., *IIIE Transaction on Operations Engineering*, **2000**.
- ¹⁸⁴ Corberán A., Fernandez E., Laguna M., Marti R., Heuristic Solutions to the Problem of Routing School Buses with Multiple Objectives., *Journal of the Operational Research Society*, **2001**, 53, pp. 427–435.
- ¹⁸⁵ Glover F., Laguna M., Marti R., New Ideas and Applications of Scatter Search and Path Relinking New Optimization Techniques in Engineering, Onwubolu and Babu (eds.), Springer-Verlag **2004**.
- ¹⁸⁶ Rego, C., Leao P., A Scatter Search for the Vehicle Routing Problem., Research Report HCES-08-02, Hearin Center for Enterprise Science, School of Business Administration, University of Mississippi, **2002**.
- ¹⁸⁷ David B. C., Julio B. S., Clara C. R., Nature-inspired components of the Scatter Search., *First European Symposium on Nature-inspired Smart Information Systems*, Algarve, Portugal, **2005**.
- ¹⁸⁸ Van Laarhoven P. J. M., Aarts, E. H. L., Simulated Annealing: Theory and Applications. Dordrecht, Netherlands: Kluwer, **1987**.
- ¹⁸⁹ Metropolis M., Rosenbluth A., Rosenbluth M., A. Teller and E. Teller, Equation of state calculation by fast computing machines., *Journal of Chemical Physics*, **1953**, 21, pp. 1087–1092.
- ¹⁹⁰ Kirkpatrick S., Gelatt C.D. Jr., Vecchi M.P., Optimisation by simulated annealing., *Science*, **1983**, 220, pp. 671–680.
- ¹⁹¹ Cerny V., Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm., *Journal of Optimization Theory and Applications*, **1985**, 45, pp. 41–51.
- ¹⁹² Laarhoven P. J., Theoretical and Computational Aspects of Simulated Annealing., Stichting Mathematisch Centrum, Amsterdam, **1988**.
- ¹⁹³ Laarhoven P. J., Aarts E. H., Simulated Annealing: Theory and Applications., D. Reidel Publishing Company, Dordrecht, Holland, **1987**.
- ¹⁹⁴ Dueck G., Scheuer T., Threshold Accepting: A General Purpose Optimization Algorithm., *Journal of Computational Physics*, **1990**, 90, pp. 161–175.
- ¹⁹⁵ Perttunen C.D., Ramasurbramian N., Deterministic simulated annealing for system optimization Systems., Man and Cybernetics; Decision Aiding for Complex Systems, Conference Proceedings., IEEE International Conference, **1991**, pp. 603 - 608.
- ¹⁹⁶ Kaku I., Xiao Y., Xia G., The Deterministic Annealing Algorithms for Vehicle Routing Problems., *International Journal of Smart Engineering System Design*, **2003**, 5, pp. 327–339.
- ¹⁹⁷ Ingber L., Very fast simulated re-annealing., *Mathl. Comput. Modelling*, **1989**, 12, pp. 967–973.
- ¹⁹⁸ Ingber L., Adaptive simulated annealing (ASA): Lessons learned., *Control and Cybernetics*, **1996**, 25, pp. 33–54.
- ¹⁹⁹ Azencott, R. (ed.), Simulated Annealing: Parallelization Techniques., Chichester: Wiley, **1992**.
- ²⁰⁰ Verhoeven M.G.A., Aarts E.H.L., Parallel Local Search Techniques., *Journal of Heuristics*, **1996**, 1, pp. 43–65.

- ²⁰¹ Onbaşoğlu E., Özdamar L., Parallel Simulated Annealing Algorithms in Global Optimization., *Journal of Global Optimization*, **2001**, 19, pp. 27-50.
- ²⁰² Ram D.J., Sreenivas T.H., Subramaniam K.G., Parallel Simulated Annealing Algorithms., *Journal of Parallel and Distributed Computing*, **1996**, 37, pp. 207-212.
- ²⁰³ Chardaire P., Lutton J.L., Sutter A., Thermostistical persistency: A Powerful Improving Concept for Simulated Annealing Algorithms., *European Journal of Operational Research*, **1995**, 86, pp. 565– 579.
- ²⁰⁴ Hedar A.-R., Fukushima M., Hybrid Simulated Annealing and Direct Search Method for Nonlinear Unconstrained Global Optimization., *Optimization Methods and Software*, **2002**, 17, pp. 891-912.
- ²⁰⁵ Wah B.W., Chen Y., Hybrid constrained simulated annealing and genetic algorithms for nonlinear constrained optimization., *Evolutionary Computation, Proceedings of the 2001 Congress*, **2001**, 2, pp. 925 – 932.
- ²⁰⁶ Chiu Y.-C., Chang L.-C., Chang F.-J., Using a hybrid genetic algorithm-simulated annealing algorithm for fuzzy programming of reservoir operation., *Hydrological Processes*, **2007**, 21, pp. 3162 – 3172.
- ²⁰⁷ Chen D., Lee C.-Y., Park C. H., Hybrid genetic algorithm and simulated annealing (HGASA) in global function optimization., *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence*, **2005**, pp. 126 – 133.
- ²⁰⁸ Ingber L., Simulated annealing: practice versus theory., *Mathl. Comput. Modelling*, **1993**, 18, pp. 29-57.
- ²⁰⁹ Deboeck G. J. (ed.) Trading On The Edge, Wiley, **1994**.
- ²¹⁰ Crama Y., Schyns M., Simulated annealing for complex portfolio selection problems., *European Journal of Operational Research*, **2003**, 150, pp. 546-571.
- ²¹¹ Goffe W.L., Ferrier G.D., Rogers J., Global optimisation of statistical functions with simulated annealing., *J. Econometrics*, **1994**, 60, pp. 65-100.
- ²¹² Ingber L., Wehner M.F., Jabbour G.M., Barnhill T.M., Application of statistical mechanics methodology to term-structure bond-pricing models., *Mathl. Comput. Modelling*, **1991**, 15, pp. 77-98.
- ²¹³ Yamada T., Nakano R., Job-Shop Scheduling by Simulated Annealing Combined with Deterministic Local Search., In: Kluwer Academic Publishers, USA, **1996**, pp. 237-248.
- ²¹⁴ Ohlmann J. W., Thomas B. W., A Compressed-Annealing Heuristic for the Traveling Salesman Problem with Time Windows., *INFORMS Journal on Computing*, **2007**, 19, pp. 80-90.
- ²¹⁵ Černý V., Thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm., *Journal of Optimization Theory and Applications*, **1985**, 45, pp. 41–51.
- ²¹⁶ Buckham B. J., Lambert C., Simulated annealing applications., Seminar presentation: MECH620 Quantitative Analysis, Reasoning and Optimization Methods in CAD/CAM and Concurrent Engineering, Department of Mechanical Engineering, University of Victoria, **1999**.
- ²¹⁷ Wales D.J., Doye J.P.K., Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms., *Chem. Phys. Lett.*, **1997**, 269, pp. 408-412.
- ²¹⁸ Coleman T., Shalloway D., Wu Z., A parallel build-up algorithm for global energy minimizations of molecular clusters using effective energy simulated annealing., *Journal of Global Optimization*, **1994**, 4, pp. 171-85.
- ²¹⁹ Wilson S., Cui W., Applications of simulated annealing to peptides., *Biopolymers*, **1990**, 29, pp. 225-235.
- ²²⁰ Biswas R., Hamann D.R., Simulated Annealing of silicon atom clusters in Langevin molecular dynamics., *Phys. Rev. B*, **1986**, 34, pp. 895-901.
- ²²¹ Bollweg W., Kroll H., Maurer H., Numerical prediction of crystal structures by simulated annealing., In: Bomze I.M., Csendes T., Horst R., Pardalos P.M. (eds.), *Developments in Global Optimization*, **1997**, pp. 253-288.
- ²²² Pannetier J., Bassas-Alsina J., Rodriguez-Carvajal J., Calgnaert V., Prediction of crystal structures from crystal chemistry by simulated annealing., *Nature*, **1990**, 346, pp. 343-345.
- ²²³ Romeijn H.E., Zabinski Z.B., Graesser D.L. Neogi S., New Reflection Generator for Simulated Annealing in Mixed-Integer/Continuous Global Optimization., *Journal of Optimization Theory and Applications*, **1999**, 101, pp. 403-427.
- ²²⁴ Laplume D., Lamblin D., Simulated annealing applied to the optimal plastic design of circular steel plates., *International Journal for Numerical Methods in Engineering*, **2008**, 73, pp. 1047-1060.
- ²²⁵ Monem M. J., Namdarian R., Application of simulated annealing (SA) techniques for optimal water distribution in irrigation canals., *Irrigation and Drainage*, **2005**, 54, pp. 365-373.
- ²²⁶ Martinez-Alfaro H., Flugrad D. R., Collision-free path planning for mobile robots and/or agvs using simulated annealing., In: *Proceedings of the IEEE International Conference on Humans, Information and Technology*, **1994**, 1, pp. 270–275.
- ²²⁷ Malhotra A., Oliver J. H., Tu W., Synthesis of spatially and intrinsically constrained curves using simulated annealing. ASME Transactions., *Journal of Mechanical Design*, **1996**, 118, pp. 53–61.
- ²²⁸ Martínez-Alfaro H., Valenzuela-Rendón M., Using simulated annealing for paper cutting optimization., In: *Advances in Artificial Intelligence, Lecture Notes in Computer Science*, **2004**, 2972, pp. 11–20.
- ²²⁹ Sen M. K., Stoffa P. L., Nonlinear one-dimensional seismic waveform inversion using simulated annealing., *Geophysics*, **1991**, 56, pp. 1624–1638.

- ²³⁰ Mladenović N., Hansen P., Variable neighborhood search., *Computers Oper. Res.*, **1997**, 24, pp. 1097–1100.
- ²³¹ Brimberg J., Hansen P., Mladenović N., Taillard E., Improvements and comparison of heuristics for solving the Multisource Weber problem., *Oper. Res.*, **2000**, 48, pp. 444-460.
- ²³² Hansen P., Mladenović N., J-Means: A new local search heuristic for minimum sum-of-squares clustering., *Pattern Recognition*, **2001**, 34, pp. 405-413.
- ²³³ Belacel N., Mladenović N., Hansen P., Fuzzy J-Means: A new heuristic for Fuzzy clustering., *Pattern Recognition*, **2002**, 35, pp. 2193-2200.
- ²³⁴ Hansen P., Mladenović N., Perez-Britos D., Variable Neighborhood Decomposition Search., *Journal of Heuristics*, **2001**, 7, pp. 335-350.
- ²³⁵ Avanthay C., Hertz A., Zufferey N., Variable neighborhood search for graph coloring., *European J. Oper. Res.*, **2003**, 151, pp. 379–388.
- ²³⁶ Hansen P., Jaumard B., Mladenović N., Parreira A., Variable neighbourhood search for Weighted maximum satisfiability problem., *Les Cahiers du GERAD G- 2000-62*, Montr.eal, Canada, **2000**.
- ²³⁷ Lopez F.G., Batista B.M., Moreno Perez J.A., Moreno Vega J.M., The parallel Variable Neighborhood Search for the p -Median problem., Research Report, University of La Laguna, Spain, **2000**.
- ²³⁸ Crainic T., Gendreau M., Hansen P., Hoeb N., Mladenović N., Parallel Variable neighborhood search for the p -Median, *MIC'2001*, **2001**, pp. 595-599.
- ²³⁹ Rodriguez I., Moreno-Vega M., Moreno-Perez J., Heuristics for Routing-median problems., SMG Report, Universite Libre de Bruxelles, Belgium, **1999**.
- ²⁴⁰ Burke E., De Causmaecker P., Petrovićand S., Berghe G.V., Variable neighborhood search for nurse rostering problem, *MIC'2001*, **2001**, pp. 755-760.
- ²⁴¹ Brimberg J., Hansen P., Mladenović N., Taillard E., Improvements and comparison of heuristics for solving the Multisource Weber problem., *Oper. Res.*, **2000**, 48, pp. 444-460.
- ²⁴² Paraskevopoulos D. C., Repoussis P. P., Tarantilis C. D., Ioannou G., Prastacos G. P., A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows *Journal of Heuristics*, *Journal of Heuristics*, **2007**.
- ²⁴³ Martins S.L., Resende M.G.C., Ribeiro C.C., Pardalos P., A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy., *J. of Global Optimization*, **2000**, 17, pp. 267-283.
- ²⁴⁴ Ribeiro C., Uchoa E., Werneck R., A hybrid GRASP with perturbations for the Steiner problem in graphs., Technical report, Computer Science Department, Catholic University of Rio de Janeiro, **2001**.
- ²⁴⁵ Festa P., Pardalos P., Resende M., Ribeiro C., GRASP and VNS for Max-cut, *Proceedings of MIC'2001*, **2001**, pp. 371-376.
- ²⁴⁶ Loudin S., Boizumault P., VNS/LDS + CP: A hybrid method for constraint optimization in anytime contexts, *MIC'2001*, **2001**, pp. 761-765.
- ²⁴⁷ Shaw P., Using constraint programming and local search methods to solve vehicle routing problems., In: *Principles and practice of constraint programming*, **1998**, pp. 417-431.
- ²⁴⁸ Polacek M., Doerner K. F., Hartl R. F., Maniezzo V., A Variable Neighborhood Search for the Capacitated Arc Routing Problem with Intermediate Facilities, *Journal of Heuristics*, **2008**.
- ²⁴⁹ Burke E.K., Cowling P., Keuthen R., Effective local and guided Variable neighborhood search methods for the asymmetric traveling salesman problem., *Lecture Notes in Computer Science*, **1999**, pp. 203-212.
- ²⁵⁰ Gonzales C.G., Brito D.P., A Variable neighborhood search for solving the linear ordering problem, *MIC'2001*, **2001**, pp. 181-185.
- ²⁵¹ Avanthay C., Hertz A., Zufferey N., A variable neighborhood search for graph coloring., *European Journal of Operational Research*, **2003**, 151, pp. 379-388.
- ²⁵² Santana R., Larranaga P., Lozano J. A., Side chain placement using estimation of distribution algorithms., *Artificial Intelligence in Medicine*, **2007**, 39, pp. 49-63.
- ²⁵³ Liberti L., Lavor C., Maculan N., Double VNS for the Molecular Distance Geometry Problem., *Proc. of Mini Euro Conference on Variable Neighbourhood Search*, Tenerife, **2005**.
- ²⁵⁴ Belacel N., Cuperlovic-Culf M., Ouellette R., Boulassel M.R., The Variable Neighborhood Search Metaheuristic for Fuzzy Clustering cDNA Microarray Gene Expression Data., *Proceedings of IASTED-AIA-04 Conference*. Innsbruck, Austria, **2004**.
- ²⁵⁵ Belacel N., Mladenović N., Hansen P., Fuzzy J-Means: A new heuristic for Fuzzy clustering., *Pattern Recognition*, **2002**, 35, pp. 2193-2200.
- ²⁵⁶ Dražić M., Lavor C., Maculan N., Mladenović N., A continuous variable neighborhood search heuristic for finding the three-dimensional structure of a molecule., *European Journal of Operational Research* 185, **2008**, pp. 1265–1273.
- ²⁵⁷ Mladenovic N., Brimberg J., Hansen P., Moreno-Perez J., The p -Median Problem: A Survey of Metaheuristic Approaches., *European Journal of Operational Research*, **2007**, 179, pp. 927-939.
- ²⁵⁸ Hansen P., Brimberg J., Urosevic D., Mladenovic N., Primal-Dual Variable Neighbourhood for the Simple Plant Location Problem., *INFORMS Journal on Computing*, **2007**, 19, pp. 552-564.
- ²⁵⁹ Puchinger J., Raidl G. R., Pferschy U., The Multidimensional Knapsack Problem: Structure and Algorithms., Technical Report No. 006149, National ICT Australia, Melbourne, Australia, **2007**.

- ²⁶⁰ Fleszar K., Hindi K. S., New heuristics for one-dimensional bin-packing, *Computers and Operations Research*, **2002**, 29, pp. 821-839.
- ²⁶¹ Brimberg J., Hansen P., Lih, K.-W., Mladenović N., Breton M., An Oil Pipeline Design Problem., *Operations Research*, **2003**, 51, pp. 228-239.
- ²⁶² Hansen P., Mladenović N., Urošević D., Variable Neighbourhood Search for the Maximum Clique Problem., *Discrete Applied Mathematics*, **2004**, 145, pp. 117-125.
- ²⁶³ Ribeiro C., Souza C., Variable neighborhood descent for the degree-constrained minimum spanning tree problem., *Discrete Applied Mathematics*, **2002**, 118, pp. 43-54.
- ²⁶⁴ Festa P., Pardalos P., Resende M., Ribeiro C., GRASP and VNS for Max-cut, *Proceedings of MIC'2001*, **2001**, pp. 371-376.
- ²⁶⁵ Costa M.-C., Monclar F.-R., Zrikem M., Variable neighborhood search for the optimization of cable layout problem., *MIC'2001*, 2001, pp. 749-753.
- ²⁶⁶ Mladenović N., Urošević D., Variable neighborhood search for the k -cardinality tree., *MIC'2001*, **2001**, pp. 749-753.
- ²⁶⁷ Brimberg J., Urošević D., Mladenović N., Variable neighborhood search for the vertex weighted k ., *European Journal of Operational Research*, **2006**, 171, pp. 74-84.
- ²⁶⁸ Hu B., Leitner M., Raidl G. R., Computing generalized minimum spanning trees with variable neighborhood search., In: Hansen P., Mladenović N., Moreno Pérez J. A., Batista B. M., Moreno-Vega J. M. (eds.), *Proceedings of the 18th Mini Euro Conference on Variable Neighborhood Search*, Tenerife, Spain, **2005**.
- ²⁶⁹ Burke E., De Causmaecker P., Petrovic S., Berghe G.V., Variable neighbourhood search for nurse rostering problem, *MIC'2001*, **2001**, pp. 755-760.
- ²⁷⁰ Fleszar K., Hindi K.S., Solving the resource-constrained project scheduling problem by a variable neighborhood search., *European Journal of Operational Research*, **2004**, 155, pp. 402-413.
- ²⁷¹ Leitner M., Hu B., Raidl G.R., Variable neighborhood search for the generalized minimum edge biconnected network problem., In: Fortz B. (ed.), *Proceedings of the International Network Optimization Conference*, **2007**, Spa, Belgium.
- ²⁷² Hu B., Leitner L., Raidl G.R., The generalized minimum edge biconnected network problem: Efficient neighborhood structures for variable neighborhood search., Technical Report TR 186-1-07-02, Institute of Computer Graphics and Algorithms, Vienna University of Technology, **2007**.
- ²⁷³ Audet C., Brimberg J., Hansen P., Le Digabel S., Mladenović N., Pooling Problem: Alternate Formulations and Solution Methods., *Management Science*, **2004**, 50, pp. 761-776.
- ²⁷⁴ Glover F., Future paths for integer programming and links to artificial intelligence., *Computers and Operations Research*, **1986**, 13, pp. 533-549.
- ²⁷⁵ Hansen P., The Steepest Ascent Mildest Descent Heuristic for Combinatorial Programming., *Congress on Numerical Methods in Combinatorial Optimization*, Capri, Italy, **1986**.
- ²⁷⁶ Glover F., Tabu search—Part I., *ORSA Journal on Computing*, **1989**, 1, pp. 190-206.
- ²⁷⁷ Glover F., Tabu search—Part II., *ORSA Journal on Computing*, **1990**, 2, pp. 4-32.
- ²⁷⁸ de Werra D., Hertz A., Tabu Search Techniques: A tutorial and an application to neural networks, *OR Spektrum*, **1989**, pp. 131-141.
- ²⁷⁹ Glover F., Taillard E., de Werra D., A user's guide to Tabu Search, *Annals of Operations Research*, **1993**, 41, pp. 3-28.
- ²⁸⁰ Soriano P., Gendreau M., Diversification Strategies in Tabu Search Algorithms for the Maximum Clique Problems., *Annals of Operations Research*, **1996**, 63, pp. 189-207.
- ²⁸¹ Rochat Y., Taillard E. D., Probabilistic Diversification and Intensification in Local Search for Vehicle Routing., *Journal of Heuristics*, **1995**, 1, pp. 147-167.
- ²⁸² Glover F., Laguna M., Marti R., Fundamentals of Scatter Search and Path Relinking., *Control and Cybernetics*, **2000**, 39, pp. 653-684.
- ²⁸³ (a) Battiti R., Tecchiolli G., The reactive tabu search., *ORSA Journal on Computing*, **1994**, 6, pp.126 -140; (b) Battiti R., Tecchiolli R., The continuous reactive tabu search: blending combinatorial optimization and stochastic search for global optimization., *Annals of Operations Research*, **1996**, 63, pp.153-188.
- ²⁸⁴ Al-Sultan K.S., Al-Fawzan M.A., A tabu search Hooke and Jeeves algorithm for unconstrained optimization., *European Journal of Operational Research*, **1997**, 103, pp. 198-208.
- ²⁸⁵ Hooke R., Jeeves T. A., Direct search solution of numerical and statistical problems, *J. Ass. Comput. Mach.*, **1961**, 8, pp. 212-229.
- ²⁸⁶ Hedar A., Fukushima M., Tabu Search directed by direct search methods for non-linear global optimization., *European Journal of Operational Research*, **2006**, 170, pp. 329-349.
- ²⁸⁷ Nelder J. A., Mead R., A simplex method for function minimization, *Comput. J.*, **1965**, 7, pp. 308-313.
- ²⁸⁸ Cordeau J.-F., Laporte G., Mercier A., A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows., *Journal of the Operational Research Society*, **2001**, 52, pp. 928-936.
- ²⁸⁹ Toth P., Vigo D., The Granular Tabu Search and Its Application to the Vehicle-Routing Problem., *INFORMS Journal on Computing*, **2003**, 15, pp. 333-346.

- ²⁹⁰ Siarry P, Berthiau G., Fitting of tabu search to optimize functions of continuous variables., *International Journal for Numerical Methods in Engineering*, **1997**; 40, pp. 2449-2457.
- ²⁹¹ Cvijovic D., Klinowski J., Taboo search: An approach to the multiple minima problem, *Science*, **1995**, 667, pp. 664-666.
- ²⁹² Cvijovic D., Klinowski J., Handbook of Global Optimization, Kluwer: Boston, MA, **2002**, pp. 387-406.
- ²⁹³ Franzè F., Speciale N., A tabu-search-based algorithm for continuous multim minima problems., *Int. J. Numer. Meth. Engng*, **2001**, 50, pp. 665-680.
- ²⁹⁴ Chelouah R., Siarry P., Tabu search applied to global optimization., *European Journal of Operational Research*, **2000**, 123, pp. 256-270.
- ²⁹⁵ Fink A., Vos S., Generic application of tabu search methods to manufacturing problems., In: *Proceedings of the IEEE International Conference on Systems*, Piscataway, **1998**, 3, pp. 2385-2390.
- ²⁹⁶ Cao B., Glover F., Tabu search and ejection chains-application to a node weighted version of the cardinality-constrained TSP., *Management Science*, **1997**, 43, pp. 908-921.
- ²⁹⁷ Schneider J. J., Kirkpatrick S. (eds.), Tabu Search Applied to TSP, *Scientific Computation*, 2006, pp. 441-447.
- ²⁹⁸ McLoughlin J. F. III, Cedeno W., The enhanced evolutionary tabu search and its application to the quadratic assignment problem., In: *Proceedings of the 2005 conference on Genetic and evolutionary computation*, **2005**, pp. 975-982.
- ²⁹⁹ Glover F., Laguna M., Tabu search., In; Pardalos P. M., Du D.-Z. (eds.), Handbook of Combinatorial Optimization, Kluwer Academic Publishers, Springer Netherlands, **1998**, 3, pp. 621-757.
- ³⁰⁰ Sexton R. S., Alidaee B., Dorsey R. E., Johnson J. D., Global optimization for artificial neural networks: A tabu search application., *European Journal of Operational Research*, **1998**, 106, pp. 570-584.
- ³⁰¹ Szachniuk M., Popena L., Gdaniec Z., Adamiak R.W., Błażewicz J., NMR analysis of RNA bulged structures: Tabu search application in noe signal assignment., In: *Proceedings of the 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, San Diego, USA, **2005**, pp. 172-178.
- ³⁰² Hwang G.-J., Yin P.-Y., Yeh S.-H., A tabu search approach to generating test sheets for multiple assessment criteria., *IEEE Transactions on Education*, **2006**, 49, pp. 88-97.
- ³⁰³ Toth P., Vigo D., The granular tabu search and its application to the vehicle-routing problem., *INFORMS Journal on Computing*, **2003**, 15, pp. 333-346.
- ³⁰⁴ Lin B., Miller D. C., Tabu search algorithm for chemical process optimization., *Comp. Chem. Eng.*, **2004**, 28, pp. 2287-2306.
- ³⁰⁵ a) Baumann K., Albert H., von Korff M., A systematic evaluation of the benefits and hazards of variable selection in latent variable regression. Part I. Search algorithm, theory and simulations., *Journal of Chemometrics*, **2002**, 16, pp. 339-350. b) Baumann K., von Korff M., Albert H., A systematic evaluation of the benefits and hazards of variable selection in latent variable regression. Part II. Practical applications., *Journal of Chemometrics*, **2002**, 16, pp. 351-360.
- ³⁰⁶ Whitley D., Watson J.P., Complexity theory and the no free lunch theorem., In: Burke E.K., Kendall G. (eds.), Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. Kluwer, Boston, **2005**, pp. 317-339.
- ³⁰⁷ Jensen F., Introduction to Computational Chemistry, Wiley, New York, **1999**.
- ³⁰⁸ a) Cerjan C. J., Miller W. H., On finding transition states., *J. Chem. Phys.*, **1981**, 75, pp. 2800-2806; b) Simons J., Jorgensen P., Taylor H., Ozment J., Walking on potential energy surfaces., *J. Phys. Chem.*, **1983**, 87, pp. 2745-2753; c) Banerjee A., Adams N., Simons J., Shepard R., Search for stationary points on surfaces, *J. Phys. Chem.*, **1985**, 89, pp. 52-57.
- ³⁰⁹ Kolossvary I., Guida W. C., Low mode search. An efficient, automated computational method for conformational analysis: Application to cyclic and acyclic alkanes and cyclic peptides., *J. Am. Chem. Soc.*, **1996**, 118, pp. 5011-5019.
- ³¹⁰ a) Quapp W., A gradient-only algorithm for tracing a reaction path uphill to the saddle of a potential energy surface., *Chem. Phys. Lett.*, **1996**, 253, pp. 286-292. b) Quapp W., Reduced gradient methods and their relation to reaction paths, *J. Theor. Comput. Chem.*, 2003, 3, pp. 385-417.
- ³¹¹ Chelouah R., Siarry P., Meta Heuristics Advances and Trends in Local Search Paradigms for Optimization, Kluwer: Dordrecht, **1999**, pp. 49-61.
- ³¹² Chelouah R., Siarry P., A hybrid method combining continuous tabu search and Nelder-Mead simplex algorithms for the global optimization of multim minima functions., *Eur. J. Oper. Res.*, **2005**, 161, pp. 636-654.
- ³¹³ Floudas C. A., Pardalos P. M., Adjiman C. S., Esposito W. R., Gumus Z., Harding S. T., Klepeis J. L., Meyer C. A., Schweiger C. A. (eds.), Handbook of Test Problems for Local and Global Optimization; Kluwer: Boston, MA, **1999**.
- ³¹⁴ Jiang H., Cai W., Shao X., A random tunneling algorithm for the structural optimization problem., *Phys. Chem. Chem. Phys.*, **2002**, 4, pp. 4782-4788.
- ³¹⁵ Andressen R. S., Optimization, University of Queensland Press: St. Lucia, Australia, **1972**; p. 27.

- ³¹⁶ Korff M., Neue Algorithmen zur Variablenselektion und Variablenwichtung in der Chemometrie., Dissertation; Bayerischen Julius-Maximilians-Universität Würzburg, **2001**.
- ³¹⁷ Stanton A. F., Bleil R. E., Kais S., A New Approach to Global Minimization., *J. Comp. Chem.*, **1997**, 18, pp. 594-599.
- ³¹⁸ Barhen J., Protopopescu V., Reister D., TRUST: A Deterministic Algorithm for Global Optimization., *Science*, **1997**, 276, pp.1094-1097.
- ³¹⁹ Barbulescu L., Watson J., Whitley L., Dynamic representations and escaping local optima: Improving genetic algorithms and local search., *The seventeenth National Conference on Artificial Intelligence*, Austin, Texas, **2000**.
- ³²⁰ Cai W., Shao X., A fast annealing evolutionary algorithm for global optimization., *J. Comput. Chem.*, **2002**, 4, pp. 427-435.
- ³²¹ Potter M., De Jong K., The Third Parallel Problem Solving from Nature III., Springer-Verlag: Berlin, **1994**, pp. 249-257.
- ³²² Mühlebein N., Schomisch M., Born J., The parallel genetic algorithm as function optimizer, *J. Parallel Comput.*, **1991**, 17, pp. 619-632.
- ³²³ Voigt H.-M., Soft Genetic Operators in Evolutionary Algorithms., *Evolution and Biocomputation*, **1995**, pp. 123-141.
- ³²⁴ Wolfram Research, Inc. Mathematica Version 5.1; Wolfram Research, Inc.: Champaign, IL, **2004**.
- ³²⁵ Zhigljavsky A. A., Theory of Global Random Search., Kluwer: Dordrecht, Netherlands, **1991**.
- ³²⁶ Pintér J. D., Global Optimization in Action., Kluwer: Dordrecht, Netherlands, **1996**.
- ³²⁷ Törn A., Zilinskas A., Global Optimization., Springer-Verlag: New York, **1989**.
- ³²⁸ Ilonen J., Kamarainen J. K., Lampinen J., Differential Evolution Training Algorithm for Feed-Forward Neural Networks., *J. Neural Process Lett.*, **2003**, 17, pp. 93-105.
- ³²⁹ Plagianakos V. P., Vrahatis M. N., Parallel evolutionary training algorithms for. 'hardware-friendly' neural networks., *Nat. Comput.*, **2002**, 1, pp. 307-322.
- ³³⁰ Price K., Storn R., Differential Evolution., *Dr. Dobb's J.*, **1997**, 264, pp. 18-24.
- ³³¹ Storn R., Price K., Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces., *J. Global Optimization*, **1997**, 11, pp. 341-359.
- ³³² Storn R., Sytem Design by Constraint Adaptation and Differential Evolution., *IEEE Trans. on Evolutionary Computation*, **1999**, 3, pp. 22 - 34.
- ³³³ Stepanenko S., Engels B., Gradient Tabu Search, *J. Comput. Chem.*, **2007**, 28, 601-611.
- ³³⁴ Dennis J. E., Schnabel R. B., Numerical Methods for Unconstrained Optimization., Philadelphia: SIAM, **1996**.
- ³³⁵ Gill P.E., Murray W., Wright M. H., Practical Optimization., San Diego: Academic Press, **1981**.
- ³³⁶ Nocedal J., Wright S. J., Numerical Optimization., New York: Springer, **1999**.
- ³³⁷ Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P., Numerical Recipes in C++, 2nd ed., New York: Cambridge University Press, **2002**.
- ³³⁸ Stepanenko S., Engels B., New Tabu Search Based Global Optimization Methods Outline of Algorithms and Study of Efficiency., *J. Comput. Chem.*, **2008**, 29, pp. 768-780.
- ³³⁹ Glover F., Tabu Search for Nonlinear and Parametric Optimization (with Links to Genetic Algorithms), *Discrete Applied Mathematics*, **1994**, 49, pp. 231-255.
- ³⁴⁰ Laguna M., Marti R., Experimental Testing of Advanced Scatter Search Designs for Global Optimization of Multimodal Functions., *Journal of Global Optimization*, **2005**, 33, pp. 235-255.
- ³⁴¹ Lavor C., Maculan N., A Function to Test Methods Applied to Global Minimization of Potential Energy of Molecules: Theory and Practice in Optimization., In: Martínez J. M., Yuan J. Y. (guest eds.), *Numerical Algorithms*, **2004**, 35, pp. 287-300.
- ³⁴² Chelouah R., Siarry P., Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multimimima functions., *European Journal of Operational Research*, **2003**, 148, pp. 335-348.
- ³⁴³ Chelouah R., Siarry P., A continuous genetic algorithm designed for the global optimization of multimodal functions., *Journal of Heuristics*, **2000**, 6, pp. 191-213.
- ³⁴⁴ Roger L.S., Srinivas M. S., Rangaiah G.P., Global Optimization of Benchmark and Phase Equilibrium Problems Using Differential Evolution., *The Journal of the Institution of Engineers*, 46.
- ³⁴⁵ Birbil S.I., Fang S.-C., An Electromagnetism-like Mechanism for Global Optimization., *Journal of Global Optimization*, **2003**, 25, pp. 263-282.
- ³⁴⁶ Schlund S., Müller R., Graßmann C., Engels B., Conformational analysis of arginine in gas phase - A strategy for scanning the potential energy surface effectively., *J. Comput. Chem.*, **2008**, 29, pp. 407-415.
- ³⁴⁷ Schlund S., Schmuck C., Engels B., How Important Is Molecular Rigidity for the Complex Stability of Artificial Host-Guest Systems? A Theoretical Study on Self-Assembly of Gas-Phase Arginine., *Chem. Eur. J.*, **2007**, 13, pp. 6644-6653.

- ³⁴⁸ Helten H., Schirmeister T., Engels B., Theoretical Studies about the Influence of Different Ring Substituents on the Nucleophilic Ring Opening of Three-Membered Heterocycles and Possible Implications for the Mechanisms of Cysteine Protease Inhibitors., *Journal of Organic Chemistry*, **2005**, 70, pp. 233-237.
- ³⁴⁹ Adam W., Bottke N., Engels B., Krebs O., An Experimental and Computational Study on the Reactivity and Regioselectivity for the Nitrosoarene Ene Reaction: Comparison with Triazolinedione and Singlet Oxygen., *J. Am. Chem. Soc.*, **2001**, 123, pp. 5542-5548.
- ³⁵⁰ Musch P., Engels B., The Importance of the Ene Reaction for the C²-C⁶ Cyclization of Enyne-Allenenes., *J. Am. Chem. Soc.*, **2001**, 123, pp. 5557-5562.
- ³⁵¹ Anet F.A. L., Inflection Points and Chaotic Behavior in Searching the Conformational Space of Cyclononane., *J. Am. Chem. Soc.*, **1990**, 112, pp. 7172-7178.
- ³⁵² a) Mezey P.G., Potential Energy Hypersurfaces., Elsevier: New York, **1987**; b) Boswell D.R., Coxon E.E., Coxon J. M., Molecular Modeling of Carbohydrates., *Advances in Molecular Modeling*, **1995**, 3, pp. 195-224.
- ³⁵³ Schlund S., Schmuck C., Engels B., Knock-Out" Analogues as a Tool to Quantify Supramolecular Processes: A Theoretical Study of Molecular Interactions in Guanidiniocarbonyl Pyrrole Carboxylate Dimers., *J. Am. Chem. Soc.*, **2005**, 127, pp. 11115-11124.
- ³⁵⁴ Suter H. U., Engels B., Theoretical-Study of Electron-Spin-Resonance Parameters - H₂CN and H₂CO⁺., *Journal of Chemical Physics*, **1994**, 100, pp. 2936-2942.
- ³⁵⁵ Schlick T., Optimization methods in computational chemistry., In: Lipkowitz K. B., Boyd D. B. (eds.), *Reviews in Computational Chemistry*, VCH Publishers, New York, **1992**, 3, pp. 1-71.
- ³⁵⁶ Ngo J. T., Marks J., Computational complexity of a problem in molecular structure prediction., *Protein Eng.*, **1992**, 5, pp. 313-321.
- ³⁵⁷ Leach A. R., A survey of methods for searching the conformational space of small and medium-sized molecules., In: Lipkowitz K. B., Boyd D. B. (eds.), *Reviews in Computational Chemistry*, VCH Publishers, New York, **1991**, 2, pp. 1-55.
- ³⁵⁸ Howard A.E., Kollman P.A., An analysis of current methodologies for conformational searching of complex molecules., *J. Med. Chem.*, **1988**, 31, pp. 1669-1675.
- ³⁵⁹ Leach A. R., Molecular Modelling Principles and Applications., 2nd Edition, Prentice Hall, Pearson Education, Harlow, England, **2001**.
- ³⁶⁰ Vásquez M., Nemethy G., Scheraga, H. A., Conformational energy calculations on polypeptides and protein., *Chem. Rev.*, **1994**, 94, pp. 2183-2239.
- ³⁶¹ Eisenhaber F., Persson B., Argos P., Protein structure prediction: Recognition of primary, secondary, and tertiary structural features from amino acid sequence., *Crit. Rev. Biochem. Mol. Biol.*, **1995**, 30, pp. 1-94.
- ³⁶² Bohm G., New approaches in molecular structure prediction., *Biophys.Chem.*, **1996**, 59, pp. 1-32.
- ³⁶³ Neumaier A., Molecular modeling of proteins and mathematical prediction of protein structure., *SIAM Rev.*, **1997**, 39, pp. 407-460.
- ³⁶⁴ Scheraga H. A., Lee J., Pillardy J., Ye Y. J., Liwo A., Ripoll D., Surmounting the multiple-minima problem in protein folding., *J. Global Optim.*, **1999**, 15, pp. 235-260.
- ³⁶⁵ Floudas C. A., Klepeis J. L., Pardalos P. M., Global optimization approaches in protein folding and peptide docking., In: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, American Mathematical Society, New Jersey, **1999**, 47, pp. 141-171.
- ³⁶⁶ Goodman J. M., Still W. C., An Unbounded Systematic Search of Conformational Space, *J. Comput. Chem.*, **1991**, 12, pp. 1110-1117.
- ³⁶⁷ Bruccoleri R. E., Karplus M., Prediction of the folding of short polypeptide segments by uniform conformational sampling., *Biopolymers*, **1987**, 26, pp. 137-168.
- ³⁶⁸ Gibson K. D., Scheraga H. A., Revised algorithms for the build-up procedure for predicting protein conformations by energy minimization., *J. Comput. Chem.*, **1987**, 8, pp. 826-834.
- ³⁶⁹ Chang G., Guida W.C., Still, W.C., An internal coordinate Monte Carlo method for searching conformational space, *J. Am. Chem. Soc.*, **1989**, 111, pp. 4379-4386.
- ³⁷⁰ Li Z., Scheraga H.A., Monte Carlo-minimization approach to the multiple-minima problem in protein folding., *Proc. Natl. Acad. Sci. USA*, **1987**, 4, pp. 6611-6615.
- ³⁷¹ von Freyberg B., Braun W., Efficient search for all low energy conformations of polypeptides by Monte Carlo methods., *J. Comput. Chem.*, **1991**, 12, pp. 1065-1076.
- ³⁷² Senderowitz H., Guarnieri F., Still W.C., A smart Monte Carlo techniques for free energy simulations of multiconformational molecules. Direct calculations of the conformational populations of organic molecules., *J. Am. Chem. Soc.*, **1995**, 117, pp. 8211-8219.
- ³⁷³ Ozkan S.B., Meirovitch H., Conformational search of peptides and proteins: Monte Carlo minimization with an adaptive bias method applied to the heptapeptide deltorphin., *J. Comput. Chem.*, **2004**, 25, pp. 565-572.
- ³⁷⁴ Ripoll D. R., Scheraga H. A., The multiple-minima problem in the conformational analysis of polypeptides. III. An electrostatically driven Monte Carlo method: Tests on enkephalin., *J. Protein Chem.*, **1989**, 8, pp. 263-287.
- ³⁷⁵ Wilson S. R., Cui W., Moskowitz J.W., Applications of simulated annealing to the conformational analysis of flexible molecules., *J. Comput. Chem.*, **1991**, 12, pp. 342-349.

- ³⁷⁶ Morales L.B., Garduno-Juárez R., Romero D., Applications of simulated annealing to the multiple-minima problem in small peptides., *J. Biomol. Struct. Dyn.*, **1991**, 8, pp. 721-735.
- ³⁷⁷ Lipton M., Still W.C., The multiple minimum problem in molecular modeling. Tree searching internal coordinate conformational space., *J. Comput. Chem.*, **1998**, 9, pp. 343-355.
- ³⁷⁸ Yang Y., Liu H., Genetic algorithms for protein conformation sampling and optimization in a discrete backbone dihedral angles space., *J. Comput. Chem.*, **2006**, 27, pp. 1593-1602.
- ³⁷⁹ Le Grand S.M., Merz K.M., The genetic algorithm and protein structure prediction., In: Merz K. M., Le Grand S. M. (eds.), *the Protein Folding Problem and Tertiary Structure Prediction*, Birkhäuser, Boston, **1994**, pp.109-124.
- ³⁸⁰ Schulze-Kremer S., Genetic algorithm and protein folding., In: Webster D. M. (ed.), *Protein Structure Prediction – Methods and Protocols*, Humana Press, New Jersey, **2000**, pp. 175-222.
- ³⁸¹ Taylor W.R., Aszódi A., Building protein folds using distance geometry: Towards a general modeling and prediction method., In: Merz K.M., Le Grand S.M. (eds.), *The Protein Folding Problem and Tertiary Structure Prediction*, Birkhäuser, Boston, **1994**, pp. 165-192.
- ³⁸² Grippen G.M., Havel, T.F., Distance Geometry and Molecular Conformation, Wiley, New York, **1988**.
- ³⁸³ Preishoff C.E., Dixon J.S., Improvements to the distance geometry algorithms for conformational sampling of cyclic structures., *J. Comput. Chem.*, **1992**, 13, pp. 565-569.
- ³⁸⁴ Weiner P.K., Profeta S., Wipff G., Havel T., Kuntz I.D., Langridge R., Kollman P.A., A distance geometry study of ring systems: application to cyclooctane, 18-crown-6, cyclododecane and androstanedione., *Tetrahedron*, **1983**, 38, pp. 1113-1121.
- ³⁸⁵ Grippen G. M., Exploring the conformational space of cycloalkanes by linearized embedding., *J. Comput. Chem.*, **1992**, 13, pp. 351-361.
- ³⁸⁶ a)Piela L., Kostrowicki J., Scheraga H. A., The multiple-minima problem in the conformational analysis of molecules: Deformation of potential energy hyper surface by diffusion equation method., *J. Phys. Chem.*, **1989**, 93, pp. 3339–3346. b) Kostrowicki J., Scheraga H. A., Application of diffusion equation method for global optimization to oligopeptides., *J. Phys. Chem.*, **1992**, 96, pp. 7442–7449.
- ³⁸⁷ Sun Y., Kollman P.A., Conformational sampling and ensemble generation by molecular dynamics simulations: 18-Crown-6 as a test case., *J. Comput. Chem.*, **1992**, 13, pp. 33-40.
- ³⁸⁸ Byrne D., Li J., Platt E., Robson B., Weiner P., Novel Algorithms for Searching Conformational Space., *J. Comput.-aided. Mol. Des.*, **1994**, 8, pp. 67-82.
- ³⁸⁹ Li Z., Laidig K. E., Daggett V., Conformational Search Using a Molecular Dynamics-Minimization Procedure: Applications to Clusters of Coulombic Charges, Lennard-Jones Particles, and Waters., *Journal of Computational Chemistry*, **1998**, 19, pp. 60-70.
- ³⁹⁰ Gibson K. D., Scheraga H. A., Variable step molecular dynamics: An exploratory technique for peptides with fixed geometry., *J. Comput. Chem.*, **1990**, 11, pp. 468–486.
- ³⁹¹ a) Goto H., Osawa E., Corner flapping: a simple and fast algorithm for exhaustive generation of ring conformations., *J. Am. Chem. Soc.*, **1989**, 111, pp. 8950-8951. b) Goto H., Further Developments in the Algorithm for Generating Cyclic Conformers. Test with Cycloheptadecane., *Tetrahedron Lett.*, **1992**, 33, pp. 1343-1346.
- ³⁹² Goto H., Osawa E., An Efficient Algorithm for Searching Low-energy Conformers of Cyclic and Acyclic Molecules, *J. Chem. Soc.*, **1993**, Perkin Trans. 2, pp. 187-198.
- ³⁹³ Kolossvary I., Guida W.C., Torsional flexing: Conformational searching of cyclic molecules in biased internal coordinate space., *J. Comput. Chem.*, **1993**, 14, pp. 691-698.
- ³⁹⁴ Kolossvary I., Guida, W.C., Low mode conformational search elucidated: application to C₃₉H₈₀ and flexible docking of 9-deazaguanine inhibitors into pnp., *J. Comp. Chem.*, **1999**, 20, pp. 1671-1684.
- ³⁹⁵ Kolossvary I., Keseru G., Hessian-free low mode conformational search for large-scale protein loop optimization: Application to c-jun N-terminal kinase JNK3., *J. Comp. Chem.*, **2001**, 22, pp. 21-30.
- ³⁹⁶ Vengadesan K., Gautham N., A new conformational search technique and its applications., *Current Science*, **2005**, 88, pp. 1759-1770.
- ³⁹⁷ a) Baker J., Hehre W. J., Geometry optimization in cartesian coordinates: The end of the Z-matrix?, *J. Comput. Chem.*, **1991**, 12, pp. 606-610; b) Baker J., Techniques for geometry optimization: A comparison of cartesian and natural internal coordinates., *J. Comput. Chem.*, **1993**, 14, pp. 1085-1100.
- ³⁹⁸ Cramer C. J., Essentials of computational chemistry: Theories and models., John Wiley& Sons: Chichester, **2004**, 2nd.ed.
- ³⁹⁹ Pulay P., Fogarasi G. Pang F., Boggs J. E., Systematic ab initio gradient calculation of molecular geometries, force constants, and dipole moment derivatives., *J. Am. Chem. Soc.*, **1979**, 101, pp. 2550-2560.
- ⁴⁰⁰ Hehre W. J., Radom L., v. Schleyer P. R., Pople J. A., Ab initio molecular orbital theory., Wiley, New York, **1986**.
- ⁴⁰¹ Chass G. A., Sahai M. A., Law J. M. S., Lovas S., Farkas Ö., Perczel A., Rivail J.-L., Csizmadia I. G., Toward a computed peptide structure database: The role of a universal atomic numbering system of amino acids in peptides and internal hierarchy of database., *Int. J. Quantum. Chem.*, **2002**, 90, pp. 933-968.

-
- ⁴⁰² Sahai M. A., Lovas S., Chass G. A., Botond P., Csizmadia I. G., A modular numbering system of selected oligopeptides for molecular computations: using pre-computed amino acid building blocks., *J. Mol. Struct.*, **2003**, 666-667, pp. 169-218.
- ⁴⁰³ Frey R. F., Coffin J., Newton S. Q., Ramek M., Cheng V. K. W., Momany F. A., Schäfer L., Importance of correlation-gradient geometry optimization for molecular conformational analyses., *J. Am. Chem. Soc.*, **1992**, 114, pp. 5369-5377.
- ⁴⁰⁴ Jalkanen K. J., Suhai S., N-Acetyl-L-Alanine N'-Methylamide: A Density Functional Analysis of the Vibrational Absorption and Vibrational Circular Dichroism Spectra., *Chem. Phys.*, **1996**, 208, pp. 81-116.
- ⁴⁰⁵ Jalkanen K. J., Nieminen R. M., Knapp-Mohammady M., Suhai S., A vibrational analysis of various isotopomers of L-Alanyl-L-Alanine in aqueous solution: vibrational absorption (VA), vibrational circular dichroism (VCD), Raman and Raman optical activity (ROA) spectra., *Int. J. Quantum. Chem.*, **2003**, 92, pp. 239-259.
- ⁴⁰⁶ Mazur A. K., Abagyan R. A., New Methodology For Computer-Aided Modelling of Biomolecular Structure and Dynamics. 1. Non-Cyclic Structures., *J. Biomol. Struct. Dyn.*, **1989**, 6, pp. 815-832.
- ⁴⁰⁷ Abagyan R. A., Mazur A. K., New Methodology for Computer-Aided Modelling of Biomolecular Structure and Dynamics. 2. Local Deformations and Cycles., *J. Biomol. Struct. Dyn.*, **1989**, 6, pp. 833-845.
- ⁴⁰⁸ Abagyan R. A., Totrov M. M., Kuznetsov D. A., Icm: A New Method For Protein Modeling and Design: Applications To Docking and Structure Prediction From The Distorted Native Conformation., *J. Comp. Chem.*, **1994**, 15, pp. 488-506.
- ⁴⁰⁹ Chass A. G., Toward a computed structure database: methodology for effective molecular orbital computations., *Journal of Molecular Structure (Theochem)*, **2003**, 666-67, pp. 61-67.
- ⁴¹⁰ Echenique P., Alonso J. L., Definition of systematic approximately separable and modular internal coordinates (SASMIC) for macromolecular simulation., *J. Comput. Chem.*, **2006**, 27, pp. 1076-1087.
- ⁴¹¹ Rappé A. K., Casewit C. J., Colwell K. S., Goddard III W. A., Skid, W. M., UFF, a Full Periodic Table Force Field for Molecular Mechanics and Molecular Dynamics Simulations, *J. Am. Chem. Soc.*, **1992**, 114, pp. 10024-10039.
- ⁴¹² Casewit C. J., Colwell K. S., Rappé A. K., Application of a Universal Force Field to Organic Molecules., *J. Am. Chem. Soc.*, **1992**, 114, pp. 10035-10046.