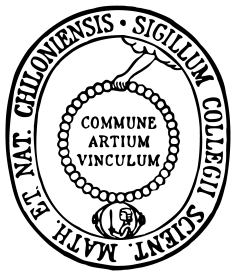
Genetic Algorithms in Theoretical Chemistry

Development and Applications of a General Purpose Framework.

Doctoral Thesis



submitted in partial fulfillment
of the requirements for the doctoral degree
Dr. rer. nat.
to the Faculty of Mathematics and Natural Sciences
of the Christian-Albrechts-University of Kiel

Submitted by

Johannes M. Dieterich

Kiel 2010

Referent	Prof. Dr. Bernd Hartke
Korreferent	Prof. Dr. Roy L. Johnston
Tag der mündlichen Prüfung	11. Oktober 2010
Zum Druck genehmigt	11. Oktober 2010
Der Dekan	Prof. Dr. Lutz Kipp

Kurzzusammenfassung

Die vorliegende Arbeit behandelt die Implementierung und Anwendung genetischer Algorithmen in der Theoretischen Chemie. Genetische Algorithmen sind eine elegante und effiziente Möglichkeit, *NP-harte* Probleme global zu optimieren bzw. zu lösen [1].

Es wurde ein neues Programmpaket, OGOLEM, entwickelt. Diese Entwicklung ist rein objektorientiert und hochparallel. Das Programmpaket besitzt drei Hauptteile zur Optimierung von Clusterstrukturen beliebiger Zusammensetzung, zur Parameteroptimierung und zum Design von optimal schaltbaren Molekülen.

Es werden die verwendeten Methoden und Techniken vorgestellt und eingeordnet. Alle Programmteile sind durch Anwendungen repräsentiert. Diese Anwendungen umfassen teilweise mehr als einen Programmteil. Dies kann als Hinweis auf die gute Verzahnung der verschiedenen Programmteile gesehen werden.

Die vorgestellten Anwendungen umfassen ausführliche Leistungstests des Frameworks, die Parametrisierung eines LJ(6,16,2)-Kraftfeldes und die Anwendung desselbigen auf die Strukturoptimierung stark gemischter Lennard-Jones-Cluster. Des Weiteren werden Kanamycin-A-Dimere optimiert und mit Blick auf das Experiment analysiert. Als Beispiel für das Design optimal schaltbarer Moleküle werden auf der Basis des verbrückten Azobenzols Schalter entwickelt, deren vertikale Anregungsenergien identisch zur Wellenlänge kommerziell erhältlicher Laserpointer ist.

Die Anwendungen werden in den Gesamtzusammenhang eingeordnet und wenn nötig durch Informationen ergänzt, die zu detailliert für die entsprechenden Publikationen sind.

Ein Ausblick wird sowohl mit Blick auf weitere Entwicklungsmöglichkeiten des Programms als auch auf mögliche Anwendungen oder Vertiefungen der vorgestellten Anwendungen versucht.

Short Summary

This work deals with the implementation and application of genetic algorithms in theoretical chemistry. Genetic algorithms provide elegant and efficient means of optimizing/solving *NP-hard* problems [1].

A new framework, OGOLEM was developed for this purpose. The development is purely object-oriented and highly parallel. The framework is separated into three major parts, the global optimization of cluster structures of arbitrary composition, the optimization of parameters and the design of optimally switchable molecules.

The used methods and techniques are presented and put into the context. All program parts are presented through applications. These applications may cover more than one program part. This can be seen as a hint on the good interlocking of the different program parts.

The presented applications include detailed performance benchmarks of the framework, the parametrization of a LJ(6,16,2)-type force field und its application to the optimization of highly mixed Lennard-Jones clusters. Furthermore, dimers of Kanamycin A are optimized and analyzed with the aim of providing information for experiment. As an example for the design of optimally switchable molecules, switches based on the bridged azobenzene backbone are developed. Their vertical excitation energies are tuned to be in agreement with the wavelengths of commercially available laser pointers.

All applications are put into the overall context and, if necessary, additional information is added which was too detailed to appear in the corresponding publication.

It is attempted to give an outlook both for future development possiblities of the framework as well as possible applications or further studies on the presented ones.

Contents

1 Introduction **1**

2 Techniques and Methods **5**

2.1 Hypersurfaces: Introduction 6
2.2 Minimum Finding: Introduction 7
2.3 Minimum Finding: Local Techniques 7
2.3.1 Methods using Derivatives 8
2.3.2 Methods without Derivatives 10
2.4 Minimum Finding: Global Techniques 11
2.4.1 Genetic Algorithms 12
2.4.2 Other Techniques 15
2.4.3 Applications of Global Minimum Finding 18
2.5 Non-parametrized Methods 19
2.5.1 Quantum Mechanics: Introduction 19
2.5.2 Hartree-Fock Approximation 21
2.5.3 Møller-Plesset Perturbation Theory 25
2.5.4 Coupled-Cluster Theory 30
2.5.5 Further Developments 32
2.6 Parametrized Methods 34

2.6.1 Force Fields . . . 34
2.6.2 Semiempirical Methods . . . 36
2.6.3 Density Functional Theory . . . 38
2.7 Programming Techniques . . . 41
2.7.1 Choosing a Programming Language . . . 41
2.7.2 Object-Oriented Programming . . . 43
2.7.3 Parallelization . . . 44
2.8 State of the Art . . . 45
2.8.1 Programming Techniques . . . 45
2.8.2 Cluster Structure Optimization . . . 47

3 The OGOLEM Framework 51
3.1 Scope of the Project . . . 52
3.2 Own Contribution . . . 52
3.3 Publication . . . 53
3.4 Additional Information . . . 54
3.4.1 Collision Detection and Dissociation Detection . . . 54
3.4.2 Object-Oriented Design Concepts . . . 55

4 Benchmarking the Framework 59
4.1 Scope of the Project . . . 60
4.2 Own Contribution . . . 60
4.3 Publication . . . 61

5 Highly Mixed LJ Clusters 83
5.1 Scope of the Project . . . 84
5.2 Own Contribution . . . 84
5.3 Publication . . . 85
5.4 Additional Information . . . 85
5.4.1 Notes on the Parameter Fit . . . 85

6 Kanamycin A Dimers 89
6.1 Scope of the Project . . . 90
6.2 Own Contribution . . . 90
6.3 Publication . . . 91

7 Design of Switchable Molecules **111**
7.1 Scope of the Project 112
7.2 Own Contribution 112
7.3 Publication 113
7.4 Additional Information 113
7.4.1 Outlook 113

8 Summary **115**

Acknowledgments **119**

References **121**

Declaration **133**

Curriculum Vitae **135**

CHAPTER

1

Introduction

> What I cannot create, I do not understand.
>
> Richard P. Feynman

Optimization problems play an important role in theoretical chemistry. Examples are the optimization of molecular and geometrical structures, design of molecules suiting a certain purpose and fitting of any set of parameters in general. In principle, all optimization problems can be divided into two groups: those where just one solution exists and those with multiple solutions of different quality. While the former group can be tackled relatively easily, for the latter classical analytical solutions are not feasible since any found solution can still be non-optimal.

Knowing about structures on the molecular level plays a key-role in chemical understanding. Already the alchemists in the old days tried to visualize the molecular level, with rather raw models though. As experiments and theories evolved, the view on structures on the molecular level became more and more refined.[1] The importance of the correct

[1] Actually, nowadays certain fields of chemistry dealing with biological compounds are back in a rather

geometry of a chemical compound is highlighted in the context of the lock-and-key and induced-fit principle in biochemical systems [2, 3]. Only molecules of the correct geometric and chemical shape can access for example a binding pocket in an enzyme and react. This connection needs to be obeyed in drug design, requiring knowledge on the geometric structure of the drug molecules [4].

Unfortunately, the problem of predicting the correct structure[2] proves to be difficult in various respects. In general, the problem is considered to be of *NP-hard* nature [5, 6], making it unsolvable within polynomial time. Employing heuristic algorithms, this problem remains difficult, with the difficulty of solving it with said algorithms increasing at least $\mathcal{O}(N^3)$ with the problem dimensionality[3]. This ultimately requires a lot of energy evaluations, easily exceeding multiple thousands already for relatively small systems. Additionally, very exact methods are required to correctly model smallest energy differences between candidate structures. It is obvious that the so-called *chemical intuition* is doomed to fail in the context of any non-trivial structure or cluster, requiring an unbiased, automatic search for the correct structure. Such a search should ideally be both independent of the actual cluster composition – which building blocks are present in which amounts – but also from the wanted level of theory.

The challenges of the global optimization of structures lies in requiring a high accuracy in and a high number of energy/gradient evaluations. Due to the high number of required evaluations, highly accurate methods are not feasible since their footprint is simply too big. This problem can partly be circumvented by combining the speed of lower-level methods with the accuracy of higher-level ones. By reparametrizing the lower-level methods to highly accurate ones, one achieves – within the fitting regime – high accuracy at the cost of a small computational effort. Obviously, such an approach is perfect for the global optimization of structure and even more elegant if the parameter fitting is carried out in a globally optimizing way.

As mentioned before, especially in drug design structure can be directly translated to effect. This holds true also for other problems, like more generalized molecular design.

symbolic description of structures.

[2] In the context of this work, it will be assumed that the lowest energy structure is the correct one. It should be noted though that due to thermodynamical effects, the *natural* structure is a blend of various low energetic structures.

[3] Expressing this in numbers, it translates to a factor of 1000 in difficulty if the dimensionality increases by a factor of 10.

By modifying the structure of a molecule, the electronic structure is modified as well. This ultimately modifies measurable properties. Again, for non-trivial cases this is not accessible through chemical intuition. Modeling a molecule for a specific purpose shows just the same requirements and difficulties as pointed out above for the global optimization of structures, the only difference being that molecular design is a discrete problem while the above described problems are non-discrete ones.

Unifying the solution of the problems described above is the target of this work. By developing a new framework for global optimization and using it to solve problems of molecular design – the design of optimally switchable molecules –, global parameter fitting and the global optimization of structures, both on standard levels of theory as well as employing reparametrized methods, a generalized approach towards these problems is taken. The project as such acts upon two maxims. The first is a novel approach towards the implementation of genetic algorithms for chemical applications, the second is to assist experiment in regions and tasks that are not (yet) accessible to it.

This thesis will first give an introduction to techniques and methods employed in this work. All following publications feature a short introduction trying to put it into the overall context. If necessary, additional informations are provided that might be of interest for the reader but were too verbose to be included into the corresponding publication. A short summary and outlook are then provided.

CHAPTER

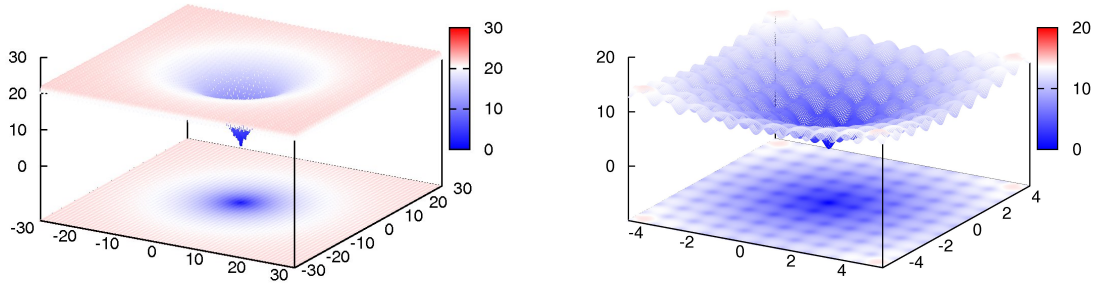
2

Techniques and Methods

> God does not care about our mathematical difficulties. He integrates empirically.
>
> Albert Einstein

This work is not suitable to and will not make any attempts to replace any textbooks on either global optimization techniques or methods to evaluate energies of chemical systems.

Its sole aim is to give a simplified – therefore in parts probably not absolutely accurate – description of the used methods and techniques in this project. If a more detailed depiction is wanted or needed, either the quoted literature or textbooks are to be consulted.



(a) total view

(b) fine structure

Figure 2.1: An example for a hypersurface: Ackley's function in 2D.

2.1 Hypersurfaces: Introduction

Hypersurfaces are the generalized version of ordinary surfaces in three-dimensional space to the case of an n-dimensional space. The dimensionality of the hypersurface is always one less than that of the ambient space [7]. An example for an analytical hypersurface is depicted in fig. 2.1, Ackley's function [8] in two dimensions, since obviously any higher dimensionality can unfortunately not be depicted.

Hypersurfaces are allowed to show minima, maxima, saddle points and singularities, making them the general representation of non-discrete systems in this work. Obviously, hypersurfaces can possess both multiple minima and maxima, showing the difference between local minima and the global minimum[1].

Ideally, a minimum search on the hypersurface of the system should be independent of how the surface was obtained. Translating this to the problems mentioned within this work, this means through which methods the fitness of a system was obtained should not have any impact on the way the optimization is carried out. These two should be encapsulated from each other.

[1]Cases where multiple minima are exactly of same function value are extremely rare with real-world systems.

2.2 Minimum Finding: Introduction

General minimum finding problems can be divided into two subclasses. One class is discrete problems where the parameters (or part of them) can just take certain well-defined values, the other being non-discrete problems, where all parameters can take real values. In the case of non-discrete problems, allowed intervals are possible, e.g, since the function is not defined outside them.

In the context of this work, both problem classes have been investigated. The non-discrete class is represented by parameter and cluster optimizations, the discrete class by molecular design.

The actual minimum finding can be either of local or global nature, differing only in the quality of the possible solution. While a local optimization will only yield the optimal solution either by a fluke or when there is just one solution, a global optimization is targeted at finding *the* optimal solution.

Various techniques exist targeted at either local or global optimization of either discrete or non-discrete problems. In the following subsections, the methods used within this work will be explained briefly and compared to other important ones.

2.3 Minimum Finding: Local Techniques

Local techniques to get to a minimum of a multidimensional function can be divided into two subgroups. The first subgroup consists of methods that depend on gradient information, in the second one gradients are not necessary.

Both can obviously only be applied to non-discrete systems. For those systems, they provide powerful and efficient means of reaching a local minimum once inside of a funnel.

2.3.1 Methods using Derivatives

If a functional expression is easy enough to be differentiated analytically, a gradient can be calculated roughly as fast as a single function evaluation. The gradient provides effective means of locating a local minimum. In contrast to the function value is is a vector of length N, providing a guaranteed downhill direction.

A very important class of gradient-based optimization algorithms is the class of *quasi-Newton* methods, which have been used within this project. The basic idea that all members of this class share is to iteratively build up a good approximation to the inverse Hessian $\mathbf{A^{-1}}$, defined as

$$\lim_{i \to \infty} \mathbf{H}_i = \mathbf{A^{-1}}. \tag{2.1}$$

The availability of the inverse Hessian allows for a step directly into the minimum of a local quadratic approximation.

Obviously, it would be better to achieve the limit after N iterations, not after an infinite number of them. In this approximative Hessian lies also the difference why these methods are named *quasi-Newton*. In the classical Newton algorithm to search for a minimum, the exact Hessian is computed while in quasi-Newton methods only the current approximation to it is used.

This approximation in comparison to the classical method is not, as one might easily assume, harmful in terms of the convergence pattern of the algorithm. Actually, the convergence is at least staying constant, since the use of an approximated Hessian provides a better convergence far off a minimum. Taking a classical Newton step might well lead to an increased function value in these situations, while the quasi-Newton methods guarantee the step to move in a downhill direction. In these regions, the exact Hessian might not be positive-definite, leading to a Newton step into a region of function increase. The quasi-Newton method in contrast starts from a positive-definite, symmetric inverse Hessian and builds the series of approximative Hessians $\mathbf{H}_i$ conserving these properties.

When being close to a minimum, the update of the approximative Hessian approaches the exact Hessian, making these methods quadratically convergent. The difficulty lies in the actual design of the updating formula. As an example, the updating formula of

the *BFGS* (Broyden-Fletcher-Goldfarb-Shanno) method will be given [2].

Starting from an initial guess for the approximate Hessian, normally the unit matrix, the update is defined by the BFGS method as

$$\begin{aligned}\mathbf{H}_{i+1} = \mathbf{H}_i &+ \frac{(\mathbf{x}_{i+1} - \mathbf{x}_i) \otimes (\mathbf{x}_{i+1} - \mathbf{x}_i)}{(\mathbf{x}_{i+1} - \mathbf{x}_i) \cdot (\bigtriangledown f_{i+1} - \bigtriangledown f_i)} \\ &- \frac{[\mathbf{H}_i \cdot (\bigtriangledown f_{i+1} - \bigtriangledown f_i)] \otimes [\mathbf{H}_i \cdot (\bigtriangledown f_{i+1} - \bigtriangledown f_i)]}{(\bigtriangledown f_{i+1} - \bigtriangledown f_i) \cdot \mathbf{H}_i \cdot (\bigtriangledown f_{i+1} - \bigtriangledown f_i)} \\ &+ [(\bigtriangledown f_{i+1} - \bigtriangledown f_i) \cdot \mathbf{H}_i \cdot (\bigtriangledown f_{i+1} - \bigtriangledown f_i)]\mathbf{u} \otimes \mathbf{u}\end{aligned} \tag{2.2}$$

where $\mathbf{u}$ is defined as

$$\mathbf{u} = \frac{(\mathbf{x}_{i+1} - \mathbf{x}_i)}{(\mathbf{x}_{i+1} - \mathbf{x}_i) \cdot (\bigtriangledown f_{i+1} - \bigtriangledown f_i)} - \frac{\mathbf{H}_i \cdot (\bigtriangledown f_{i+1} - \bigtriangledown f_i)}{(\bigtriangledown f_{i+1} - \bigtriangledown f_i) \cdot \mathbf{H}_i \cdot (\bigtriangledown f_{i+1} - \bigtriangledown f_i)} \tag{2.3}$$

and $\bigtriangledown f_i$, $\bigtriangledown f_{i+1}$ as the gradients at position i and $i+1$, respectively. $\mathbf{x}_i$, $\mathbf{x}_{i+1}$ denote the solution vectors at i and $i+1$.

Without getting into the details of this update formula, a general tendendy for numerical methods can be observered. The BFGS algorithm is robust and fast converging. At the same time, the update formula is not simple. This trade-off is typical for more advanced numerical methods like BFGS and normally is accepted for the sake of algorithmic advantage.

Limits of the described BFGS method arise from the increasing memory requirement, which scales $\mathcal{O}(N^2)$ with the problem dimensionality due to the fully stored approximative Hessian, and the availability of a gradient for the to be optimized functional expression. The solution for the first problem is to use a linear scaling BFGS, LBFGS [15]. The latter can be solved either by using numerical gradients, which is only feasible for low-dimensional problems, or by employing gradient-free minimization procedures discussed in the following section.

[2] No explicit derivation of the formula will be provided, this can be found in the series of original papers [9–14].

2.3.2 Methods without Derivatives

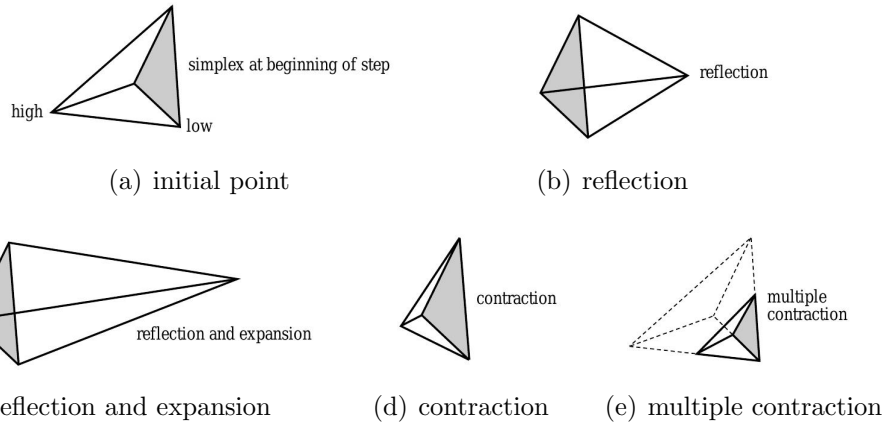
Methods that do not require derivatives come into play if an analytical gradient is not available for a given problem, e.g. because the function is too difficult to be differentiated. Since the calculation of a numerical gradient requires at least a two-point stencil, the number of function evaluations needed for building a numerical gradient scales at least $\mathcal{O}(N)$ with the dimensionality N of the function in comparison to $\mathcal{O}(1)$ when analytical gradients are available. This makes methods attractive that do not require the calculation of a gradient and still yield a local minimum within a reasonable amount of function evaluations.

Various different algorithms exist [15, 16] which partly are optimized for specific function patterns, e.g. smooth or non-smooth functions. As a general approach towards the problem of function minimization without derivatives, the *downhill simplex* method will be described since it is an elegant and easily visualizable algorithm.

A *simplex* is the geometrical figure consisting of $N + 1$ vertices in N dimensional space. Therefore, the starting guess requires $N + 1$ points, defining an initial simplex. The algorithm then iterates towards a minimum, carrying out a series of steps, consisting of some elementary patterns, depicted in fig. 2.2. Most of these elementary steps are so-called *reflection* steps, only moving the point where the function is largest through the opposite face of the simplex to reach a lower point. These steps maintain the volume of the simplex as such. The volume is enlarged if possible to take a larger step, yielding a combined *reflection and expansion* step. Once the algorithm reaches the floor of a basin, a *contraction* step is carried out, contracting the simplex in the transverse direction to reach the minimum of the basin. In closest vicinity to the minimum, the simplex is multiply contracted, yielding the solution vector of this minimum.

Termination occurs once a criterion is fulfilled, e.g. the step length being below a tolerance criterion. This approach is error prone in some multidimensional cases, possibly yielding a solution which is not a minimum but which may converge to an actual minimum if restarted. This may e.g. be the case in a flat but not minimal area of the hypersurface. A restart in this case initializes the N of the $N+1$ vertices of the simplex again – increasing its volume –, causing a better progress.

Although the downhill simplex method has a poor convergence pattern and might require



(a) initial point (b) reflection

(c) reflection and expansion (d) contraction (e) multiple contraction

Figure 2.2: Graphical representation of the elementary steps of the downhill simplex algorithm for the case of a tetrahedron. Figure taken from ref. [15].

restarts to converge correctly, it is very efficient in terms of development effort. Even the reference implementation of *Numerical Recipes* in C++ is less than 100 SLOC long[3] [15]. The Java implementation in ref. [17] is also less than 100 SLOC long, but with the advantage of a full object-oriented design and exception handling.

In the context of this project, the more advanced *principal axis* method from Brent has been used through a Java translation of the NUMAL package for ALGOL60 procedures of numerical mathematics [18, 19]. The principal axis method converges quadratically in terms of steps but still requires a high number of function evaluations for a line search.

Summarizing, the local optimization algorithms that do not require a gradient should only be used if an analytical gradient is too tedious to obtain and/or compute.

2.4 Minimum Finding: Global Techniques

The global techniques presented here share the common pattern that they are applicable in a universal manner. With more knowledge about a specific system, it is possible to either tune the techniques or to develop system-specific algorithms providing superior

[3]SLOC is the abbreviation of Single Line Of Code. It is used as a measure to approximate development effort. The code length of different programming languages might not be easily comparable, e.g. Java is approximately as verbose as C++ which is less verbose than FORTRAN.

performance.

The techniques presented here follow a *stochastic heuristic* approach. This is motivated by some problems of global optimization in general being considered to be of *NP-hard* nature[4]. Whether or not a problem required the here presented techniques is not necessarily known *a priori*. As a rule of thumb, any non-trivial, multidimensional problem normally can efficiently solved with the here presented techniques. No analytical solution is feasible for these cases, since the number of minima scales at least exponentially with the dimensionality of the problem, making a total enumeration of all minima impossible.

2.4.1 Genetic Algorithms

Genetic algorithms are inspired by nature. By representing important steps of genetics as algorithms, this method asymptotically locates the global minimum. Various different occurrences exist, all sharing the same primitives stemming from the ones described in detail in ref. [1], where a primitive is a single and elementary operation.

First some definitions are required. An *individual* is a possible solution to the optimization problem. A *genome* refers to the sum of all genes of one individual. It can be either binary or real-number encoded, depending on the exact implementation of the later mentioned genetic operators, either of the two representations can be more advantegous. The *fitness* is a measure for how optimal the individual is as a solution[5]. By a simple modification of the fitness function, a minimization can be turned into a fitness maximization and vice versa. They *genetic pool* is a snapshot of the fittest individuals at a certain point in the optimization run.

The general idea is to create offspring from parents. By carrying this out multiple and successive times, genetic progress is created. This genetic progress is optimal due to the inherent communication during the optimization process. This guarantees the solutions to improve in an iterative way during the optimization run. Since one works with a

[4]Proofs of the NP-hard nature exist for the structural optimization of clusters [5, 6] but their correctness is disputed [20].

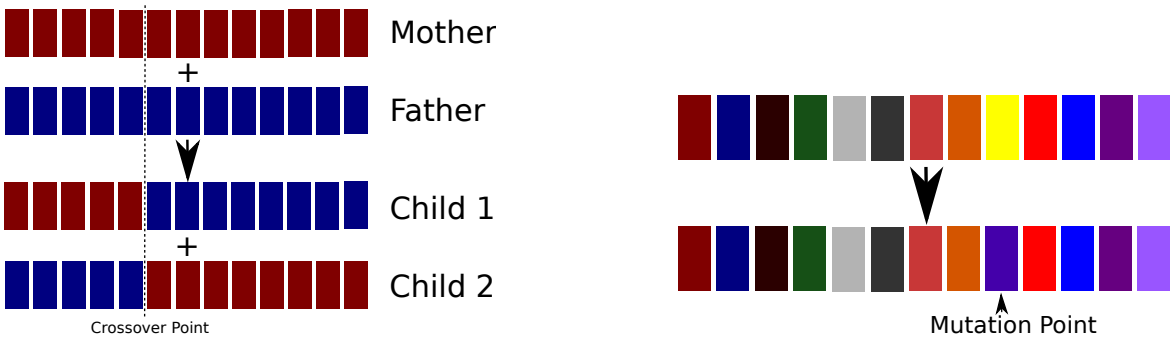
[5]It should be noted that cases exist where the optimal fitness is not known in advance, e.g. the global optimization of cluster structures.

set of individuals, one also obtains a set of multiple solution, among them the global optimum.

The first primitive is *mating*. Mating describes how two parent individuals are chosen to create offspring. While this might sound like a relatively trivial task, it is of crucial importance as a first step to ensuring optimal progress of the genetic pool. Two individuals that are very much alike will likely show little genetic progress, which is only good if both are in close vicinity to the global minimum. At the same time, just mating two individuals because they show the biggest genetic difference will likely cause a rather big step in the search space. The approach taken in this work, similar to other work [21, 22], is to choose the first individual, the *father*, ranked by fitness. The second individual is chosen randomly, to ensure optimal genetic progress and diversity.

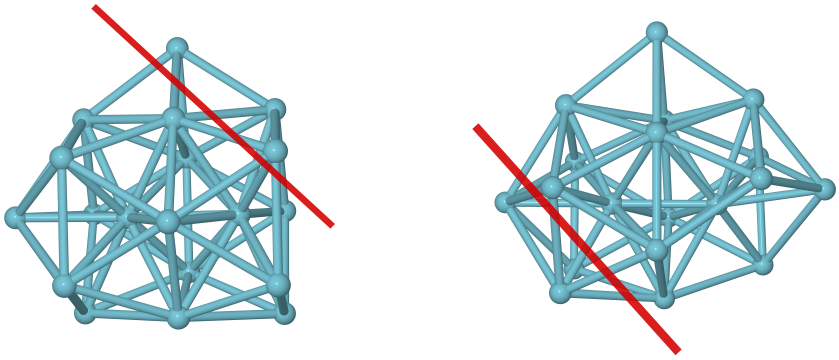
The next primitive is *crossing*. Crossing is, equivalently to its biological definition, defined as the mixture of the two parent genomes. A graphical representation is depicted in fig. 2.3. One or more randomized crossover points are picked. At these points, genoms are cut and interchanged, yielding two children individuals as in the graphical representation.

Systematic differences arise in through what representation the crossover cut(s) are carried out. If the genome is used as a string of values, either binary or real-number encoded, the term *genotype* crossover is used and the cut is carried out as depicted in fig. 2.3. If the cut goes through a structural representation in the case of the global optimization of clusters, as depicted in fig. 2.4, a *phenotype* crossover operator is used [21, 23, 24]. This is motivated by the global optimization of clusters being an optimization of a 3D



(a) Crossover (b) Mutation

Figure 2.3: Genetic operation primitives represented graphically.



(a) father (b) mother

Figure 2.4: Phenotype crossover. Plane depicting the actual cut.

arrangement. A phenotype cut preserves the local neighborhood structure outside of the immediate vicinity of the cutting plane in difference to a genotype cut through a 1D representation where a direct mapping from 1D to 3D might not be trivial. While this might sound rather specific to the problem of the global optimization of cluster structures, it is a good example on how more accurate knowledge about a system can lead to changes in the exact algorithmic implementation. Genetic algorithms are flexible enough to allow for such changes.

The *mutation* primitive is also depicted graphically in fig. 2.3. With a certain mutation probability, one or more genes of a single genome are changed starting at a random point. The concept of *directed mutation* [21, 25–27] allows for a better short-range exploration. Being based on a phenotype representation of the genom, it allows for small modifications of a good solution in difference to rather big, randomized jumps within the search space caused by traditional mutation.

These primitives together form a global optimization task. How these tasks are combined and executed, depends on the scheme used. In case of a traditional *generation-based* scheme, as explained in detail in ref. [1], some of those tasks, the number being the *generation size*, are executed in parallel, results are gathered and the next generation is then started. In the more flexible *pool* scheme, as described in ref. [22] and used within this work. In this scheme, the concept of a genetic generation is completely removed in favor of a *genetic population*. This fixed-size population is constantly updated to account for genetic change, allowing for an execution of global optimization tasks without any serial gathering of results.

Difficulties arise for genetic algorithms in the short-range exploitation of the search space. This feature is of importance when the genetic population is already rather close to the global minimum. For this, hybrid schemes have been developed, where genetic algorithms are paired with a local optimization of the children, allowing for an efficient exploitation of the nearest minimum on the hypersurface. Obviously, this extension can only be used in the case of non-discrete problems.

Further developments include, but are not limited to[6], extensions to reduce the search space size and to reduce the number of, sometimes costly, fitness evaluations. The first is for example represented by so-called *cultural algorithms* [31]. The basis of these algorithms is to use experience gained at runtime to direct the genetic progress. Costly fitness evaluations, occurring with methods mentioned later, can be omitted using *taboo-search*. Before a fitness evaluation is carried out, the individual is checked against taboos. These taboos are, in the easiest case, just a collection of all known individuals. Only if the individual has not been encountered before the fitness is evaluated.

Advantages of genetic algorithms are their easy yet elegant theory and possible implementation, a very powerful long-range exploration and the possiblity of a system-specific implementation or extension of core primitives. Problems arise from their rather limited short-range exploration. This can partly be circumvented by using local optimization methods to exploit a given funnel or directed mutation to explore the immediate vicinity of a funnel.

Genetic algorithms are a powerful and flexible tool for the global optimization of multidimensional multiminima problems. Through extensions, the performance can be improved and the GA can be adapted for the problem under study.

2.4.2 Other Techniques

Various other techniques exist and even more are under development or will be developed in the future. Therefore, this can just be a mere summary of some important techniques and a glance at recent developments.

[6] Algorithms were a dynamic or static grid is used are not considered within this work but exist as another extension of GAs [28–30].

Besides genetic algorithms, evolutionary strategies [32, 33], genetic programming [34] and evolutionary programming [35] are classified to be evolutionary algorithms. Differences between different algorithms of this class can vary between subtle and major. Therefore, every algorithm should be analyzed individually. For example, hybrid metaheuristics like GAs with the previously mentioned taboo search or with so-called *kangaroo*-extensions [36][7] have been successfully applied. Additionally, evolutionary algorithms have been made aware of quantum logic, employing qubit patterns [37], enlarging the variety of available methods within this class even more.

Probably the most important techniques besides evolutionary algorithms for the global optimization of chemical structures are the ones inspired by molecular dynamics. Namely, these are *simulated annealing* [38] and *basin hopping* [39] as well as the more recently developed *minimum hopping* [40]. The disscusion here will be restricted to the first method, since it is generally applied and very educative. Additionally, an algorithmic overlap with the previously described downhill simplex method can be constructed.

The classical simulated annealing (SA) technique has been inspired by the thermodynamical crystallizing of liquids or annealing of metals. In a melted metal, normally at rather high temperatures, the atoms move strongly. When cooling the melt down, thermal energy and therefore mobility is removed from the system. If this happens slowly enough, the atoms order themselves and, in case of a pure melt, form a perfect crystal structure. This structure is the global minimum of the system. If the cooling is too rapid, so-called *quenching*, no perfect crystal structure is obtained. SA employs this idea by means of an algorithmic translation. The idea of finding the global optimum only through a sufficiently slow cooling and a sufficiently long relaxation time is seductive in its elegance. Unfortunately, it suffers from a fundamental uncertainty. One does not know *a priori* what sufficiently long or slow is. Getting back to the algorithmic implementation, the application of SA to an optimization problem requires [15]

1. a description of possible system configurations,

2. a generator of (possibly random) changes in the configurations,

3. a to be optimized observable, resembling the energy in the crystal structure anal-

[7] The name is a well-chosen metaphor for the behavior of the algorithm. It is designed to jump out of a local minimum trap.

ogy,

4. a control parameter resembling the temperature and

5. a protocol how the control parameter is lowered, resembling the cooling protocol.

This can be, as pointed out in ref. [15], again be implemented for non-discrete problems employing the already known simplex pattern. By substituting the solution vector with a simplex, one can again employ the same moves as described in sec. 2.3.2. The control parameter can then be used to modify the simplex vertices in a randomized manner. The advantage of this approach is the build-in local optimization scheme, since in the limit of a vanishing control parameter, the algorithm is a standard downhill simplex method as described in detail in sec. 2.3.2. Although also other SA implementations converge into the nearest minimum once the control paramter is set to zero, the simplex local optimization is a useful scheme.

For all the mentioned MD-inspired algorithms, a performance comparison with EAs shows different strengths and weaknesses. While the short-range exploitation of MD-based algorithms is generally considered to be better than the one of pure EAs, EAs generally provide a better long-range exploration. Employing hybrid EAs with enabled local optimization steps in general makes the EAs be at least on par with MD-based methods[8]

The described simulated annealing algorithm can also be used for more general global optimization problems. Besides this, it can be extended just like the previously described GAs to adapt better to specific situations by employing specifically designed annealing protocols.

Additionally, swarm intelligence techniques should be mentioned as a popular contemporary global optimization technique [42, 43]. The swarm optimization techniques are, just like GAs, inspired by nature but focus on modelling the movements of individuals in a group, e.g., a single fish in a swarm instead of the biological evolution of the fish species. Particle swarm optimization (PSO) has been applied to the problem of the structure op-

[8]By modifying benchmarking conditions, one can make one algorithm superior to the other, as done e.g. in ref. [41]. Ref. [41] implements a phenotype crossover but fails to find nonicosahedral global minima for cluster sizes greater than LJ_{38}. Those have been found already in the 1990ies (e.g. in ref. [21]) employing such techniques.

timization of clusters [44]. Being a relatively new technique, its applicatibility has only been shown for small systems. For these, PSO has, like GAs, the advantage of being inherently parallelizable and, in contrast to GAs, seems to require a smaller generation size.

2.4.3 Applications of Global Minimum Finding

Applications of global minimum finding are numerous. They include, but are for sure not limited to, scheduling problems [45–47], logistics problems [48–50], chip design [51, 52], and materials design [53]. In general, global optimization techniques come into play, as pointed out before, where *NP-hard* problems are suspected. It should be noted that although the ultimate target of any global optimization of course is the global optimum, it might be acceptable for certain applications to find a very good minimum, close to the global one in fitness. This for example might be the case in time-critical applications like the previously mentioned logistics and scheduler problems, where the footprint to find the global optimum simply cannot be afforded. This problem is generalized in so-called limited budget situations. For these, specialized algorithms or approximative heuristics might be more suitable that possess a particular strength in finding very fast a very good minimum.

Getting more into scientific applications within chemistry and physics, structural optimization [54–57], modelling of analytical potentials [58], molecular and drug design [56, 59, 60], crystal structure prediction [56, 61] and structure determination from diffraction data [56, 62–64] are examples for the application of global optimization techniques.

Summarizing, global optimization techniques play a big role in scientific as well as real-world applications. Any optimization problem where the number of local minima is assumed to be too high for a complete enumeration can be explored using for example evolutionary algorithms. Assuming a general problem, the only adaption required when using EAs is a genome representation of the problem, the definition of a fitness function suitable for the problem and, if more advanced genetic operators are wanted or needed, an adaption of the crossover and/or the mutation operator. The definition of the fitness function for chemical systems ultimately requires energy evaluation techniques. An exemplary selection of these is presented in the upcoming sections.

2.5 Non-parametrized Methods

Non-parametrized, or *ab initio*, methods play a key role as reference techniques when experimental data is either sparse or unreliable. Also, their results can, if the system is fitting both from size and kind, be of course directly used for reliable modeling. Their strength is the possibility to refine results in a systematic manner, in contrast to the rather unsystematic behavior of the later discussed parametrized methods. The methods most important for this work are discussed in the following.

2.5.1 Quantum Mechanics: Introduction

In Quantum Mechanics (QM), any system is described through its wavefunction Ψ. In principle the wavefunction is time-dependent but can be separated for most systems and applications described in this work into a time-independent and time-dependent part

$$\Psi(\mathbf{x}, t) = \psi(\mathbf{x})\vartheta(t) \tag{2.4}$$

with $\mathbf{x}$ being a vector of the generalized coordinates (spatial and spin coordinates) of the system. The system can now be described by means of the time-independent Schrödinger equation (SE)

$$\hat{H}\psi = E\psi. \tag{2.5}$$

The SE defines which stationary states ψ are allowed in the system described through the Hamiltonian $\hat{H}$. The wavefunction needs to be an eigenfunction of the operator and the energy E the eigenvalue. The Hamiltonian for a system with N electrons and M cores is, in atomic units[9]:

$$\hat{H} = -\frac{1}{2}\sum_{i=1}^{N}\Delta_i - \frac{1}{2}\sum_{m=1}^{M}\frac{1}{m_m}\Delta_m - \sum_{m=1}^{M}\sum_{i=1}^{N}\frac{Z_m}{r_{im}} + \sum_{i<j}\frac{1}{r_{ij}} + \sum_{m<n}\frac{Z_m Z_n}{r_{mn}}. \tag{2.6}$$

with i and j being the electron indices, m and n the atomic core indices, Z_m the atomic number and the mass of the atomic core m_m. r_{jn} is the distances between two particles,

[9] Atomic units are designed to ease a lot of expressions in quantum mechanics. By defining the mass of an electron m_e, the charge of a proton e, the atomic unit of action $\hbar$ and the permittivity $4\pi\epsilon_0$ to be fundamental units, the derived units of energy E_h (hartree) and a_0 (bohr) are obtained.

in this case between electron j and core n.

Therefore, the first two terms are describing the kinetic energy of electrons and atomic cores, while the last terms are the Coulomb energies of the particle interaction. Obviously, one can split the Hamiltonian into a kinetic and potential part

$$\hat{H} = \hat{T} + \hat{V}. \tag{2.7}$$

In the potential part $\hat{V}$ one could also add external potentials, e.g., electromagnetic fields[10].

Due to the complexity, an exact, analytical solution of this equation is only possible for the easiest cases up to the hydrogen atom. For more complex system, approximations are necessary. This can be either an approximation to the wavefunction[11] and/or to the Hamiltonian.

A common approximation is the one introduced by Born and Oppenheimer[12] [66], which is applicable in most cases. The atomic cores are many orders of magnitude heavier than the electrons. This causes the electrons to move much faster than the cores, allowing for a decoupling of these two classes of movement. The preceding separation is the Born-Oppenheimer separation which already includes the concept of electronic potential energy surfaces (PESs) but has them coupled and is therefore exact. The BO approximation is also called *adiabatic* approximation and holds true for most systems within a certain degree of accuracy. Exceptions to be named are e.g. photo reactions where conical intersections occur.

An electronic Hamiltonian $\hat{H}_{el}$ can then be obtained

$$\begin{aligned} \hat{H}_{el} &= -\frac{1}{2}\sum_{i=1}^{N}\Delta_i - \sum_{m=1}^{M}\sum_{i=1}^{N}\frac{Z_m}{r_{im}} + \sum_{i<j}^{N}\frac{1}{r_{ij}} \\ &= \sum_{i=1}^{N}\hat{h}(i) + \sum_{i<j}^{N}\frac{1}{r_{ij}}. \end{aligned} \tag{2.8}$$

[10] The most easy way to include a continuum-like solvation around a molecule or cluster of molecules.
[11] The Pauli principle must be obeyed in any case. It requires the wavefunction of fermions to be antisymmetric concerning the exchange of particles.
[12] A concise description on this can be found in ref. [65].

The partial solution for the core terms is added *a posteriori* after solving the electronic Schrödinger equation.

In the following sections, methods to solve the electronic Schrödinger equation in an approximative manner are presented, discussing both their accuracy and relevance.

2.5.2 Hartree-Fock Approximation

The Hartree-Fock (HF) approximation is probably the most basic approximation in *ab initio* methods. It is a single determinant method, meaning that the wavefunction is composed of a single Slater determinant of the form

$$\psi^{HF} = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_1(\mathbf{x}_1) & \psi_2(\mathbf{x}_1) & \dots & \psi_N(\mathbf{x}_1) \\ \psi_1(\mathbf{x}_2) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \psi_1(\mathbf{x}_N) & \dots & \dots & \psi_N(\mathbf{x}_N) \end{vmatrix} = |\psi_1(\mathbf{x}_2)\psi_2(\mathbf{x}_2)\cdots\psi_N(\mathbf{x}_N)\rangle\,. \tag{2.9}$$

It is important to note that the electronic coordinates $\mathbf{x}_i$ are a vector of both spatial and spin coordinates

$$\mathbf{x}_i = \{\mathbf{r}_i, \mathbf{s}_i\}\,. \tag{2.10}$$

Following the Pauli principle, the final wavefunction is the antisymmetrized product of molecular spin orbitals $\{\psi_i\}$. We will, for the sake of simplicity, assume restricted closed-shell conditions, meaning that the number of electrons is even and that all electrons are paired in molecular orbitals.

The Slater-Condon rules [67] define the expectation value of the energy to be

$$\left\langle \psi^{HF} \middle| \hat{H} \middle| \psi^{HF} \right\rangle = \sum_{i}^{N/2} 2 \left\langle i \middle| \hat{h} \middle| i \right\rangle + \sum_{ij}^{N/2} \left[2(ii|jj) - (ij|ji)\right]. \tag{2.11}$$

with

$$\left\langle i \left| \hat{h} \right| j \right\rangle = \int \phi_i^*(\mathbf{r}_1) \hat{h} \phi_j(\mathbf{r}_1) d\mathbf{r}_1 \tag{2.12}$$

$$(ij|kl) = \int \phi_i^*(\mathbf{r}_1) \phi_j(\mathbf{r}_1) r_{12}^{-1} \phi_k^*(\mathbf{r}_2) \phi_l(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \tag{2.13}$$

in Mulliken notation[13]. Additionally, the integration over spin coordinates has been assumed, making the theory spin-free.

Since the HF approximation is a variational concept, the HF energy is an upper boundary to the exact energy. This relation can be used to optimize the wavefunction, since any lower expectation value of the energy corresponds to a better description of the system as such. An optimized HF wavefunction therefore needs to be an extremum[14], turning the derivative of eq. 2.11 in respect to the change of orbitals to zero, conserving the orthonormality condition. For this, the *Lagrangian function*

$$\mathcal{L} = \left\langle \psi^{HF} \left| \hat{H} \right| \psi^{HF} \right\rangle - 2 \sum_{ij}^{N/2} \epsilon_{ij} \left[\langle i|j \rangle - \delta_{ij} \right] \tag{2.14}$$

is minimized, with the Lagrangian multipliers ϵ_{ij} and $\langle i|j \rangle = S_{ij}$ as the overlap integral of orbitals i and j.

Setting the derivative of the Lagrangian for every orbital to zero yields the HF equations

$$\hat{f} \left| i \right\rangle = \sum_{j}^{N/2} \epsilon_{ij} \left| j \right\rangle \tag{2.15}$$

where the *Fock operator* $\hat{f}$ is defined as

$$\hat{f}(i) = \hat{h}(i) + \sum_j \left[2\hat{J}_j(i) - \hat{K}_j(i) \right] = \hat{h}(i) + \hat{g}(i). \tag{2.16}$$

[13] The straightforward bra/ket-notation is assumed throughout the text. It is introduced in all textbooks on the subject of wavefunction based methods, e.g. in ref. [68].

[14] Actually, a minimum is of course targeted.

The Coulomb operator $\hat{J}_j(i)$ and exchange operator $\hat{K}_j(i)$ are given as

$$\hat{J}_j(i)\phi_i(\mathbf{r}_1) = \int \phi_j^*(\mathbf{r}_2)\frac{1}{r_{12}}\phi_j(\mathbf{r}_2)\phi_i(\mathbf{r}_1)d\mathbf{r}_2 \tag{2.17}$$

$$\hat{K}_j(i)\phi_i(\mathbf{r}_1) = \int \phi_j^*(\mathbf{r}_2)\frac{1}{r_{12}}\phi_j(\mathbf{r}_1)\phi_i(\mathbf{r}_2)d\mathbf{r}_2 \tag{2.18}$$

and named according to their property, describing the Coulomb interaction between two electrons and exchaning them, respectively.

The approximation of HF theory is obvious when comparing the exact electronic Hamiltonian in eq. 2.8 with the Fock operator in eq. 2.16: All interelectronic interactions are substituted with an averaged field, defined by $\hat{g}(i)$.

The *canonical* HF equations are obtained using a unitary transformation into a basis where the Fock operator is diagonal

$$\langle i|\hat{f}(1)|j\rangle = F_{ij} = \delta_{ij}\epsilon_i \tag{2.19}$$

This particular basis is the canonical orbital basis. The diagonal elements of the Fock matrix F_{ij} are the corresponding orbital energies ϵ_i. It is important to note that the total energy is not equal to the sum over all occupied orbitals.

In principle, by using a complete basis set (CBS), one could obtain the 'exact' solution in the context of the HF approximation. Obviously, this is not feasible in praxis. Therefore, one uses a finite basis set, nowadays mainly constructed from atom centered Gaussians, to obtain the orbitals approximated as a linear combination of atomic orbitals (LCAO)

$$\phi_i(\mathbf{r}) = \sum_\mu C_{\mu i}\chi_\mu(\mathbf{r}). \tag{2.20}$$

Using this approach, the HF energy is yielded in dependence of the atomic orbital inte-

grals as

$$\begin{aligned} E_{HF} &= \sum_{\mu\nu} D_{\mu\nu} \left\{ h_{\mu\nu} + \frac{1}{2} \sum_{\rho\sigma} D_{\rho\sigma} \left[(\mu\nu|\rho\sigma) - \frac{1}{2}(\mu\sigma|\rho\nu) \right] \right\} \\ &= \frac{1}{2} \sum_{\mu\nu} D_{\mu\nu} (h_{\mu\nu} + f_{\mu\nu}) \end{aligned} \tag{2.21}$$

where the matrices $h_{\mu\nu}$ and $f_{\mu\nu}$ are integrals of the operators $\hat{h}$ and $\hat{f}$ in the AO basis. The matrix $\mathbf{D}$ is the one electron density matrix

$$D_{\mu\nu} = 2 \sum_{i}^{N/2} C_{\mu i} C_{\nu i}^{*}. \tag{2.22}$$

The sum, and therefore the HF energy, is only dependent upon the AO basis and the first $N/2$ MO coefficients. The remaining $N_{AO} - N/2$ orbitals are so-called virtual orbitals. They are of no special importance in the context of the HF approximation[15] but of utmost importance in the context of *post-HF* theories.

Summarizing, the approximation of an averaged field is also the weakness of HF theory. Although the HF approximation yields approx. 99% of the total energy, all electron-electron interactions are just described in a very simplified way. The final percent of the total energy is the *correlation energy*, yielded from dynamic electron-electron interactions. The HF approximation alone is not satisfying to model most chemical reactions and (especially) not to differentiate between different structures of the same system. This can be easily visualized by the energetic difference between a chemical reaction, typically in the range of hundreds of kJ·mol^{-1}, and the total energy easily exceeding multiple thousands of kJ·mol^{-1} already for a single atom[16]. To obtain a more exact treatment, *post-HF* or *correlation methods* are available and needed.

A more exact description of the correlation energy corresponds to a more exact result. One therefore can systematically improve the result by using a better correlation de-

[15] Only in Koopman's theorem virtual orbitals are to be considered.

[16] Krypton atoms have a total energy of approx. 1.2 GJ·mol^{-1}, the HF approximation yielding more than 99.9% of it. The, admittedly rather special, *van-der-Waals* interaction between two krypton atoms has a potential well of approx. 1.5 kJ·mol^{-1}.

scription. In principle, using a defined basis set for a defined system, one could obtain the total energy through a *full configuration interaction* (FCI). For this, the wavefunction is constructed as a linear combination of all possible Slater determinants, each with a different occupation of the molecular orbitals. Although this approach seems to be simple and elegant, it very quickly proofs to be prohibitive due to the sheer number of possibilities. Therefore, one uses other methods to obtain the correlation energy. The computational effort for these methods increases also dramatically with accuracy. Therefore, the systematic improvement is more of theoretical nature when dealing with large systems. For smaller systems – to-date up to approx. 50 atoms – it is a feasible technique though.

2.5.3 Møller-Plesset Perturbation Theory

Perturbation theory provides a simple approach to calculate the correlation energy. The basic idea is that a previously solved problem is not too different from the one to be solved. The previously gained solution does not need to be exact, an approximative one is also sufficient for this theory to apply. The only restriction is that previous solution is not allowed to be principally wrong.

One defines a Hamiltonian that consists of two parts

$$\hat{H} = \hat{H}^{(0)} + \lambda \hat{H}^{(1)} \tag{2.23}$$

where $\hat{H}^{(0)}$ is the operator of the unperturbed system, $\hat{H}^{(1)}$ the perturbation operator and λ a measure of the strength of the perturbation.

The Schrödinger equation including perturbation is again

$$\hat{H}\psi = E\psi. \tag{2.24}$$

Obviously, for $\lambda = 0$ the unperturbed SE results, since $\hat{H} = \hat{H}^{(0)}$, $\psi = \psi^{(0)}$ and $E = E^{(0)}$. For a finite perturbation, both the wavefunction ψ and the energy W can be written as

a Taylor expansion

$$E = \lambda^{(0)}E^{(0)} + \lambda^{(1)}E^{(1)} + \lambda^{(2)}E^{(2)} + \lambda^{(3)}E^{(3)} + \dots \tag{2.25}$$

$$\psi = \lambda^{(0)}\psi^{(0)} + \lambda^{(1)}\psi^{(1)} + \lambda^{(2)}\psi^{(2)} + \lambda^{(3)}\psi^{(3)} + \dots \tag{2.26}$$

The wavefunction and energy for $\lambda = 0$ are named zeroth order wavefunction and energy, respectively. Analogously, $\psi^{(1)}$ and $\hat{H}^{(1)}$ are first order corrections, $\psi^{(2)}$ and $\hat{H}^{(2)}$ second order corrections.

The *Møller-Plesset perturbation theory* uses the Fock operator as a reference

$$\hat{H}^{(0)} = \sum_{i}^{N/2} \hat{f}(i) = \sum_{i}^{N/2} \left[\hat{h}(i) + \hat{g}(i)\right] \tag{2.27}$$

and therefore yields for the perturbation operator

$$\hat{H}^{(1)} = \hat{H} - \sum_{i}^{N/2} \left[\hat{h}(i) + \hat{g}(i)\right]. \tag{2.28}$$

Inserting the Taylor expansions for both wavefunction and energy in the eigenvalue equation, one obtains

$$\begin{aligned} &\lambda^0 \left(\hat{H}^{(0)} - E^{(0)}\right) \left|\psi^{(0)}\right\rangle + \lambda \left[\left(\hat{H}^{(0)} - E^{(0)}\right) \left|\psi^{(1)}\right\rangle + \left(\hat{H}^{(1)} - E^{(1)}\right) \left|\psi^{(0)}\right\rangle\right] \\ &+ \dots = 0. \end{aligned} \tag{2.29}$$

When the expansions are stopped after a defined order n, $n + 1$ equations need to be solved. For every exponent of λ the corresponding term needs to be zero. If the wavefunction of nth order is known, one can calculate the energy correction of order $n + 1$[17].

[17]Indeed, by applying the *turnover*-rule, one can obtain the correction of order $2n + 1$ [69].

The HF wavefunction can be used as a reference function, since it is an eigenfunction of $\hat{H}^{(0)}$. The energies of zeroth and first order are

$$E^{(0)} = \left\langle \psi^{HF} \left| \hat{H}^{(0)} \right| \psi^{HF} \right\rangle = \left\langle \psi^{HF} \left| \sum_{i}^{N/2} \hat{f}_i \right| \psi^{HF} \right\rangle = \sum_{i}^{N/2} \epsilon_i \tag{2.30}$$

$$E^{(1)} = \left\langle \psi^{HF} \left| \hat{H}^{(1)} \right| \psi^{HF} \right\rangle = -\frac{1}{2} \sum_{ij} \left[2(ii|jj) - (ij|ji) \right] \tag{2.31}$$

which when added correspond to the HF approximation. Therefore, the first correction to the HF result is contained in the energy correction of second order, $E^{(2)}$. To calculate this term, the wavefunction of first order $\psi^{(1)}$ is necessary. The *Brillouin theorem* signifies that singly excited configurations do not interact with the reference and therefore do not take part in the first order wavefunction. The wavefunction is then constructed as a combination of doubly excited configurations

$$\left| \psi^{(1)} \right\rangle = \frac{1}{2} \sum_{ij} \sum_{ab} T_{ab}^{ij} \left| \phi_{ij}^{ab} \right\rangle . \tag{2.32}$$

Where the function $\left| \phi_{ij}^{ab} \right\rangle$ is noted by

$$\left| \phi_{ij}^{ab} \right\rangle = \hat{E}_{ai} \hat{E}_{bj} \left| \psi^{HF} \right\rangle . \tag{2.33}$$

The operators $\hat{E}_{ai}$ and $\hat{E}_{bj}$ are spin adapted excitation operators stemming from second quantization, which excite an electron from an occupied orbital i into a virtual orbital a and from j into b. The amplitudes T_{ab}^{ij} weigh different configurations.

Since the configurations ϕ_{ij}^{ab} are neither orthogonal nor normed, *contravariant* configurations and amplitudes are used

$$\tilde{\phi}_{ij}^{ab} = \frac{1}{6} \left(2\phi_{ij}^{ab} + \phi_{ji}^{ab} \right) \tag{2.34}$$

$$\tilde{T}_{ab}^{ij} = 2T_{ab}^{ij} - T_{ab}^{ji} \tag{2.35}$$

which possess the following properties

$$\left\langle \tilde{\phi}_{ij}^{ab} \middle| \phi_{kl}^{cd} \right\rangle = \delta_{ac}\delta_{bd}\delta_{ik}\delta_{jl} + \delta_{ad}\delta_{bc}\delta_{il}\delta_{jk} \tag{2.36}$$

$$\left\langle \tilde{\phi}_{ij}^{ab} \middle| \psi^{(1)} \right\rangle = T_{ab}^{ij} \tag{2.37}$$

$$\langle \tilde{\phi}_{ij}^{ab} | \hat{H} | \psi^{HF} \rangle = K_{ab}^{ij} \tag{2.38}$$

where the introduced matrix element K_{ab}^{ij} is an exchange integral of form

$$K_{ab}^{ij} = (ij|ab). \tag{2.39}$$

The contravariant configurations and amplitudes simplify the resulting relationships between matrix elements with excited configurations. The MP2 correlation energy is the first correction of the HF energy and is obtained as

$$\begin{aligned} E^{(2)} = \Delta E_{MP2} &= \left\langle \psi^{HF} \middle| \hat{H} \middle| \psi^{(1)} \right\rangle \\ &= \sum_{ij} \sum_{ab} \left\langle \psi^{HF} \middle| \hat{H} \middle| \tilde{\phi}_{ij}^{ab} \right\rangle \tilde{T}_{ab}^{ij} \\ &= \sum_{ij} \sum_{ab} K_{ab}^{ij} \tilde{T}_{ab}^{ij} \end{aligned} \tag{2.40}$$

To obtain the amplitudes, the *doubles residuals* need to be calculated for a specific electron pair excitation first. They can be obtained by multiplication of eq. 2.29 from left with a contravariant configuration as

$$R_{ab}^{ij} = \left\langle \tilde{\phi}_{ij}^{ab} \middle| \hat{H}^{(0)} - E^{(0)} \middle| \psi^{(1)} \right\rangle + \left\langle \tilde{\phi}_{ij}^{ab} \middle| \hat{H} \middle| \psi^{(0)} \right\rangle = 0. \tag{2.41}$$

This can be simplified using second quantization

$$R_{ab}^{ij} = K_{ab}^{ij} + \sum_{c} \left(f_{ac} T_{cb}^{ij} + T_{ac}^{ij} f_{cb} \right) - \sum_{k} \left(f_{ik} T_{ab}^{kj} + T_{ab}^{ik} f_{kj} \right) \tag{2.42}$$

where f_{rs} are the elements of the Fock matrix. If one uses canonical orbitals, which is

assumed up to this point, the expression simplifies to

$$R_{ab}^{ij} = K_{ab}^{ij} + (\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j)T_{ab}^{ij} \tag{2.43}$$

Now the amplitudes T_{ab}^{ij} can be calculated directly under the condition that the matrix elements K_{ab}^{ij} are known.

Summarizing these amplitudes with the eq. 2.40 yields

$$\Delta E_{MP2} = \sum_{ij} \sum_{ab} \frac{K_{ab}^{ij} \left(2K_{ab}^{ij} - K_{ab}^{ji}\right)}{\epsilon_j + \epsilon_i - \epsilon_a - \epsilon_b}. \tag{2.44}$$

In principle it is possible to account for higher order excitations. They are named MPn energies for a correction of order n. It should be denoted though that this series does not converge in all cases. Problems with oscillating or divergent energies are possible [69, 70]. Additionally, it is difficult to estimate the quality of the MP2 energy since the energy is highly dependent upon the quality (and in-principle applicability) of the HF reference wavefunction. This means that the correction easily gets overestimated in case of a bad description by the HF method.

The MP perturbation theory is size consistent[18] but not variational. The computing time of canonical MP2 scales with $\mathcal{O}(\mathcal{N}^5)$, where $\mathcal{N}$ is the size of the system. It is possible to reduce this scaling to linear scaling if further approximations are introduced which are described at a later point.

MP2 yields relatively reasonable energies for systems where HF is a valid approximation, making it possible to differentiate between different structures already at this level of theory in most cases.

[18] Size consistency says that the result of two molecules which are far apart and do not interact is identical to the doubled result of a single molecule.

2.5.4 Coupled-Cluster Theory

Just as the MP2 method, the *coupled cluster* (CC) theory is a size consistent but not variational method to calculate the correlation energy. In contrast to MP, CC converges to the FCI limit when introducing higher order excitations.

The basic assumption of CC is a wavefunction with an exponential excitation factor

$$\left|\psi^{CC}\right\rangle = \exp(\hat{T})\left|\psi^{HF}\right\rangle . \tag{2.45}$$

$\hat{T}$ is the so-called cluster operator, containing excitations up to an arbitrary order. The excitation operators of different orders are defined as

$$\hat{T}_1 = \sum_i \sum_a \hat{E}_{ai} T_a^i \tag{2.46}$$

$$\hat{T}_2 = \frac{1}{2} \sum_{ij} \sum_{ab} \hat{E}_{ai} \hat{E}_{bj} T_{ab}^{ij} \tag{2.47}$$

and analogous.

The exponential *Ansatz* of the wavefunction can be written as a Taylor expansion

$$\exp(\hat{T}) = 1 + \hat{T} + \frac{\hat{T}^2}{2!} + \frac{\hat{T}^3}{3!} + \frac{\hat{T}^4}{4!} + \cdots . \tag{2.48}$$

If excitations up to order $N \to \infty$ would be accounted for, the FCI limit would be obtained. It should be mentioned though that the size of the energetic contribution gets smaller with the order of excitation. Therefore, the inclusion of lower order excitations (up to second or third order) is normally sufficient to obtain a result of very acceptable quality. Including single and double excitations yields the systematic named CCSD. The Taylor expansion can then be written as

$$\exp(\hat{T}_1 + \hat{T}_2) = 1 + \hat{T}_1 + \left(\hat{T}_2 + \frac{\hat{T}_1^2}{2!}\right) + \left(\hat{T}_2\hat{T}_1 + \frac{\hat{T}_1^3}{3!}\right) + \cdots . \tag{2.49}$$

Higher order excitations are constructed from single and double excitations. This is the

basis of the size extensivity[19] of this method.

The CCSD correlation energy ΔE^{CCSD} is obtained by applying the Hamiltonian and multiplying from the left with the reference wavefunction as

$$\begin{aligned}
\Delta E^{CCSD} &= \left\langle \psi^{HF} \middle| \hat{H} \middle| \psi^{CCSD} \right\rangle \\
&= \left\langle \psi^{HF} \middle| \hat{H} \left(\hat{T}_1 + \hat{T}_2 + \frac{1}{2}\hat{T}_1^2 \right) \middle| \psi^{HF} \right\rangle \\
&= \left\langle \psi^{HF} \middle| \hat{H}\hat{T}_1 \middle| \psi^{HF} \right\rangle + \left\langle \psi^{HF} \middle| \hat{H} \left(\hat{T}_2 + \frac{1}{2}\hat{T}_1^2 \right) \middle| \psi^{HF} \right\rangle \\
&= \sum_{ai} T_a^i \left\langle \psi^{HF} \middle| \hat{H} \middle| \phi_i^a \right\rangle + \sum_{ij} \sum_{ab} \left\langle \psi^{HF} \middle| \hat{H} \middle| \phi_{ij}^{ab} \right\rangle \left(T_{ab}^{ij} + \frac{1}{2} t_a^i t_b^j \right)
\end{aligned} \tag{2.50}$$

Again, this relation can be simplified using second quantization

$$\Delta E^{CCSD} = \sum_{ai} 2 f_{ai} t_a^i + \sum_{abij} \left[2\mathbf{K}^{ij} - \mathbf{K}^{ji} \right]_{ab} \left(T_{ab}^{ij} + \frac{1}{2} t_a^i t_b^j \right) \tag{2.51}$$

where $\mathbf{K}$ are matrix elements between the HF ground state and excited configurations.

Just like in section 2.5.3, the single and double residuals are obtained by multiplying from the left with the corresponding contravariant configurations

$$r_a^i = \left\langle \tilde{\phi}_i^a \middle| \hat{H} - E^{CCSD} \middle| \psi^{CCSD} \right\rangle \tag{2.52}$$

$$R_{ab}^{ij} = \left\langle \tilde{\phi_{ij}^{ab}} \middle| \hat{H} - E^{CCSD} \middle| \psi^{CCSD} \right\rangle \tag{2.53}$$

A rather popular extension of the CCSD method is to use an additional, perturbative triples correction (T) to yield the CCSD(T) correlation energy as

$$\Delta E^{(T)} = \left\langle \psi^{HF} \middle| \left(\hat{T}_1 + \hat{T}_2 \right)^\dagger \hat{V} \hat{T}_3 \middle| \psi^{HF} \right\rangle \tag{2.54}$$

where $\hat{V}$ is the perturbation operator[20]. In the case of canonical orbitals, the orbitals

[19] Size extensivity is a term introduced by Bartlett [71]. Methods are size extensive if they scale linearly with the number of particles. The particles are allowed to interact for this relation.

[20] As a historical sidenote, the CCSD(T) approximation was first published by Raghavachari *et al.* in 1989 [72], two years after their paper on the very similar QCISD(T) approximation [73]. They

are decoupled and the triples correction can be non-iteratively solved, saving computing time.

The total computing time for CCSD(T), which was the method of choice for some highly exact references in the context of this work, scales with $\mathcal{O}(\mathcal{N}^7)$, keeping in mind that the CCSD iteration can be calculated independently of the (non-iterative) triples. Again, a reduced scaling is possible by applying further techniques.

CCSD(T) is the method of choice for single reference problems if enough computing time is available. Given a sufficiently large basis set, results of chemical accuracy can be obtained[21]. It also provides generally more exact results than the CCSDT method using non-perturbative triples.

2.5.5 Further Developments

The general tendency of *post-CC* developments is to reduce the needed computing time and still yield at least the same quality of results in comparison to canonical CC.

Two major branches can be named trying to accomplish this task. One tries to reduce the needed computing time by reducing scaling to linearity for a given method and basis set. Another tries to enhance the accuracy of a given method and basis set combination so that, e.g., a smaller basis set will yield the same accuracy as a larger one[22].

The least intrusive of the possibilities to reduce the scaling is probably *density fitting* (DF)[23]. By using an auxiliary basis set, a three electron integral can be decomposed into two electron integrals. Given a sufficiently large and accurate DF basis set, the error employed by this technique is negligible.

A more intrusive way of reducing the scaling are local correlation methods [74–80]. The

defined the perturbative triples energy as $\Delta E_T = \sum_t^T (E_0 - E_t)^{-1} |\langle\Psi|V|\Psi\rangle|^2$ where $(E_0 - E_t)$ is the triples excitation energy using the Fock Hamiltonian. They conclude, very cautious, with "We expect both the QCISD(T) and CCSD(T) schemes to be useful for studying electron correlation effects in molecules.".

[21] Chemical accuracy is normally defined as being less than a kcal·mol^{-1}.

[22] Lately, the two branches start to grow together again by combining lower scaling methods with higher accuracy extensions.

[23] Sometimes also referred to as *resolution of the identity* (RI).

canonical orbitals are transformed into a local molecular orbital (LMO) basis $|\phi_i^{loc}\rangle$

$$\begin{aligned} \left|\phi_i^{loc}\right\rangle &= \sum_{\mu} |\chi_\mu\rangle \, L_{\mu i} \\ &= \sum_{k} |\phi_i^{can}\rangle \, U_{ki} \end{aligned} \tag{2.55}$$

by means of an unitary transformation matrix $\mathbf{U}$ defined as

$$\mathbf{L} = \mathbf{CU}. \tag{2.56}$$

It should also be explicitly noted in this work that a localization alone only introduces an overhead but does not reduce the scaling in any way. The reduction of the scaling only arises from employing distance criteria based on the fact that electron correlation is a short-range effect decaying with r^{-6} and neglecting the more distant contributions. By careful tuning of these distance criteria and the neglect of long-range interactions, linear scaling correlation methods can be obtained for suitable systems[24] keeping the accuracy of a canonical method.

The other branch, namely to increase the accuracy without an increase in computing time, lately focuses on *explicitly correlated* methods such as the various R12/F12 methods. By introducing an explicitly correlated factor, a better fitting to Kato's cusp condition [81]

$$\left(\frac{\partial \Psi}{\partial r_{ij}}\right)\Bigg|_{r_{ij}} = \frac{1}{2}\Psi(r_{ij} = 0) \tag{2.57}$$

can be obtained already for smaller basis sets. Unfortunately, these factors require careful tuning since the cusp width is system dependent and are to-date not purely *ab initio*. Additionally, the factors complicate the integrals. This complication can partly be circumvented by neglecting certain integrals/contributions.

[24] For a general system a reduction to exact linearity might not always be possible.

2.6 Parametrized Methods

Parametrized methods, in contrast to ab initio methods, provide no possibility to (theoretically) systematically improve results, but can, careful parametrization and applicatebility to the problem assumed, result in impressively accurate results. The computing time is in some cases only a fraction of the time required for an *ab initio* method of the same quality.

In the context of this work, heavy use has been made of various parametrized methods and system specific reparametrization has been implemented and accomplished.

2.6.1 Force Fields

Force fields describe interatomic interactions by means of analytical expressions. Obviously, these interactions can be of very different nature. Force fields compensate for that by different functional forms. Of course it is impossible in the context of this work to describe every class of force field that is in use. Therefore, the discussion will be of exemplary nature, merely describing some commonly used patterns and possibilities to improve the modeling of a specific interaction.

In general, force fields compose the energy E_{FF} using some or all of the following terms

$$E_{FF} = E_{str} + E_{bend} + E_{tors} + E_{vdW} + E_{el} + E_{cross} \tag{2.58}$$

where E_{str} describes the stretching of a bond, E_{bend} the bending and E_{tors} the torsion of an angle, E_{vdW} van-der-Waals interaction, E_{el} electrostatic interactions and E_{cross} couplings of the first three terms.

As an example, the AMBER/GAFF force field [82] is given in its functional form since it has been used in the context of this work

$$\begin{aligned} E_{AMBER} = &\sum_{bonds} \frac{1}{2} k_b (l - l_0)^2 + \sum_{angles} \frac{1}{2} k_a (\theta - \theta_0)^2 + \sum_{torsions} \frac{1}{2} V_n \left[1 + \cos\left(n\omega\gamma\right)\right] \\ &+ \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} \left\{ \epsilon_{i,j} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\}. \end{aligned} \tag{2.59}$$

The force field complies very well with the definition in eq. 2.58. Also, the amount of parameters can be estimated for any non-trivial molecule containing different atom(types) in the context of this representative force field.

Getting back to the initial idea of improving a single term, the stretch energy is picked as an example. If we assume E_{str} to be the function of bond stretching between two atoms A and B, the easiest form is a Taylor expansion around the equilibrium distance R_0 which has been stopped after the second order term

$$E_{str}(\Delta R^{AB}) = E_0 + \left(\frac{dE}{dR}\right)_0 (\Delta R^{AB}) + \frac{1}{2}\left(\frac{d^2E}{dR^2}\right)_0 (\Delta R^{AB})^2. \tag{2.60}$$

Normally, one sets the term E_0 to zero and calculates the derivatives at the point $R = R_0$ [69]. Then the function can be rewritten as

$$E_{str}(\Delta R^{AB}) = k^{AB}(\Delta R^{AB})^2 \tag{2.61}$$

where k_{AB} is the force constant of the AB-bond. Obviously, in its easiest form the bond stretching energy is therefore described as an harmonic potential.

Of course cases exist where such a simple description is not sufficient for various reasons, requiring a refinement of the functional form. Either the Taylor expansion is then stopped at a higher order or a completely different functional expression is chosen. In the first case, unphysical behaviors might still arise, e.g., stopping the expansion after the third oder term yields an energy $-\infty$ for long bond lengths. Stopping after the fourth term yields $+\infty$, which is still unphysical [69], since of course a correct behavior would yield the dissociation energy D for long distances.

To obtain such a behavior, a different functional form is required. For example the Morse potential published 1929 [83]

$$E_{Morse}(\Delta R) = D\left[1 - e^{-\alpha \Delta R}\right]^2 \tag{2.62}$$

fulfills this requirement. D is the dissociation energy and the variable α can be obtained form the force constant k via

$$\alpha = \sqrt{\frac{k}{2D}} \tag{2.63}$$

Even the Morse potential is not the final solution since both the attractive and repulsive

branch of the potential are not absolutely accurate. Also the computational evaluation of the exponential function is more expensive than polynomial ones. The latter (speed) is very important for MD simulations and since most force field engines are optimized for a good MD performance, polynomial expansions are implemented in them. Only the criterium that the nth derivative at point R_0 is equal to the derivative of the Morse potential at the same point might occur in implmentations.

In the context of this work, specific interest has been also the improvement of the vdW-Term by means of a similar extension of the Taylor expansion. This is equivalent to the above described situation for the bond stretch term and is described in detail in the corresponding publication.

In principle, if a suitable analytical expression is chosen and the free variables are well-parametrized, an accuracy similar to the reference method/values can be obtained. Unfortunately, this proves to be very difficult for arbitrary systems and arbitrary particle interactions, making a system-specific parametrization a fruitful approach.

2.6.2 Semiempirical Methods

Semiempirical methods are, in general, based on a HF-like SCF cycle. They were developed at a time when computer time was highly expensive and rare, with the target of providing an accuracy comparable to HF at a fraction of the computational cost [84]. Since then, they evolved in parallel with the usable *ab initio* methods. Nowadays, both a higher accuracy and bigger systems are possible. Applications can actually model whole proteins with an accuracy far better than force fields [85][25].

The basic idea that all semiempirical methods have in common is to reduce the number of two electron integrals needed to construct the Fock matrix. Since this is the most expensive step of a HF calculation, it is the optimal working point. To achieve this target, a number of approximations are used. First, only valence orbitals are taken into account. Core orbitals are accounted for by either reducing the nuclear charge according to the number of electrons in them or by analytical expressions designed to model the combined interactions of the atomic core and the core electrons, obviously requiring

[25] Actually, this is based on a linear scaling SCF which uses localizations and distance criteria just as described in Sec. 2.5.5.

parametrization. The next step is to only use a minimal basis, normally composed of Slater type orbitals which show a more physical description than Gaussians.

The central approximation is *zero differential overlap* (ZDO), neglecting all products of kind

$$\mu_A(i) \cdot \nu_B(i) = 0 \tag{2.64}$$

where μ and ν are basis functions at atoms A and B.

It is important to note the products and *not* the integral over these products are set to zero, causing the overlap matrix $\mathbf{S}$ to be a unit matrix. More importantly, all one-electron three-indices integrals are set to zero and all two-electron three-indices and two-electrons four-indices integrals are neglected. Being the most numerous integrals, this obviously reduces both scaling and prefactor[26].

To provide still a sufficient accuracy, the few remaining integrals are turned into parametrized analytical expressions. The differences between the various semiempirical methods can be found in exactly how many integrals are neglected and what parametrization is used.

The definition of the Fock matrix can be rewritten as

$$F_{\mu\nu} = h_{\mu\nu} + \sum_{\lambda\sigma}^{N_{AO}} D_{\lambda\sigma} \left[\langle \mu\nu \,|\, \lambda\sigma \rangle - \langle \mu\lambda \,|\, \nu\sigma \rangle \right] \tag{2.65}$$

with

$$h_{\mu\nu} = \langle \mu | \hat{h} | \nu \rangle \tag{2.66}$$

and employing the standard semiempirical notation using λ, σ, ν and μ for the basis functions.

In the context of this work, only relatively modern semiempirical methods have been used, namely *Austin model 1* (AM1) [86], *parametrized method 3,5* and *6* (PM3/5/6) [87–90]. These methods include no additional approximations. The PM methods are based on AM1, only reparametrizing it and removing two Gaussians for the description of the core electron/nuclei repulsion. In general, the accuracy of these methods increases

[26] Actually, no numerical scaling will be given here since formally, the scaling stays $\mathcal{O}(N^3)$ due to needed matrix inversions. Practically, the scaling can be said to reduce to linearity. A discussion on this can be found in ref. [85]

in the row

$$\text{AM1} < \text{PM3} < \text{PM5} < \text{PM6} \tag{2.67}$$

which is also equivalent to the order of their development/parametrization [91].

The difficulty of these parametrizations is not to find a set of parameters fitting a certain system and some of its properties. The highly non-trivial task is to optimize the parameters universally, for as many systems as possible, describing as many properties as possible in a correct manner. If done correctly, these parameters are applicable for the atom in all bonding situations. For example, in difference to force fields[27], a carbon atom can be described in a methyl group as well as in an aromatic system and in a carboxyl function with the same parameters.

Semiempirical methods have been used in this project both with universal parameters as well as with system-specific ones. While having systematic weaknesses [69], these methods come into play in situations where the accuracy of force fields is not sufficient and the computational effort to use either *post-HF* theories or the DFT methods described in the next section cannot be afforded.

2.6.3 Density Functional Theory

The *density functional theory* (DFT) goes back to the work of Hohenberg and Kohn. By proving that the energy of the electronic ground state can be calculated from the electronic density ρ of the system [92], the difficulty is reduced to the definition of a functional[28] The final target of DFT is the definition of a universal functional. Since to-date this functional has not been found, various other functionals are in use and new ones are in constant development.

[27] The only exception being universal force fields (UFF) but their accuracy is orders of magnitudes worse than semiempirics.

[28] Assuming a basis with enough flexibility to describe the electron density.

The general form of a spin-independent functional is

$$\begin{aligned} E\left[\rho(\mathbf{r})\right] = &-\frac{1}{2}\sum_i \left\langle \phi_i | \nabla^2 | \phi_i \right\rangle + \int \nu(\mathbf{r})\rho(\mathbf{r}) d\mathbf{r} \\ &+ \frac{1}{2}\int\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r} d\mathbf{r}' + E_{xc}\left[\rho(\mathbf{r})\right] \end{aligned} \tag{2.68}$$

where $\nu(\mathbf{r})$ in the second term is the external potential, in a normal molecule equal to the potential defined by the atomic cores. This term is the Coulomb energy between electrons and atomic cores. The third term is the classical Coulomb energy between different electrons, the prefactor eliminating double counting. Finally, the E_{xc} term separates the different density functionals by means of an exchange correlation functional.

The first term is similar to the *ab initio* treatments of the kinetic energy. It should be noted however that although DFT is based on electron density descriptions, the kinetic energy is calculated by means of molecular orbitals ϕ_i. An *orbital free* theory has been used at the very beginning, employing density functionals also for the kinetic energy. Due to huge errors in the kinetic energy and non-bonding problems[29] it was eventually overcome by the current usage of orbitals, first proposed by Kohn and Sham [94]. They solved the equation

$$\left[-\frac{1}{2}\nabla_i^2 + \nu(\mathbf{r}) + \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} + \frac{\delta E_{xc}[\rho(\mathbf{r})]}{\delta\rho(\mathbf{r})}\right] |\phi_i\rangle = \epsilon_i |\phi_i\rangle \tag{2.69}$$

in a self-consistent manner for each orbital, defining the electron density as

$$\rho(\mathbf{r}) = 2\sum_{i=1}^{N} \phi_i^* \phi_i. \tag{2.70}$$

The biggest advantage of DFT is speed paired with accuracy. The computational effort is comparable with HF calculations but the results are, through the implicit treatment of electron correlation, generally better than HF.

As an example for how the exchange correlation functional of a typical DFT functional

[29]The non-bonding theorem was introduced by Lieb and Simon, proving that in the context of the Thomas Fermi Dirac model, no molecular system would be stable compared to dissociation. A property which has been briefly yet accurately summarized as "Goodbye World!" in another thesis [93].

is composed, we pick the commonly applied B3LYP functional [95]. The exchange correlation term of B3LYP is defined as

$$E_{xc}^{B3LYP} = 0.2\ E_x^{HF} + 0.72\ E_x^{B88} + 0.08\ E_x^{S} + 0.81\ E_c^{LYP} + 0.19\ E_c^{VWN80} \qquad (2.71)$$

where E_x^{HF} is the exact HF exchange energy as given in eq. 2.18, hence the definition of a class of *hybrid functionals*. Further exchange terms are added, stemming from the gradient corrected B88 functional [96] and the Slater Dirac functional [97]. To describe electron correlation, electron correlation terms are added from the locally approximated VWN80 functional [98] and the gradient corrected LYP functional [99].

The shown numerical parameters are optimized to reproduced ionization potentials, proton affinities and atomization energies of the G1 testset of molecules. In the context of a critical assessment of density functional theory, these exact numerical parameters cause the conclusion of that it is a *semiempirical* method[30].

DFT is based on a universally applicable functional, without knowing either its form nor its sheer existence. Therefore a great number of functionals have been developed and a choice needs to be made for every system whether a certain functional is "suitable"[31].

Additionally, even with DFT probably being the most evolved of the semiempirical methods, no systematic improvement of results is possible in contrast to wavefunction based methods. In the context of structure prediction of clusters, where the cluster is normally stabilized by non-bonded interactions of long-range nature, it should be noted that DFT has *per definitionem* extreme shortcomings in this field due to the wrong asymptotics of all standard functionals.

[30] Attempts to describe density functional theory either as *ab initio* or *from first principles* are just trying to obfuscate its true nature, namely being – as long as *the* functional has not been found – a parametrized and therefore *semiempirical* method.

[31] "Suitable" being equivalent to "parametrized for this particular case".

2.7 Programming Techniques

As this cannot be a textbook on theoretical chemistry, this is can even be less of a reference on programming techniques. Numerous detailed and general textbooks exist on this subject [100, 101], therefore the only target is to motivate why the choice has been made for the used programming technique and which inherent advantages it provides for this project. The general importance of this topic arises from modern theoretical chemistry being highly dependent on the computer as its major "machine". Therefore, the "toolbox", the programming languages and design principles, is of utmost importance[32].

2.7.1 Choosing a Programming Language

Multiple hundreds of programming languages exist. The fewest of them share a similar scope, resulting partly also from their very different dates of development. Obviously, with such a broad variety, only a few can be considered to be of either universal and/or fitting nature. Additionally restrictions can be either compatibility with legacy code or external libraries[33]. For this project, the only candidates were C/C++, Java and FORTRAN.

Among these, FORTRAN is the oldest language. FORTRAN originally appeared in 1953, taking its name from the abbreviation for *FORmula TRANslation*. Since then, the specification got extended a couple of times, in principle preserving backwards compatibility. Having a strong background in scientific computing from the very beginning, a lot of legacy code exists today and is still actively maintained and extended. FORTRAN has a reputation as providing an excellent performance and a relatively easy syntax. Drawbacks exist as well, namely the complete neglect of *object-orientation*[34], a complicated string parsing for input and very limited availability of recent data structures like, e.g., hash tables and trees.

[32] It will not be discussed here, if the mere pressing of buttons already qualifies as "using the machine" or if Feynman's quotation applies.
[33] For example, binary compatibility with some existing program suite or the ability to be parallelized.
[34] Latest specifications try to extend FORTRAN in that direction. The huge amount of legacy code makes a transition towards that direction relatively difficult.

The original C was introduced at the same time as UNIX operating systems got available in 1973[35]. The specification got very quickly extended in 1979 to support the upcoming *object-oriented* programming model, the fork supporting that was named C++ but is today still backwards compatible with C. Since then C/C++ stayed unchanged, including only minor revisions to make the specification even more powerful. Despite the powerful specification and very good performance, C/C++ is very closely tied to the underlying hardware and operating system level. While this is necessary for e.g. kernel developments, it complicates the portability of programs to other architectures. Even more, since the standard does not guarantee numerical stability among platforms of different bitness, it can make porting from e.g. 32 bit to 64 bit very difficult[36].

Java, on the other hand, is a relatively young language, initially developed by Sun Microsystems in 1995 as *Project Oak*. It was designed to be a *consolidation language*, with the idea in mind to provide a C/C++-like syntax without needing to comply to any backwards compatibility, eliminating certain build-in flaws to C/C++ like highly architecture specific pointer arithmetics and adding strengths of other languages, like a *garbage collector* (GC) and type-safety. Combined with a totally new approach towards architectural neutrality and the promotion of *object orientation* from a "later add-on" to a build-in language feature, Java was a big step towards making modern programming tools popular [103]. Additionally, the *String* is a build-in object and easy string parsing was a major target in the language design.

The decision was made to use Java in the context of this project. Being architectural neutral, purely object-oriented and providing a similar performance to native languages like C/C++ [17, 104][37], it eases the development process of bigger frameworks.

[35] Ironically enough, the work on C began in the context of developing a FORTRAN compiler in the Bell Laboratories. [102]

[36] Universal data types provide a remedy to this problem but they are seldomly used.

[37] The problem with comparing Java to C/C++ performancewise lies in the very different language approaches. Tiny microbenchmarks will very likely show a far better performance of C/C++. Longer, more realistic benchmarks tend to show a converse picture [17]. Another fact worth mentioning is the difference between different execution environments and versions for Java. In general, the latest JRE/JDK in its server form is the best pick for computing intensive applications. It is also worth mentioning, that the concept of a virtual machine for code execution now becomes popular in the C/C++ world due to its performance advantages [105].

2.7.2 Object-Oriented Programming

Object orientation is sometimes misunderstood as just providing means of bundling informations within a data structure and accessing these informations either directly or via *getter/setter* functions[38]. This is not only ignoring higher-level programming techniques like *design patterns* [106] but also the very basics of object orientation.

Namely, the three signs for an object-oriented design are

1. encapsulation,

2. class hierarchy and inheritance and

3. polymorphism.

Encapsulation is sometimes explained in a simplified manner that each object "hides its internal datastructure from the rest of the system". While this is not wrong, it misses the more important point of "hiding the internal implementation from the rest of the system". The difference between the two is subtle, yet important. An object is more than just the sum of its fields, the *state*, it also is important how that state is accessed and manipulated through functions. The implementation of these functions can and should be outside the scope of the rest of the system, also to allow for an easy replacement of those with, e.g., a better algorithm. Encapsulation is the major tool of providing a good maintainability of any non-trivial program. Once an object is fully implemented and tested, it should not be affected by changes in the rest of the program and vice versa.

Class hierarchy and *inheritance* was the keystone of any object oriented design. The *class* being the abstract description of both state and functions of an object, class hierarchy describes the relationship between a general *superclass* and its specialized *subclasses*. Inheritance, the formalism how subclasses access properties of the superclass, as a pattern has lost some of its popularity in the advent of *interface-driven* development, being partly replaced by stronger polymorphism.

Polymorphism as the last sign for a good object orientation is, in the context of an

[38] Although it is possible to argue that *getter/setter* functions are in general bad style within an object-oriented design and should be omitted, they are a necessary evil and still far better than direct access to the state of an object.

interface-driven development, of utmost importance. It describes that several classes that do not need to be dependent upon each other through inheritance can be manipulated through a common set of methods.

2.7.3 Parallelization

Parallelization is the concept unifying all attempts to speed a program up using more processing units. On a theoretical level, the speedup can be approximated[39] for any program using *Amdahl's law* [107]

$$\text{Speedup} = \frac{1}{(1-P) + \frac{P}{N}} \tag{2.72}$$

where P is the portion of the program that benefits from parallelization and N is the number of processing units. A maximum speedup can be obtained by examining the asymptotic behavior

$$\text{max. Speedup} = \lim_{N\to\infty} \frac{1}{(1-P) + \frac{P}{N}} = \frac{1}{(1-P)}. \tag{2.73}$$

In the limit of an infinite number of processing units[40], the maximum speedup only depends upon how big the parallelizable fraction is. As an example, if (excellent) 99% of the program benefit, the maximum speedup is still only 100.

It is therefore of utmost importance to eliminate serial bottlenecks in the program, if a good parallelizability is targeted. Nevertheless, it should be kept in mind that an infinite speedup is impossible, since some serial bottlenecks, like I/O, cannot be circumvented[41], even if a so-called *trivial parallel* case is subjected.

In the implementation of parallelization, two major classes can be spotted. One is the case where the processing units share the same resources, like memory and storage. The other is the case of independent subunits that only communicate by means of an

[39] Amdahl's law stems from a time when parallel execution was rather unpopular and therefore is nowadays considered to be too pessimistic.

[40] A size that already contemporary supercomputers reach from a practical point of view.

[41] Garbage collection to-date is also mostly a serial process, therefore efficient memory allocation is important. A new *garbage first* (G1) collector is designed to utilize parallel environments [108] but is to-date only available as a beta.

interlink, like a network connection. These two classes are e. g. represented by *symmetric multiprocessing* (SMP) and *massively parallel processing* (MPP). In the context of this project, both parallelization schemes have been implemented to provide flexibility and optimal usage of different computing conditions. SMP has been used through Java threads[42], MPP via the MPI standard [109].

2.8 State of the Art

The discussion within this section will be restricted to the state of the art in the field of programming techniques for scientific applications and the field of the global optimization of cluster structures.

Both parameter optimization and molecular design using global optimization techniques are excluded, the latter due to very little work done so far[43], the earlier due to the sheer amount of applications carried out.

2.8.1 Programming Techniques

Independent of the chosen programming language or problem to be solved, certain ultimate requirements to any program can be formulated and defined as being state of the art. The most important ones are

- ♢ Efficiency/performance
- ♢ Reliability
- ♢ Robustness
- ♢ Usability
- ♢ Portability

[42] Actually, due to its architectural neutrality, Java also allows for using other shared memory architectures like (cc)NUMA.

[43] Section 7.1 contains a literature overview of recent work.

♢ Maintainability

Efficiency is the optimal usage of available ressources. In the advent of multi-core and multi-processor architectures, this explicitly includes optimal parallel computing for both shared and distributed systems. *Reliability and robustness* both target a good quality control of the code. *Portability* is of outmost importance especially in the scientific community. When working with cutting edge hard- and software, prerequisites change rapidly. This includes both the architecture (to-date x86-64, SPARC and POWER are the most common ones) and the operating system (to-date Linux, *BSD, Solaris and AIX among others) level. *Maintainability* also is of utmost importance for scientific applications where algorithms are constantly updated and/or replaced.

An assessment whether above mentioned targets are met by a specific program should also consider the background of the program. The history of most programs written for scientific applications dates back several decades. A good example are program packages for electronic structure calculations. The major packages (with one exception [110]) have roots dating back to the 1970s and are therefore written in different FORTRAN dialects. These programs have definitely been state of the art back then and today still fulfill part of the requirements mentioned above, namely, they are mostly very efficient for serial applications as well as reliable and robust due to the long testing period. Unfortunately, they can only adopt slowly to both new algorithmic developments as well as different environments due to their high inertia stemming from their procedural design. Therefore, their portability and maintainability is limited, clouding their future prospects.

The Orca program package for semiempirical, DFT and *ab initio* calculations [110] can be considered state of the art in the field of quantum chemistry. The object-oriented development model in C++ allows for a good maintainability, robustness and portability[44]. Additionally, the implementation is efficient and reliable due to a strict quality control.

The above described relations apply cleanly to program packages for the global optimization of clusters. Traditional programs, e.g. ref. [22], have been implemented employing a procedural design and targeted at solving a specific task, e.g. the global optimization of

[44] A proof for this claim is for example that the development for this program package started in 2000 and to-date includes at least the features other general purpose packages for electronic structure calculations contain.

water clusters using a TTM2F force field, in the most efficient way. Obviously, this constructs constraints on future implementations and applications that might be tedious to resolve. Therefore, the transition from application-specific programs to general-purpose frameworks needs to made. Using current development techniques and platforms, this task can be accomplished, meeting all the above mentioned requirements. A new development can profit from experience gained in previous – application-specific – developments, cicumventing design errors and remodelling well-designed properties. The from-scratch development of such general-purpose framework is the central aim of this work.

Any from-scratch development should make use of object-oriented programming since it eases the development process of bigger frameworks. Design patterns can be used as efficient solutions to commonly encountered development challenges [106].

2.8.2 Cluster Structure Optimization

In the global optimization of cluster structures, the algorithmic state of the art is highly correlated to the accessible systems. Algorithmic progress is made to explore new system sizes or entirely new systems. Therefore an overview is given on the studied types of systems to-date and the algorithms required to make them accessible. Full reviews on cluster structure optimizations can be found in refs. [54, 55, 57, 111–113] . Additional conclusions can be found in ref. [114].

Cluster structure optimization is to-date still dominated by atomic clusters. Within this field, Lennard-Jones (LJ) clusters are traditionally used as model systems to benchmark new algorithmic developments. Homogenous LJ cluster have been studied up to LJ_{150} using phenotype GAs, basin- and minimum-hopping [21, 39, 41, 115]. More recent developments [25, 26, 28, 116] have pushed this border up to $n = 561$ by introducing small variations to the basic algorithms and/or constructing temporary dynamic grids for particle placement. The construction of such grids is designed to overcome a traditional problem when dealing with huge systems. The general structural type assembles relatively fast but the optimal placement of atoms at the surface takes far more steps in comparison. Employing biased optimization techniques that make use of prior knowledge, some insight can be given on structures up to 1000 LJ atoms [29, 30, 117].

Heterogeneous LJ clusters have only been studied comparatively little. Using a simplified

LJ potential, Doye has studied binary LJ clusters employing basin-hopping techniques [118] up to 95 atoms. Small binary argon-xenon LJ clusters (up to 20 atoms) employing a standard LJ(6,12,6) potential and unbiased basin-hopping techniques have been studied [119]. Calvo *et al.* used binary LJ clusters as a benchmark for a novel Monte Carlo minimization with explicit exchange moves [120]. Additionally, recent work using basin-hopping techniques has been carried out, studying connectivity in the energetic landscape of binary LJ clusters [121].

Morse clusters are a another field of active study. Just like in the case of LJ clusters, homogenous systems have been studied more intensively than heterogenous cases. The only study of binary Morse clusters to-date was carried out using basin-hopping techniques [122]. Both LJ and Morse clusters serve mainly as model potentials where a functional evaluation is computationally cheap.

The development of the phenotype operator and hybrid techniques utilizing local optimization steps by Deaven and Ho [23] marked an important step. Initially, a buckminster fullerene could be located as the global minimum of C_{60} but also systems became accessible that could be studied experimentally. Studies focus on metallic clusters, salts and nanoalloys, homogenous or bimetallic, by means of different optimization techniques.

This includes various studies using the GUPTA potential for bimetallic compounds by Johnston and co-workers [111, 123, 124] employing phenotype GAs. The GUPTA potential is a force field parametrized against DFT reference calculations and experimental data. The flexibility of the DFT reference allows for reliable studies of most metals, systems which are normally not accessible to standard force fields. Interestingly, the so obtained structures show structures similar to some binary LJ clusters. For these systems, explicit atom exchange is of special importance [111]. Obviously, this importance increases if even more heterogenous systems are subject to study.

Similar to EA-based techniques, basin-hopping techniques dealing with binary systems have focused mainly on nanoalloys and clusters of metal oxides and salts [112]. Within these systems, simulated annealing persists to be a valid choice if big systems (e.g. up to 512 ZnO units) are targeted and its shortcomings – likely convergence to a good yet not optimal minimum – are accepted.

One can draw the conclusion that the state of the art in the field of the global opti-

mization of atomic clusters is the treatment of binary clusters by means of force field potentials. Depending upon the system under study, these potentials might be parameterized against higher levels of theory and/or experiment. System sizes above approx. 100 building blocks still require extensions to all mentioned global optimization techniques, introducing some likely bias.

Molecular clusters do introduce another difficulty for global optimization. Besides the position of the building block, their orientation needs to be optimized simultaneously even when assuming frozen internal degrees of freedom. An additional inclusion of internal degrees of freedom enlarges the search space to be captured even more. Therefore, the first application of pure genetic algorithms for the global optimization of molecular clusters by Xiao and Williams in 1993 [125] studied benzene, naphtalene and anthracene clusters only up to four building blocks, with frozen internal degrees of freedom. This constraint mainly persists, effectively reducing the systems to those consisting of small building blocks such as water.

Water clusters have been studied thoroughly on different levels on theory. Noticeable (algorithmic) work has been carried out within this working group during the last decade. Employing phenotype crossover operators, pure water clusters up to 30 building blocks could be studied using force fields with frozen internal degrees of freedom [126]. Using a flexible TTM2F force field, water clusters up to 34 water molecules could be studied [22]. The latter study used a pool concept in contrast to the traditional generational concept together with niching (a concept designed to preserve different structural motifs in the genetic pool) and directed mutation. Employing minimum hopping techniques, the earlier, unflexible size regime was recently extended to 37 water molecules [127].

Water clusters have also been studied on surfaces [128], in bucky balls [129] and doped with other atoms or molecules [130] employing standard techniques. Among the algorithmically more advanced applications are the modelling of clathrates by means of genetic algorithms [131]. Methane clathrates were studied using phenotype GAs. By modelling pressure effects, insights on clathrates in deep-water environments could be given. Also microsolvation studies have been carried out on various ions within smaller water clusters [132–134] up to 24 water molecules including the prediction of electronic properties to assist experiment.

Basin-hopping techniques have been applied to molecular clusters of pure oxygen and

nitrogen [135, 136] up to 38 molecular building blocks. These serve well as examples for the energy gap between structures of very different motifs closing with cluster and building block size.

Besides the mentioned molecular clusters consisting of small building blocks, clusters are studied were frozen internal degrees of freedom are a valid approximation. Besides the earlier mentioned clusters of benzene analoga, other (possibly polycyclic) aromatic compounds are accessible. For example clusters of coronene have been studied employing parallel tempering Monte Carlo techniques [113] up to 16 coronene units.

Studies of molecular clusters consisting of bigger building blocks have been carried out for example by Doye and Wales for clusters of Buckminsterfullerenes [137, 138]. In both studies, potentials were used that, by treating the bucky ball building block as a single pseudoatom, effectively reduce this problem to the optimization of an atomic cluster.

Summarizing, the global optimization techniques have been very successfully applied to a multitude of different cluster systems. Remaining challenges are

- ♢ *Heterogeneity:* Systems with three or more different species.
- ♢ *Flexibility:* Systems consisting (partly) of molecular building blocks with a high(er) amount of internal degrees of freedom.
- ♢ *Method independency:* Allow for global optimization on various levels of theory.

Method independency is of special importance when dealing with heterogenous and possibly flexible systems. The global optimization of clusters requires highly exact yet computationally cheap evaluations of the system energy. Traditionally, this requirement was met by highly specialized yet very unflexible potentials. When changing systems – once possible without constraints –, the optimal description changes. Any new development in the field of global cluster structure optimization should take these challenges into account, requiring universality by design.

CHAPTER

3

The OGOLEM Framework

I am rarely happier than when spending an entire day programming my computer to perform automatically a task that it would otherwise take me a good ten seconds to do by hand.

Douglas Adams

3.1 Scope of the Project

The overall target of this dissertation project is the development of a new framework for the global optimization of arbitrary clusters. In this context, the following publication describes the development efforts.

The paper is meant as an introduction to the program suite. It provides informations on the general, object-oriented program design and by presenting some test cases, shows parts of the functionality of the from scratch developed framework.

By using the framework for the global optimization of highly mixed Lennard-Jones clusters, it is shown that totally new systems can be targeted by design and no restrictions are necessary anymore in the maximal diversity of building blocks[1].

Using the framework for the global optimization of a biologically relevant molecule with a high number of internal degrees of freedom, the complete erasure of systematic restrictions is proven.

Additionally, scalability was a major concern in the development, therefore both shared memory as well as massively parallel processing are evaluated and presented. Showing linear scaling to at least 256 cores[2] under MPP conditions, the application of Amdahl's law yields a parallelized fraction of at least 99.6%.

3.2 Own Contribution

The own contribution of this project is the extension of *genotype* genetic operators described in ref. [1] to arbitrary mixtures of building blocks and the development of a packing operator, the implementation of MPI-based and Java threads-based parallelized frontends to a pool algorithm described in ref. [22]. Interfaces to the mentioned program packages have been written and extended since. Additionally, a classical Lennard-Jones

[1] A building block is defined as being a molecular or atomic part of the cluster. In the studied LJ clusters, this would translate to a single rare-gar atom. In the case of the Kanamycin clusters, this is a Kanamycin molecule.

[2] It was not possible to benchmark with more cores due to hardware restrictions.

(LJ) force field was implemented for arbitrary compositions of LJ clusters. These implementations were carried out in a strictly object-oriented fashion, amounting to more than 17 000 SLOC[3] to-date for these parts of the program.

All calculations for this project have been carried out using above framework by the first author.

3.3 Publication

Authors Johannes M. Dieterich and Bernd Hartke

Institut für Physikalische Chemie
Christian-Albrechts-Universität
Olshausenstraße 40
24098 Kiel, Germany

Title ogolem: Global Cluster Structure Optimization for Arbitrary Mixtures of Flexible Molecules.
A Multiscaling, Object-Oriented Approach.

Submitted September 06, 2009

Accepted October 23, 2009

Publication Data *Mol. Phys.* **108**, 279, (2010)

[3]All measures have been carried out using sloccount [139].

3.4 Additional Information

3.4.1 Collision Detection and Dissociation Detection

Any attempt at implementing a collision detection (CD) needs to deal with pairwise distances. The approach chosen in this work has been a simplistic one. By calculating pairwise distances of all atoms in the molecule, the method obviously scales with $\mathcal{O}(N^2)$.

A better scaling method would be a grid-based one, only requiring linear scaling. Such algorithm would need to construct a dynamic grid first[4] and assign all atoms to cells. By only calculating the pairwise distances between atoms in neighboring cells, linear scaling can be obtained. But the given description also gives an idea what overhead one encounters with such approach against simply using two nested loops. This would increase the prefactor, delaying the crossing point between the low prefactor $\mathcal{O}(N^2)$ and the high prefactor $\mathcal{O}(N)$ algorithm. Therefore, such approach only makes sense when the number of atoms in the system gets rather big and a bottleneck in this spot is encountered. To-date, even with systems of more than 200 atoms (four Kanamycin A plus ions), no such bottleneck was obtained. Therefore, no implementation of a grid-based algorithm seemed necessary yet.

The dissociation detection (DD) is a less obvious topic, since a simple pairwise approach is not sufficient to decide whether parts of the cluster are dissociated off. We used the graph-based Warshall algorithm [140].

A graph is a representation of points, so-called *vertices*, and their connections, the *edges*. A mathematically probably more pedantic definition can be found in ref. [141][5]:

> A *graph* G is a finite nonempty set V together with an irreflexive, symmetric relation R on V. Since R is symmetric, for each ordered pair $(u, v)\, \epsilon\, R$, the pair (v, u) also belongs to R.

This definition can obviously also be applied to chemical molecules, or, in a more general

[4] It needs to be dynamic since the cluster can change its shape radically at different points of the global optimization.

[5] The reference [141] covers the basics of graph theory in a delightful and humorous way.

sense, to interacting atoms. Every atom is then represented by a vertex, every interaction (independent whether it is of bonded or non-bonded kind) as an edge.

The Warshall algorithm scales $\mathcal{O}(N^3)$, which is of course higher than a possible linear scaling when again employing a grid-based algorithm. As can be seen from the code depicted in fig. 3.1, the algorithm can be implemented in a very straightforward and memory-saving manner, again providing an advantage against the overhead introduced by introducing grids. Even more, since no direction needs to be considered – a chemical bond/interaction is bidirectional[6] – one can make use of that, lowering the prefactor of the algorithm.

Given an adjacency matrix, Warshall's algorithm transforms the adjacency matrix by means of very efficient bitwise operations into a reachability matrix. Also the memory footprint of this algorithm is only a $\mathcal{O}(N^2)$ scaling array of integers, the adjacency/reachability matrix. The reachability matrix can then be used to check for dissociated parts of the cluster since they will of course not be reachable for the rest of the cluster and vice versa.

Once more the question occurs, when the scaling penalty introduces a bottleneck for the total global optimization. Again, this has to-date not been the case.

In case that possible systems cross the border where either the implemented CD and/or DD are starting to be inefficient, above described grid-based technologies need to be implemented. At this point in time, the systems are of a size where the implemented algorithms together are at most responsible for approx. 3% of the total CPU cycles.

3.4.2 Object-Oriented Design Concepts

As pointed out before, the part of the framework dealing with the global optimization of clusters alone amounts to more than 17 000 SLOC. In the context of an object-oriented design, this translates into more than 90 classes for this part of the framework. It is therefore not feasable to give a detailed explanation on every design descision during or in advance of the development process.

[6] Mathematically more exact: A chemical system made of atoms and interactions is a graph and not a digraph.

```
/**
 * Transforms an adjacency matrix into a reachability matrix using
 * Warshall's algorithm.
 */
static int[][] warshallDistances(int[][] adjMat) {
    final int length = adjMat.length;
    for (int k = 0; k < length; k++) {
        for (int i = 0; i < length; i++) {
            for (int j = 0; j < length; j++) {
                /*
                 * if there is a path from vertex i to vertex j that does
                 * not go through any vertex higher than k then value
                 * is 1.
                 * |= : bitwise OR and ASSIGN
                 * & : bitwise AND
                 */
                adjMat[i][j] |= adjMat[i][k] & adjMat[k][j];
            }
        }
    }

    // adjacency matrix is now actually the reachability matrix
    return adjMat;
}
```

Figure 3.1: General Java code for Warshall's algorithm.

The explainations given will focus on how an object-oriented design was helpful in achieving the targets of this project, namely the global optimization of arbitrary clusters on arbitrary levels of theory.

Getting back to the previously mentioned signs for an object-oriented design, the first step is a good abstraction and encapsulation of data and implementation structures. Discussing data structures first, the natural separation in the global optimization of clusters is between a whole cluster structure and a single building block. This has been realized within this project through the two classes `Geometry` and `Molecule` for the complete structure and the single building block, respectively. This causes the universality of the OGOLEM framework since the `Geometry` data structure is absolutely independent of the actual chemical structure of the `Molecule` in terms of treatment/manipulation. Additional data structures to be named explicitly are the `CartesianCoordinates` and `ZMatrix` objects for external and internal coordinate structures. These coordinate systems also provide a good example for the importance of the hiding of implementations from the rest of the program. The `CoordTranslation` object consists only of `static` functions – they can be used without creating a `CoordTranslation` object and its state – for the transformation of different coordinate types into each other. The function translating `ZMatrix` objects to `CartesianCoordinates` ones is `ZMatToCartesians`. This encapsulation helps in case that at a later point in time more efficient means of coordinate translation are found to reduce the required implementation effort to the absolute minimum.

The other steps towards a good object-oriented design, polymorphism and class-hierachy combined with the mentioned design patterns[7], are necessary to obtain a well-structured, easily-maintainable code. As examples, the *Singleton* pattern is used within the program to protect the genetic pool from existing more than once at runtime (which would obviously be plainly wrong). Although this might sound unnecessary and trivial, it provides an unbreakable rule for future extensions of the framework and therefore helps future developers of the codebase. It is also a good example for design patterns not being *black magic* but actually only minor (in this case less than 10 SLOC) extensions to the code making the program architecture more durable.

The *Bridge* design pattern has been used as a polymorphic representation of the local

[7] A description of the most important design patterns accompanied by reference implementations in Java can be found e.g. in ref. [106]

optimization algorithms. Within the framework, no direct access is made to any of the local optimization routines used through different program packages on various levels of theory. Instead, the tasks are abstracted in the `Newton` interface and unified in the `LocalOpt` bridge. All implementations of local optimization engines implement the interface and are accessed through the bridge. Therefore, no direct method call is made in any part of the program – except for the actual bridge – into the actual implementations. Extending the OGOLEM framework to support another program package for local optimization is then only a task of the actual implementation, an addition in the bridge and flags in the input to be able to choose this implementation. The same pattern was carried out for the global optimization algorithms and the call to energy and gradient evaluations.

Interfaces have been used excessively in other parts of the program as well. The development is interface-driven, reducing the general need for an explicit class-hierachy since polymorphism can be reached either by using the `extends` keyword (inheritance from a superclass) or the `implements` keyword (implementation of an interface) in the Java language. The pure usage of interfaces is the superior choice if the polymorphic objects share little or no common state. No general preoccupation against the `extends` keyword should be assumed from this, it has been used to allow for *multiple inheritance* through interfaces. In general it is attempted to find the optimal solution for a given design problem, preferably employing well-tested approaches (design patterns). Arguing whether this succeeded in all parts of the program is fruitful and an absolute must if a good design is ultimately targeted.

CHAPTER

4

Benchmarking the Framework

Do, or do not. There is no 'try'.

Yoda

4.1 Scope of the Project

Any new development in the field of software ultimatly requires both a test of whether its behavior is correct and whether its performance is acceptable[1]. This is difficult to evaluate with a stochastic-heuristic algorithmic approach, as followed in this work, since no reproducible predictability exists. Therefore, known problems or special benchmark functions are solved and the results, both accuracy and performance, are analyzed.

Different communities use different problems to benchmark global optimization. Analytical benchmark functions are rather universally applicable, making them a good choice for general benchmarking purposes.

The project also studied the scaling of a set of benchmarking functions with the problem dimensionality. The scaling is a crucial measure for the comparability with real-world problems. The solution of real-world problems found in chemistry normally scales at least $\mathcal{O}(N^3)$ with the problem dimensionality [21], requiring any benchmark function to resemble this behavior.

Through the study of the scaling of a set of standard benchmark functions [8, 142–145] up to 500 dimensions[2], the scaling was found to be sub-quadratic in all cases. Since these benchmark functions are commonly assumed to be very difficult to solve already in far less than 50 dimensions, these results come as a surprise. A proposal has been made to use a new class of benchmarks for global optimization, GRUNGE, consisting of randomized Gaussians.

4.2 Own Contribution

This project was carried out in the context of the program part designed to (re)optimize parameters of potential functions. Requiring a new program part, the development effort was significantly reduced due to the reusability of prior existing objects. Both the implementation of the so-called `ogolem.adaptive` part of the framework and the

[1] In general, no claim is made that the presented programs are absolutely *bug-free*. Such claim would be doomed for any non-trivial program.

[2] In one case up to 1000 dimensions.

implementation of the benchmark functions mentioned in the publication were carried out, amounting to more than 6000 SLOC to-date.

All calculations were carried out and analyzed by the first author.

4.3 Publication

Authors Johannes M. Dieterich and Bernd Hartke

Institut für Physikalische Chemie
Christian-Albrechts-Universität
Olshausenstraße 40
24098 Kiel, Germany

Title Certain classes of standard benchmark functions are too simple for evolutionary global optimization

Submitted November 29, 2010

Accepted

Publication Data *J. Theor. Comput. Chem.*

Certain classes of standard benchmark functions are too simple for evolutionary global optimization

Johannes M. Dieterich

Institut für Physikalische Chemie, Christian-Albrechts-Universität, Olshausenstraße 40, 24098 Kiel, Germany

Bernd Hartke*

Institut für Physikalische Chemie, Christian-Albrechts-Universität, Olshausenstraße 40, 24098 Kiel, Germany

We offer empirical evidence that certain classes of frequently used benchmark functions for global optimization are not enough of a challenge for modern implementations of genetic algorithms, since they neither allow for a useful discrimination between different variants of genetic operators nor exhibit a dimensionality scaling resembling that of real-world problems, in particular including that of global structure optimization of atomic and molecular clusters. The latter properties seem to be simulated better by two other types of benchmark functions. One type is designed to be deceptive, exemplified here by Lunacek's function. The other type offers additional advantages of markedly increased complexity and of broad tunability in search space characteristics. For the latter type, we use an implementation based on randomly distributed Gaussians.

keywords: global optimization, genetic algorithms, benchmark functions, dimensionality scaling, crossover operators

*hartke@phc.uni-kiel.de

1. Introduction

Global optimization has a lot of real-world applications, both of discrete and non-discrete nature. Among them are chemical applications such as structure optimization of molecules and clusters, engineering problems such as component design, logistics problems like scheduling and routing, and many others. Despite the typical practical finding that a general global optimization algorithm usually is much less efficient than specific versions tuned to the problem at hand, it is still of interest to gauge the baseline performance of a global optimization scheme using benchmark problems. Even in the most recent examples of such tests [1–6] (selected at random from this year's literature), it is customary to employ certain standard benchmark functions, with the implicit (but untested) assumption that the difficulty of these benchmark functions roughly matches that of real-world applications. Some of these benchmark functions even are advertised as particularly challenging.

We have developed evolutionary algorithm (EA) based global optimization strategies in the challenging, real-life area of atomic and molecular cluster structure optimization [7–11]. When we apply our algorithms to those traditional, abstract benchmark functions, however, neither of those two claims (challenge, and similarity to real-world applications) stands up. In fact, similar suspicions have been voiced earlier. For example, already in 1996 Whitley *et al.* [12] argued that many of the standard benchmark functions should be relatively easily solvable due to inherent characteristics like symmetry and separability. Some functions even appeared to get easier as the dimensionality of the function increases. Nevertheless, as the citations in the previous paragraph indicate, the same set of traditional benchmark functions continues to be used indiscriminately to the present day, by the majority of researchers in various fields. Therefore, with the present article, we address the need to re-emphasize those earlier findings from a practical point of view, add in other test functions, and extend the testing to higher dimensionality. In addition, we stress the conclusions that these traditional benchmark problems appear to be too simple to allow for meaningful comparisons between algorithms or implementation details, and that they do not allow conclusions about the performance of global optimization algorithms in real-life situations. We show that the latter is achieved better when using different kinds of benchmark functions.

In these contexts, theoretical considerations often focus on classifying a problem as N or NP [13,14], or on evaluation of marginally different parameter representations [12,15] or hybrid combinations of known test functions [12]. Quite independent of such problem classifications and algorithm characteristics, however, in most real-world applications scaling with problem dimension (i.e., number of parameters to be optimized) plays a pivotal role. Chemical structure optimization of clusters is an obvious example: Of central practical importance are phenomena like cluster aggregation and fragmentation, or the dependence of properties on cluster size, while isolating a single cluster size is a formidable experimental challenge. Therefore, one does not study a single cluster size but tries to systematically study a range of clus-

ters [8, 9, 16–18], only limited by the maximum computing capacity one has. It is obvious that smallest decreases in scaling (e.g. from $\mathcal{O}(N^{3.5})$ to $\mathcal{O}(N^3)$) may allow for significantly larger cluster sizes to be studied.

The problem dimensionality scaling of the number of global optimization steps needed is of course linked to features of the global optimization algorithm. For evolutionary algorithms, this includes crossover and mutation operators, possible local optimization steps and problem-specifically tuned additional operators. We would like to present here latest results of some standard benchmark functions in the context of our recently developed framework for the evolutionary global optimization of chemical problems, OGOLEM [11]. By screening the needed amount of global minimizing steps for solving up to 500 (in one case 1000) dimensional benchmark functions, we obtain the scaling of different crossover operators with the dimensionality of these functions. Additionally, we compare runs with and without local optimization steps to investigate the effect of gradient based minimization on the scaling. Last but not least, we compare the performance on these standard benchmark functions with that on different kinds of benchmark function that apparently present more serious challenges, coming closer to real-world problems in some respects.

2. Methods and Techniques

All calculations mentioned in this paper were carried out using our OGOLEM framework [11] written in Java with SMP parallelism enabled. Since differing concurrency conditions can obviously have an impact on the benchmarking results, all calculations were carried out with three concurrent threads.

OGOLEM is using a genetic algorithm (GA), loosely based on the standard GA proposed in Ref. [19] but differing in the treatment of the genetic population. Instead of a classical generation based global optimization scheme, a pool algorithm [10] is used. This has the advantage of both eliminating serial bottlenecks and reducing the number of tunables since e.g. elitism is build-in and no rate needs to be specified.

Tunables remaining with this approach are mentioned in table 1 with values kept constant in the benchmark runs.

The genetic operators are based upon a real number representation of the parameters. Within the crossover operator, the cutting is genotype based. Different crossover operators used below differ only in the numbers and positions of cuts through the genome. The positions are defined by randomized number(s) either being linearly distributed or Gaussian distributed[a], with the maximum of the Gaussian being in the middle of the genome and with the resulting Gaussian-distributed random numbers multiplied with 0.3 to make the distribution sharper. In Tab. 2 the used algorithms are summarized and explained.

Mutation and mating are the same for all algorithms and tests. The mutation

[a]We are using in both cases the standard PRNG provided by the Java standard.

Tunable (pool approach)	*Representation (generation approach)*	*value*
pool size	generation size	1000
global optimization steps	generation size times number of generations	till minimum is reached
fitness diversity	threshhold which individuals are considered to be same	$1 \cdot 10^{-8}$

Table 1. Tunables in the pool algorithm and their value during the benchmark.

Algorithm	*Crossing*	*Number of Crossings*	*Crossing Point*
Holland	no	0	N/A
Germany	yes	1	Gaussian
Portugal:1	yes	1	linear
Portugal:3	yes	3	linear
Portugal:5	yes	5	linear
Portugal:7	yes	7	linear

Table 2. Definition of the used algorithms.

is a standard one-point genotype mutation with a probability of 5%. The actual gene to be mutated is chosen with a linearly distributed random number and replaced with a random number in between allowed borders specific to every function/application/parameter.

Mating is accomplished by choosing two parents out of the genetic pool. The mother is chosen purely randomly, whilst the father is chosen based on a fitness criterion. All structures in the pool are ranked by their fitness, a Gaussian distributed random number shaped with the factor 0.1 is chosen with its absolute value mapped to the rank in the pool.

If a local optimization step is carried out, it is a standard BFGS as described e.g. in Ref. [20] with very tight convergence criteria (e.g. $1 \cdot 10^{-8}$ in fitness). The needed gradients are analytical in all cases, with the exception of Schaffer's and Lunacek's function where a numerical gradient is calculated with a two-point stencil.

Once the crossing, mutation and (if applicable) local optimization steps have been carried out on both children, only the fitter one will be returned to the pool. This fitter child will actually be added to the pool if it has a lower function value than the individual with the highest function value in the pool and does not violate the fitness diversity criterion. The fitness diversity is a measure to avoid premature convergence of the pool. Additionally, it promotes exploration by maintaining a minimum level of structural diversity, indirectly controlled via fitness values. For

example, by assuming that two individuals with the same fitness (within a threshold) are same, it eliminates duplicates. Since the pool size stays constant, the worst individual will be dropped automatically, keeping the pool dynamically updated.

For the benchmarking, we do not measure timings since they are of course dependent upon convergence criteria of the local optimization and potentially even on the exact algorithm used for the local optimization (e.g. BFSG vs. CG). Therefore, our benchmarking procedure measures the number of global optimization steps where a step is defined to consist of mating, crossing, mutation and (if applicable) local optimization. The second difficulty is to define when the global optimum is found. We are using the function value as a criterion, trying to minimze the amount of bias one introduces by any measure. It should be explicitly noted here that within the benchmarking of a given test function, this value stays constant in all dimensionalities which should in principle increase the scaling, again minimizing the amount of (positive) bias introduced.

3. Standard benchmark functions

Any approach towards global optimization should be validated with a set of published benchmark functions and/or problems. In the area of benchmark functions a broad range of published test functions exists, designed to stress different parts of a global optimization algorithm. Among the most popular ones are Schwefel's, Rastrigin's, Ackley's and Schaffer's functions. They have the strength of an analytical expression with a known global minimum and, in the case of all but the last function, they are extendable to arbitrary dimensionality allowing for scaling investigations. Contrary to assumptions made frequently, however, these benchmark functions do not allow to discriminate between algorithmic variations in the global optimization, nor do they give a true impression of the difficulty to be expected in real-life applications, as we will demonstrate in the following subsections.

3.1. *Ackley's Function*

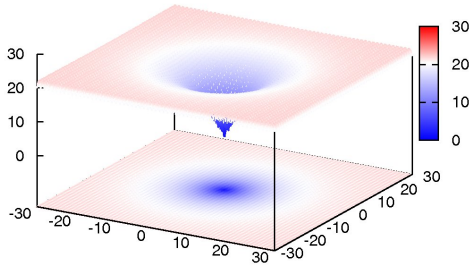
Ackley's function has been first published in Ref. [21] and has been extended to arbitrary dimensionality in Ref. [22] It is of the form

$$f(\vec{x}) = -20 \cdot \exp\left[-0.2\sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n} x_i^2}\right] - \exp\left[\frac{1}{n}\cdot\sum_{i=1}^{n}\cos(2\pi x_i)\right] + 20 + e^1 \qquad (1)$$

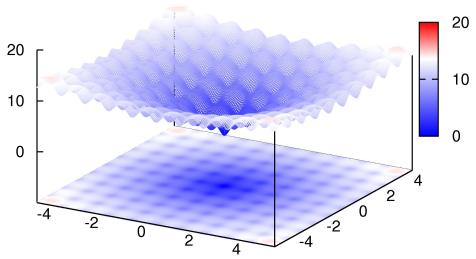
with the global minimum at $x_i = 0.0$. We considered this to be a relatively trivial function due to its shape consisting of a single funnel (see fig. 1). Nevertheless, this function type potentially has relevance for real-world applications since e.g. the free energy hypersurface of proteins is considered to be of similar, yet less symmetric, shape.

The initial randomized points were drawn from the interval

$$-32.768 \leq x_i \leq 32.768 \qquad (2)$$



(a) Full search space



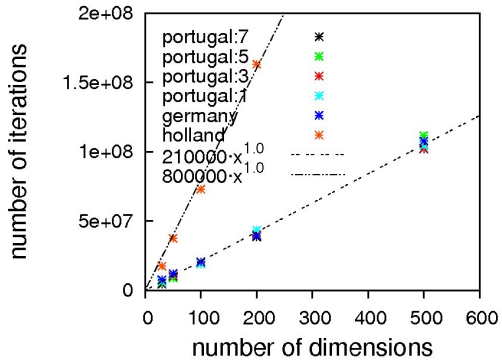
(b) Fine structure

Fig. 1. 2D plot of Ackleys function.

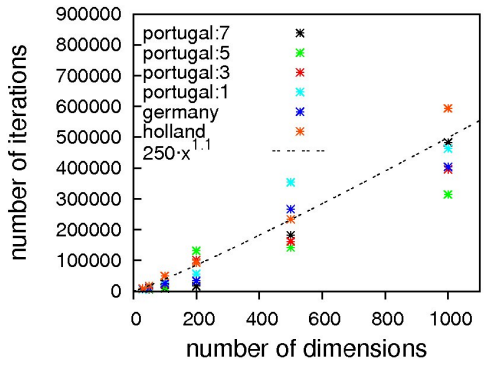
for all x_i, which is to our knowledge the normal benchmark procedure for this function.

As can be seen from Fig. 2, without local optimization steps the choice of genotype operator makes almost no difference; all cases exhibit excellent linear scaling. The only deviating case is the mutation-only algorithm, *Holland*, which has a higher prefactor in comparison to the other algorithms but still exhibits the same (linear) scaling. With local optimization enabled, the results do vary more. Results up to 500 dimensions do not allow for a concise statement on the superiority of a specific crossover operator. We therefore extended the benchmarking range up to one thousand dimensions, hoping for a clearer picture. It should be noted here that on a standard contemporary FreeBSD quadcore workstation (using three of the four cores) running openJRE6, these calculations took around 4.25 hours each, demonstrating the good performance of our framework.

Even with the extended benchmarking range, no concise picture could be obtained. This most likely means that none of the used algorithms is clearly superior



(a)



(b)

Fig. 2. Scaling results for Ackley's function. a) without, b) with local optimization.

to the others, in agreement with the results gained from the runs without local optimization. The only difference remaining is that the prefactor of the mutation-only algorithm is reduced to equality with the crossover algorithms. In general, the cases with enabled local optimization do increase the scaling slightly from 1.0 to 1.1, which is still excellent.

Still, these results obviously allow for the conclusion that Ackley's benchmark function can be considered to be of trivial difficulty since linear scaling is achievable already without local optimization.

3.2. *Rastrigin's Function*

Rastrigin's function [23,24] does have fewer minima within the defined search space of

$$-5.12 \leq x_i \leq 5.12 \tag{3}$$

but its overall shape is flatter than Ackley's function which should complicate the general convergence towards the global optimum at $x_i = 0.0$.

We defined Rastrigin's function with an additional harmonic potential outside the search space to force the solution to stay within those boundaries when using unrestricted local optimization steps.

$$f(x_0 ... x_n) = 10 \cdot n + \sum_{i=0}^{n} \begin{cases} x_i > 5.12 \vee x_i < -5.12 : & 10 \cdot x_i^2 \\ -5.12 \leq x_i \leq 5.12 : & x_i^2 - 10 \cdot \cos(2\pi x_i) \end{cases} \tag{4}$$

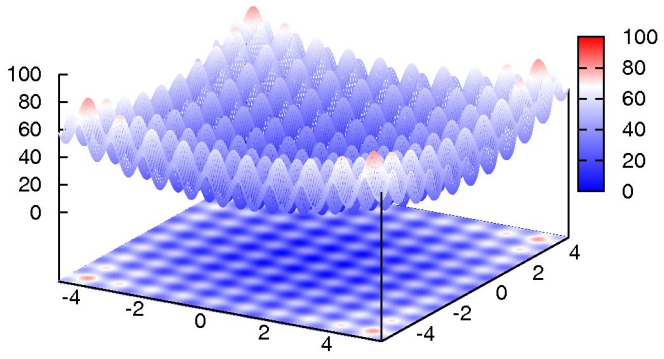


Fig. 3. 2D plot of Rastrigin's function

As can be seen from Fig. 4, the scaling is excellent with all tested crossing operators; *Holland* again being the easily rationalizable exception, in the case without local optimization. In the case with local optimization, a contrasting picture can be seen.

The scaling once more does not deviate much between the different algorithms with the prefactor making the major difference. Interestingly, by far the best prefactor and scaling is obtained with *Holland*. Probably, the rather non-intrusive behaviour of a mutation-only operator fits this problem best since it provides a better short-range exploration than any of the crossing algorithms. We assume the extremely high number of global optimization steps (also in comparison to the nonlocopt case) to be due to a repeated finding of the same, non-optimal minima. Inclusion of *taboo-search* features [25] into our algorithm might be of help for real-world problems of such a type, not reducing the number of global optimization steps but the amount of time spent in local optimizations rediscovering already known minima.

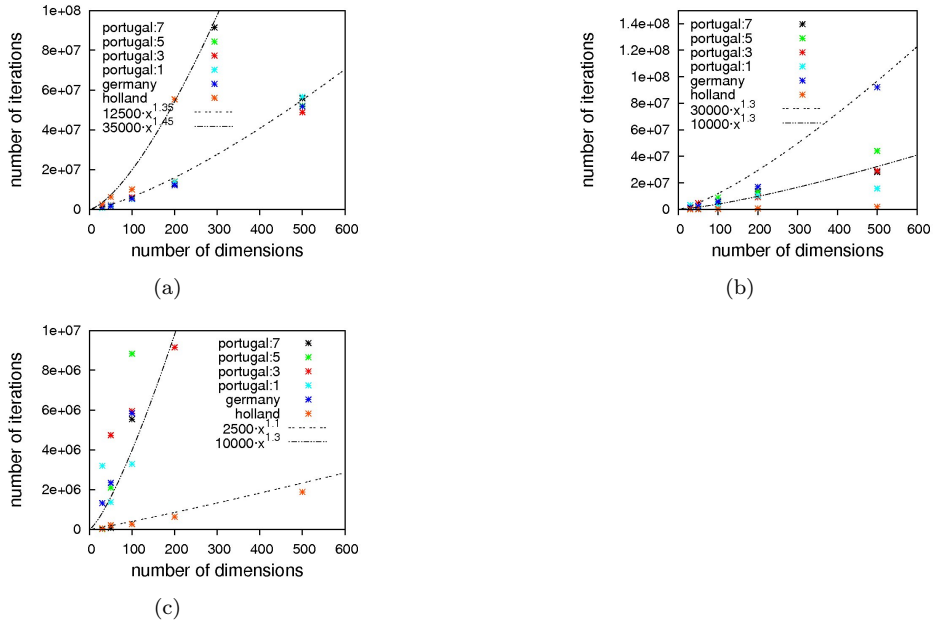


Fig. 4. Scaling results for Rastrigin's function. a) without, b) with local optimization and c) zoom with local optimization.

Nevertheless, Rastrigins function can be solved with almost linear scaling both with and without local optimization, rendering it rather unuseful as a benchmark function.

3.3. *Schwefel's Function*

In comparision to Rastrigin's function, Schwefel's function [26] adds the difficulty of being less symmetric and having the global minimum at the edge of the search space

$$-500.0 \leq x_i \leq 500.0 \tag{5}$$

at position $x_i = 420.9687$. Additionally, there is no overall, guiding slope towards the global minimum like in Ackley's, or less extreme, in Rastrigin's function.

Again, we added an harmonic potential around the search space

$$f(x_0 ... x_n) = 418.9829 \cdot n + \sum_{i=0}^{n} \begin{cases} x_i > 500 \vee x_i < -500: & +0.02 \cdot x_i^2 \\ -500 \leq x_i \leq 500: & -x_i \cdot \sin\left(\sqrt{|x_i|}\right) \end{cases} \tag{6}$$

since otherwise our unrestricted local optimization finds lower lying minima outside the principal borders.

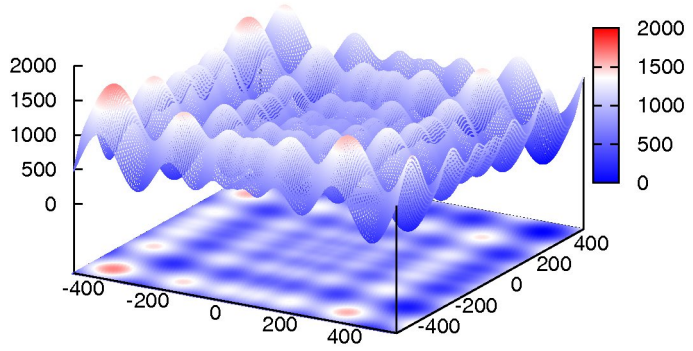


Fig. 5. 2D plot of Schwefels function

As can be seen from Fig. 6, sub-quadratic scaling can be achieved with and without local optimization. Once more, without local optimization the non-crossing algorithm has a higher prefactor but the same scaling as the others, which is equalized when turning on local optimizations. In this particular case, the usage of local optimizations reduces the scaling slightly, from 1.5 to 1.43.

Again, we must come to the conclusion that a sub-quadratic scaling is far better than what we would expect to obtain for real-world problems, making Schwefel's function not a good test for algorithms designed to solve the latter.

3.4. *Schaffer's Function*

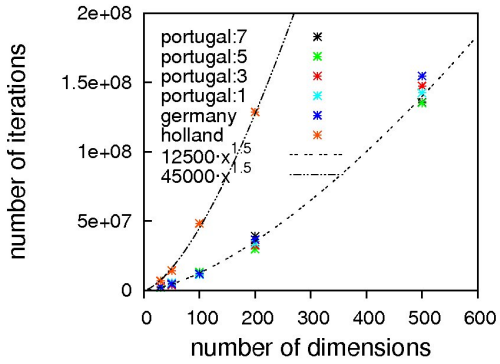
For completeness, we would like to present some non-scaling results using Schaffer's F6 function as a benchmark:

$$f(x,y) = 0.5 + \frac{\sin^2(\sqrt{x^2+y^2}) - 0.5}{\left[1 + 0.001 \cdot (x^2+y^2)\right]^2} \tag{7}$$

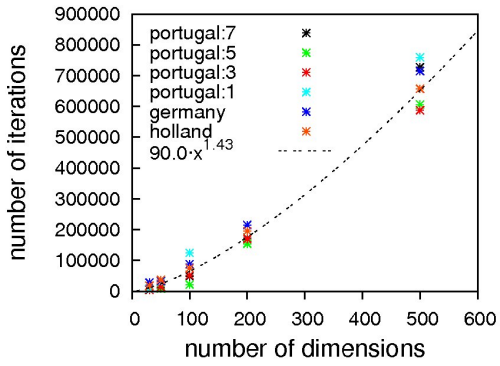
As can be seen in Fig. 7, the difficulty in this function is that the size of the potential maxima that need to be overcome to get to a minimum increases the closer one gets to the global minimum.

In Tab. 3, results can be found which were obtained with *Holland* and with the one-point crossover operators (obviously, with a real-number encoded genotype approach, not more cuts can take place for a two-dimensional function). The results are outcomes of three successive runs which is sufficient to obtain a general picture of the trend.

The difference between the *Portugal* and *Germany* approach in this very special case is that *Germany* in contrast to *Portugal* can also yield a crossover point before



(a)



(b)

Fig. 6. Scaling results for Schwefel's function. a) without, b) with local optimization.

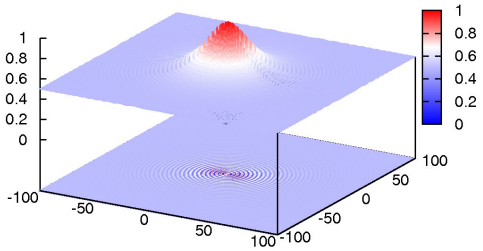
the first number, effectively reducing it to a partial non-crossing approach.

Interestingly, we do see converse tendencies between the case with and without local optimization. We see a clear preference of the *Germany* crossover operator over *Portugal* without local optimization. Taking the results of the non-crossing operator into account, it seems clear that without local optimization too much crossing is harmful in terms of convergence to the global minimum. With local optimization enabled, these differences disappear, sometimes even allowing the global minimum to be found in the initial (and therefore never crossed) pool of solutions.

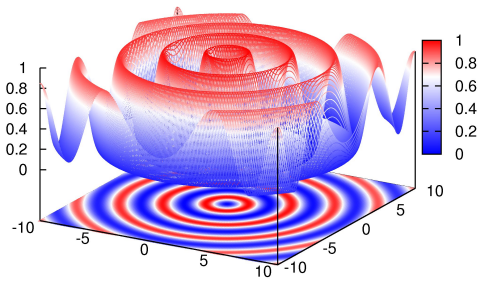
Although Schaffer's function shows an impressive difficulty for a two-dimensional function, it can still be easily solved with and without local optimization.

3.5. *Scatter of the benchmarking results*

Obviously any stochastic approach is difficult to benchmark in a reliable manner. Therefore, we would like to discuss the scatter of the benchmarking results. We will try to approximate possible deviations for every crossover operator used with and



(a) Full search space



(b) Fine structure

Fig. 7. Plot of Schaffers F6 function

without local optimization for a two hundred dimensional Ackley's function. For this, we present in Tab. 4 results from ten runs per crossover operator used.

Of course, the results do not and cannot take all or the maximum possible deviations into account since in principle there should be a probability distribution from a single iteration up to infinity which can only be captured adequately by an infinite amount of successive runs. Nevertheless, ten successive runs can be considered to give a rough idea of the location of the maximum.

The impression gained in the previous sections, namely that the exact nature of the genotype operator does not seem to make a difference when using local optimization, holds true also with enhanced statistics for this case. Similarly, the differences seen in the runs without local optimizations between the crossing operators and the non-crossing operator, *Holland*, remain also when averaging over more runs.

This allows for the conclusion that the results presented in the previous sections are giving a reasonably accurate picture, despite of course suffering from the inherent uncertainty in all stochastic methods which cannot be circumvented.

Algorithm	*w/ locopt*	*w/o locopt*
Holland	487	189728
	2112	151450
	1794	129123
Germany	4450	172470
	4392	12168
	1233	171360
Portugal:1	933	421398
	1130	496562
	3441	603775

Table 3. Different results for Schaffers F6 function with one-point and zero-point crossover operators.

Algorithm	*Local opt.*	*Maximum*	*(% dev.)*	*Minimum*	*(% dev.)*	*Average*
Holland	w/	8991	18.5	6113	19.5	7590
	w/o	10323905	13.0	7805543	14.5	9132168
Germany	w/	11289	57.1	4758	33.8	7186
	w/o	4312566	28.1	2704586	19.6	3365305
Portugal:1	w/	20464	126.8	5191	42.5	9023
	w/o	4748780	36.7	2095005	39.7	3473084
Portugal:3	w/	7172	27.6	4704	16.3	5621
	w/o	4640199	45.5	2328333	27.0	3189287
Portugal:5	w/	6268	20.2	4873	6.6	5215
	w/o	4745643	39.4	2160887	36.5	3403532
Portugal:7	w/	6510	15.3	4858	14.0	5646
	w/o	4221855	29.9	2541700	21.8	3249622

Table 4. Number of global optimization iterations from ten successive runs on the 200D Ackley function.

Upon closer examination of the data in Table 4, some seemingly systematic trends can be observed, calling for speculative explanations. A general tendency observed is the reduced spread when local optimization is turned off, probably because a higher diversity can be maintained providing a better and more reliable convergence. Another tendency is the reduced spread when more — or no — crossover points are used. In the case of more crossover points this can be explained with more crossover points causing bigger changes in each step; this improves search space coverage, which in turn makes the runs more reproducible. For the reduced scatter of the non-crossing operator, the explanation is obviously the opposite, since

this operator minimizes changes to the genome, allowing for a better close-range exploration.

Despite of these interesting observations, we refrain from further analysis since this would lead us outside of the scope of the present article.

4. Gaussian benchmark class

To our experience from the global optimization of chemical systems, real-world problems are considerably more challenging than the benchmark functions described above. For example, in the case of the relatively trivial Lennard-Jones (LJ) clusters, the best scaling we could reach is cubic [8]. Therefore, we feel a need for benchmarks with a difficulty more closely resembling real-world problems.

Defining new benchmark functions is of course not trivial since they should fulfill certain criteria.

(1) Not trivial to solve.
(2) Easy to extend to higher dimensions causing higher difficulty.
(3) Possibility to define an analytical gradient for gradient based methods.
(4) Of multimodal nature with a single and known global minimum.

To have better control over these criteria when generating benchmark functions, a few "search landscape generators" have been proposed in recent years [27–29]. The simplest and most flexible of these is the one based on randomly distributed Gaussians [27]. For convenience, we have used our own implementation of this concept, abbreviated *GRUNGE* (GRUNGE: Randomized, UNcorrelated Gaussian Extrema), defined and discussed in the following. We would like to emphasize already at this point that our intention in using *GRUNGE* is not to re-iterate known results from Ref. [27] and similar work, but to directly contrast the OGOLEM behavior displayed in section 3 with its different behavior in the *GRUNGE* benchmark. This shows strikingly that the rather uniform results in section 3 are not a feature of OGOLEM but rather a defect of that benchmark function class.

We define a function as a set of randomized Gaussians

$$f(x_0 ... x_M) = \sum_{i=0}^{N} \xi_i \cdot \exp\left[-\zeta_i \cdot \sum_{j=0}^{M} (x_j - \kappa_j)^2 \right] \tag{8}$$

with the random numbers ξ_i, ζ_i and κ_i being the weight, width and position of the i-th Gaussian in M-dimensional space. As can be easily seen, this class of benchmarking problems provides (besides the search space size) two degrees of freedom. One is the number N of randomized Gaussians in the search space and M being the dimensionality of the Gaussians.[b] More subtly, there is also a connection between these two characteristics and the Gaussian widths within the maximal coordinate

[b]As a side note, we write *GRUNGE(M,N)*, e.g. for 2000 gaussians in a ten dimensional space *GRUNGE[10,2000]*.

interval (i.e., the Gaussian density). With proper choices of these numbers, one can smoothly tune such a benchmark function between the two extremes of a "mountain range" (many overlapping Gaussians) and a "golf course" (isolated Gaussians with large flat patches in-between).

Functions in this class of benchmarks are not easy to solve, easily extendable in dimensionality and — through the use of Gaussians — the definition of an analytical gradient is trivial. The only problem remaining is to pre-determine the position and depth of the global minimum. Here, other benchmark function generators like, e.g., the polynomial ones proposed by Gaviano *et al.* [29] and Locatelli *et al.* [28], allow for more control, but at the price of a more uniform overall features of the generated test functions, which is exactly what we want to avoid. We also do not want to enforce a known global minimum by introducing a single, dominating Gaussian with excessively large weight by hand. Thus, the only remaining possibility is to define a fine grid over the search space and to run local optimizations starting at every grid point, to obtain a complete enumeration of all minima within the search space. Due to the simple functional form and to the availability of an analytical gradient, this is a realistic proposition for moderately-dimensioned examples (10D) containing a sufficiently great number of sufficiently wide Gaussians.

In contrast to the traditional benchmarks examined in the previous sections, the *GRUNGE* function is not deliberately designed to be deceptive, in any number of dimensions. Instead, due to the heavy use of random numbers in its definition, it does not contain any correlations whatsoever. To our experience, this feature makes *GRUNGE* benchmarks much harder than any of the traditional benchmarks. We cannot offer formal proofs at this stage, but our distinct impression from many years of global optimization experience is that realistic problems tend to fall in-between these two extremes, being harder than traditional benchmarks but less difficult (less uncorrelated) than *GRUNGE*.

Obviously, a full exploration of the randomize Gaussians set of benchmark functions requires an exclusive and extensive study, which has already been started by others [27]. As already mentioned above, our sole intention here is to provide a contrast to the OGOLEM behavior noted in section 3. To this end, we present results based on solving a ten-dimensional *GRUNGE* benchmark with 2000 gaussians (*GRUNGE[10,2000]*) within a search space of

$$0.0 \leq x_i \leq 10.0 \tag{9}$$

with local optimization enabled.

As can be seen from the results in Tab. 5, the average of three independent runs of all algorithms yields results within the same order of magnitude. When comparing the numbers in Tab. 5 with the results given above for the conventional benchmark functions, e.g., with those in Tab. 4, one should remember the differences in dimensionality: Here we are dealing with a 10-dimensional problem with 2000 minima, whereas in Tab. 4 we reported the performance on the 200-dimensional Ackley function with the number of minima being several orders of magnitude

Algorithm	*Run 1*	*Run 2*	*Run 3*	*Average*
Holland	5725	7676	6228	6543
Germany	2390	153	5390	2644
Portugal:1	3145	7637	4077	4953
Portugal:3	1879	2575	1339	1931
Portugal:5	2441	1647	4888	2992
Portugal:7	5533	369	6322	4074

Table 5. GRUNGE[10,2000] benchmarking results with local optimization steps.

higher. This gives an indication of what we experience as a big difference in difficulty.

It should be noted, however, that in two cases the global optimum could be found within the locally optimized initial parameter sets. While this demonstrates once more that a randomly distributed initial parameter set can have an extraordinary fitness, it also indicates that higher dimensional *GRUNGE* benchmarks are necessary to better emulate real-world problems.

We also did some tests without local optimization, showing that the *GRUNGE* benchmark with our randomly generated Gaussians is extremely difficult to solve without local optimization, requiring almost 19 million global optimization steps with *Portugal:3*. We suspect that this level of difficulty is related to inherent features of the *GRUNGE* benchmark (e.g., to the completely missing correlation between the locations and depths of the minima) but also to features of the specific GRUNGE[10,2000] incarnation used here (i.e., this particular Gaussian distribution and density), but decide to leave this sidetrack at this point.

5. Lunacek's function

Lunacek's function [30], also known as the bi- or double-Rastrigin function, is a hybrid function consisting of a Rastrigin and a double-sphere part and is designed to model the double-funnel character of some difficult LJ cases, in particular LJ_{38}.

$$f(x_0 ... x_n) = \min\left(\left\{\sum_i^N (x_i - \mu_1)^2\right\}, \left\{d \cdot N + s \cdot \sum_i^N (x_i - \mu_2)^2\right\}\right) + 10\sum_i^N (1 - \cos 2\pi(x_i - \mu_1)) \quad (10)$$

$$\mu_2 = -\sqrt{\frac{\mu_1^2 - d}{s}} \quad (11)$$

This indicates that there is an interest in developing benchmark functions of higher difficulty, and indeed the developed function provides an interesting level of difficulty, as we show below. Nevertheless, we would like to dispute the notion that it resembles certain real-world problems and the source of their difficulty. Specifically, Lunacek *et al.* claim that the global optimization of homogeneous LJ clusters is one of the most important applications of global optimization in the field of computational chemistry. Furthermore, they claim that the most difficult instances of

the LJ problem possess a double-funnel landscape. The former claim is a rather biased view and promotes the importance of homogenous LJ clusters from a mere benchmark system to a hot-spot of current reasearch. As the broad literature on global cluster structure optimization documents (cf. reviews [11, 31–33] and references cited therein), current challenges in this field rather are directed towards additional complications in real-life applications, e.g., how to taylor search steps to dual-level challenges of intra- and intermolecular conformational search in clusters of flexible molecules, or how to reconcile the vast number of necessary function evaluations with their excessive cost at the ab-initio quantum chemistry level. In terms of search difficulty, the homogeneous LJ case is now recognized as rather easy for most cluster sizes, interspersed with a few more challenging problem realizations at certain sizes, with LJ_{38} being the smallest and hence the simplest of them. This connects to the second claim by Lunacek *et al.*, namely that the difficulty of LJ_{38} arises from the double-funnel shape of its search landscape, which is captured by their test function design. It is indeed tempting to conclude from the disconnectivity graph analysis by Doye, Miller and Wales [34] that there are two funnels, a narrow one containing the fcc global minimum, separated by a high barrier from the broad but less deep one containing all the icosahedral minima. Even if this were true (to our knowledge, such a neat separation of the two structural types in search space has not been shown), it would give rise to only two funnels in a 108-dimensional search space, which is not necessarily an overwhelming challenge and also not quite the same as what eq. 10 offers.

Lunacek's function is specifically designed to poison global optimization strategies working with bigger population sizes. This is achieved through the double sphere contribution which constructs in every dimension a fake minimum, e.g., when using the settings $s = 0.7$, $d = 1.0$, with the optimal minimum located at $x_i = 2.5$. As Lunacek *et al.* have proven in their initial publication, the function is very efficient in doing this. This is an observation that we can support from some tests on 30-dimensional cases.

Clearly, this is a markedly different behavior than that observed above for Ackley's, Rastrigin's, Schwefel's or Schaffer's functions, coming closer to what we experienced in tough application cases. Therefore, it is not surprising that additional measures developed there are also of some help here. One possibility is to adopt a niching strategy, similar to what was applied to reduce the solution expense for the tough cases of homogenous LJ clusters to that of the simpler ones [8]. In essence, this ensures a minimum amount of diversity in the population, preventing premature collapse into a non-globally optimal solution.

The most trivial implementation of niching is to employ a static grid over search space and to allow only a certain maximal population per grid cell (MNIC, maximum number of individuals per grid cell). Already this trivial change allows the previously unsolvable function to be solved in 30 dimensions, as can be seen from Tab. 6. Solving higher dimensionalities (e.g. 100D) with this approach suffers again from dimensionality explosion, this time in the number of grid cells. Additionally, such

Dimensionality	*Static grid*	*Steps to solution*	*MNIC*
2	w/o	833	N/A
	w/	1063	250
5	w/o	2186	N/A
	w/	4184	100
10	w/o	392006	N/A
	w/	13922	100
15	w/o	459317	N/A
	w/	604954	50
20	w/o	not found	N/A
	w/	1153811	50
30	w/o	not found	N/A
	w/	2826707	20

Table 6. Exemplary benchmarking results of the Lunacek function. All results obtained with local optimization steps and the germany algorithm. Not found corresponds to more than 10 million unsuccessful global optimization steps. MNIC is the maximum number of individuals allowed per grid cell.

basic implementation truncates the exploitation abilities of the global optimization algorithm. This causes the algorithm with static niches to require more steps in those low dimensional cases where the problem can also be solved without. From our experience with LJ clusters, we expect more advanced niching strategies, for example dynamic grids, to prove useful with higher dimensionalities and render the function even less difficult.

As it seems to be common wisdom in this area, applications do contain some degree of deceptiveness and different degrees of minima correlation, as well as different landscape characteristics, sampling all possibilities between golf courses and funnels. All of that can be captured with the GRUNGE setup. It thus offers all the necessary flexibility and simplicity, combined in a single function definition. The obvious downsides are the absence of a pre-defined global minimum (which can also be interpreted as a guarantee for avoiding biases towards it) and the need to tune many parameters to achieve a desired landscape shape.

6. Summary and Outlook

Scaling investigations for three different, standard benchmark functions have been presented, supplemented by performance tests on a fourth function. Using straightfoward GA techniques without problem-specific ingredients, the behavior we observe in all these cases is markedly different from what we observe upon applying the same techniques to real-world problems (often including system-specific additions): All benchmarks can be solved with sub-quadratic scaling, whereas in real-world applications we can get cubic scaling at best and often have to settle for much worse.

In addition, the benchmarks often do not allow for statistically significant conclusions regarding the performance of different crossover operators, nor for a decision on whether to include local optimization or not. Thus, overall, benchmarks of this type do not seem to fulfill their purpose of test beds with relevance for practical applications in global cluster structure optimization or similar areas.

We have contrasted the behavior on those traditional benchmark functions with that on two different types of functions. One type is the "landscape generator" class, shown here in a particularly simple realization, namely search landscapes generated by randomly distributed Gaussians. Varying Gaussian characteristics (depth, width, density, dimensionality), search space features can be tuned at will, on the full scale between a "mountain range" and a "golf course". In addition, since the constituting Gaussians are completely uncorrelated, the difficulty of this problem class is inherently larger than that of the traditional benchmark functions where the minima characteristics follow a simple rule by construction. This is strikingly reflected in our tests results on this benchmark class.

As yet another class of benchmark functions we have shown the deceptive type, designed to lead global optimization astray. Their difficulty can be diminished significantly by making the global search more sophisticated, in the case of population-based searches by ensuring a sufficient degree of diversity in the population. Given a sufficiently flexible setup, this benchmark class merely is a subclass of the landscape generators.

Further work will be required to confirm our suspicion that real-world problems often fall in-between the traditional, rather simple benchmark functions on the one hand and the less correlated, more deceptive ones on the other hand, both with respect to their search space characteristics and to the difficulty they present for global optimization algorithms. In any case, we hope to have shown convincingly that due to their simplicity functions from the traditional benchmark functions should not be used on their own, neither to aid global optimization algorithm development nor to judge performance.

Acknowledgement

The authors thank the North-German Supercomputing Alliance (HLRN) for a generous grant of computer time and would like to acknowledge the friendly and competent efforts of the HLRN support staff.

1. A. B. Ozer, Expert Syst. Appl. 37 (2010) 4632.
2. R. Thangaraja, M. Panta and A. Abraham, Appl. Math. Comput. 216 (2010) 532.
3. M. J. Hirsch, P. M. Pardalos and M. G. C. Resende, Eur. J. Oper. Res. 205 (2010) 507.
4. L. Zhao, F. Qian, Y. Yang, Y. Zeng and H. Su, Appl. Soft Comput. 10 (2010) 938.
5. Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren and J. J. Liang, Appl. Math. Comput. 216 (2010) 830.
6. `http://coco.gforge.inria.fr/`

7. B. Hartke, J. Phys. Chem. 97 (1993) 9973.
8. B. Hartke, J. Comput. Chem. 20 (1999) 1752.
9. B. Hartke, Z. Phys. Chem. 214 (2000) 1251.
10. B. Bandow and B. Hartke, J. Phys. Chem. A 110 (2006) 5809.
11. J. M. Dieterich and B. Hartke, Mol. Phys. 108 (2010) 279.
12. D. Whitley, S. Rana, J. Dzubera and K. E. Mathias, Artific. Intell. 85 (1996) 245.
13. L. T. Wille and J. Vennik, J. Phys. A: Math. Gen. 18 (1985) L419.
14. G. W. Greenwood, Z. Phys. Chem. 211 (1995) 105.
15. R. Salomon, Biosystems 39 (1996) 263.
16. H. Takeuchi, J. Chem. Inf. Model. 46 (2006) 2066.
17. H. Takeuchi, J. Chem. Inf. Model. 48 (2008) 2226.
18. R. L. Johnston, *Atomic and molecular clusters*, Taylor & Francis, London, 2002.
19. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, 1989.
20. W. H. Press, S. A. Teukolsky, W. T. Vetterlin and B. P. Flannery, *Numerical Recipes, 3rd edition*, Cambridge University Press, 2007.
21. D. H. Ackley, *A connectionist machine for genetic hillclimbing*, Kluwer Academic Publishers, 1987.
22. T. Bäck, *Evolutionary algorithms in theory and practice*, Oxford University Press, 1996.
23. A. Törn and A. Zilinskas, *Global Optimization, Lecture Notes in Computer Science No. 350*, Springer Verlag, Berlin, 1980.
24. H. Mühlenbein, D. Schomisch and J. Born, Parallel Computing, 17 (1991) 619.
25. S. Stepanenko and B. Engels, J. Phys. Chem. A 113 (2009) 11699.
26. H.-P. Schwefel, *Numerical optimization of computer models*, Wiley, 1981.
27. M. Gallagher and B. Yuan, IEEE Trans. Evol. Comput. 10 (2006) 590.
28. B. Addis and M. Locatelli, J. Glob. Optim. 38 (2007) 479.
29. M. Gaviano, D. E. Kvasov, D. Lera and Y. D. Sergeyev, ACM Trans. Math. Soft. 29 (2003) 469.
30. M. Lunacek, D. Whitley and A. Sutton, *The Impact of Global Structure on Search* in *Parallel problem solving from nature*, Springer Verlag, Berlin, 2008.
31. B. Hartke, Angew. Chem. Int. Ed. 41 (2002) 1468.
32. B. Hartke, Struct. Bond. 110 (2004) 33.
33. B. Hartke, "Global Optimization", in: "Computational Molecular Science", P. R. Schreiner, W. D. Allen, M. Orozco, W. Thiel and P. Willett (Eds.), Wiley Interdisciplinary Reviews, submitted (25 Nov 2010).
34. J. P. K. Doye, M. A. Miller and D. J. Wales, J. Chem. Phys. 111 (1999) 8417.

CHAPTER

5

Highly Mixed LJ Clusters

When I am working on a problem, I never think about beauty. I think only of how to solve the problem. But when I have finished, if the solution is not beautiful, I know it is wrong.

R. Buckminster Fuller

5.1 Scope of the Project

Homogenous clusters of the noble gases are a standard benchmark for the global optimization of chemical structures [54, 117]. Global minima of those clusters are literature-known up to 1000 atoms [117]. Comparatively little work has been done on the mixed clusters which introduce new challenges [118, 123]. The search space is dramatically enlarged with the inclusion of more atom types and the accuracy requirements for the potential increase drastically.

Following an approach similar to Schwerdtfeger *et al.* [146], the standard LJ(6,12,6) potential with Lorentz-Berthelot mixing rules has been discarded in favor of a more flexible LJ(6,16,2) potential parametrized to reproduce highest accuracy *ab initio* calculations specific for every possible atomic pair.

The obtained parameters can then be used for studying structural effects introduced by including different atom types. The project focussed on one of the famous magic numbers in the homoatomic case, LJ_{38}. A stable *fcc*-type minimum could be obtained up to a ternary composition, and points of structural transition from *fcc* to an icosahedral pattern could be located.

5.2 Own Contribution

The project required highly exact *ab initio* reference calculations at the CCSD(T)/aug-cc-pV5Z level of theory which have been carried out by the first author.

Also, the implementation of a reparametrizable LJ(6,16,2) force field has been carried out and new parameters where globally fitted.

With the so obtained set of parameters, a set of representative clusters in the smaller size regime was studied. These included quinary LJ_{19} and LJ_{55} clusters as well as binary and ternary LJ_{38} clusters. These studies required the implementation of a new genetic algorithm implementation focused on an optimal combination of *genotype* and *phenotype* cuts as well as explicit exchange of building blocks by means of an *XChange* operator. These implementations and the corresponding calculations have also been carried out

by the first author.

5.3 Publication

Authors Johannes M. Dieterich and Bernd Hartke

Institut für Physikalische Chemie
Christian-Albrechts-Universität
Olshausenstraße 40
24098 Kiel, Germany

Title Composition-induced structural transitions in mixed Lennard-Jones clusters: global reparametrization and optimization

Submitted July 13, 2010

Accepted October 22, 2010

Publication Data *J. Comput. Chem.*, DOI: 10.1002/jcc.21721 (2010)

5.4 Additional Information

5.4.1 Notes on the Parameter Fit

Any polynomial fit potentially suffers from unphysical properties outside the fitting regime. This is also true for the approach followed within this project. At short pair distances, the parameters either tend towards positive or negative infinity, depending upon the specific pair.

Therefore, it is of utmost importance to compensate for this behavior by means of cutoff potentials. The approach taken in this work was to cut the potential at interatomic distances smaller than

$$r_{ij} = 1.2 \cdot (R_i + R_j) \tag{5.1}$$

where r_{ij} is the interatomic distance, R_i and R_j are the atomar radii for atoms i and j, respectively. The choice for this cutoff is equivalent to the distance criterion for the collision detection and is valid for all possible atomic pairs.

To save CPU cycles, another cutoff was introduced for interatomic distances larger than

$$r_{ij} = 7.0 \cdot (R_i + R_j) \tag{5.2}$$

where the potential is numerically zero.

The exact parameters for the functional expression of the derived LJ(6,16,2) potential

$$v_{ij} = 4\epsilon \cdot \sum_{i=6,i+2}^{16} sign(\sigma_i) \cdot \left(\frac{\sigma_i}{r_{ij}}\right)^i \tag{5.3}$$

for all possible pairs of noble gases can be found in tabs. 5.1-5.4.

Parameter	*He-He*	*He-Ne*	*He-Ar*	*He-Kr*
ϵ	1.778929451E-7	1.469297288E-7	-1.172489205E-6	4.070080152E-6
σ_6	-12.496202868	-13.570572511	11.889302401	-10.265859438
σ_8	-1.925391417	-9.367199542	5.451846984	-3.353013167
σ_{10}	8.665655532	8.426529382	-6.431974542	7.204411443
σ_{12}	3.893157020	-1.816315242	-6.500180660	-5.455125357
σ_{14}	4.986864076	7.708391041	-7.848040176	7.520689417
σ_{16}	-3.404616901	-6.005919647	-6.356001142	6.519035785

Table 5.1: LJ(6,16,2) parameters for noble gas pair potentials. All values in atomic units. Part I.

Parameter	*He-Xe*	*Ne-Ne*	*Ne-Ar*	*Ne-Kr*
ϵ	-5.175961322E-7	2.192908250E-7	6.434512112E-7	-2.866177519E-4
σ_6	14.497222733	-14.632993194	7.757331805	5.772817574
σ_8	12.369831986	-5.418814208	-13.999455706	-4.244907899
σ_{10}	-5.639583850	-7.946421006	11.738207465	-1.525435298
σ_{12}	-10.659321565	-6.184445450	7.813415979	4.297373372
σ_{14}	-7.556208528	8.174293243	-4.448904828	-5.948347650
σ_{16}	8.061788716	-6.100169226	-6.734122649	-5.038131185

Table 5.2: LJ(6,16,2) parameters for noble gas pair potentials. All values in atomic units. Part II.

Parameter	*Ne-Xe*	*Ar-Ar*	*Ar-Kr*	*Ar-Xe*
ϵ	5.985608529E-6	-6.396663093E-6	-6.514462846E-5	-4.444732080E-4
σ_6	6.873474657	9.353108372	8.905630027	-3.343458418
σ_8	-12.005186466	12.152246592	-5.012205283	8.266031615
σ_{10}	10.653080682	-10.671409062	-0.499347639	-7.980132808
σ_{12}	0.646205236	-7.435515833	2.514892244	-4.203318268
σ_{14}	-5.652448480	-6.474219149	-5.791460326	-5.066684759
σ_{16}	-6.274563098	6.317082155	-7.341596346	4.857595596

Table 5.3: LJ(6,16,2) parameters for noble gas pair potentials. All values in atomic units. Part III.

Parameter	*Kr-Kr*	*Kr-Xe*	*Xe-Xe*
ϵ	5.276000803E-6	9.417792939E-4	5.286477052E-4
σ_6	4.236290422	-6.672957783	1.629720121
σ_8	-14.195544965	-2.464994984	-9.076770216
σ_{10}	12.246376221	-2.675373770	8.742686199
σ_{12}	7.455198829	6.749548666	-4.501137419
σ_{14}	0.480894238	6.213825350	-0.686695832
σ_{16}	5.428879220	5.300000263	-6.372758498

Table 5.4: LJ(6,16,2) parameters for noble gas pair potentials. All values in atomic units. Part IV.

CHAPTER

6

Kanamycin A Dimers

You're bound to be unhappy if you optimize everything.

Donald E. Knuth

6.1 Scope of the Project

Studying the structure of clustered molecules is at the interface of experiment and theory, since it proves to be an experimentally challenging task. It requires to selectively create a specific cluster size and then identify the structure on the geometrical level. Employing the here discussed algorithms for structure prediction is a fruitful technique in assisting experiment.

Higher multicellular organisms are protected by an innate immune systeme. Human skin appears to have a "chemical barrier" function, recognizing bacterial colonization via so-called pathogene-associated molecules (PAMs), triggering various defense mechanisms [147]. Schröder and co-workers found hints that a substance closely related to Kanamycin A (KA) plays and important role and that aggregates of KA building blocks (up to 26 KA units) together with physiological cations are a PAM.

Within this project, the first steps towards studying the aggregation of KA are made by means of global optimization. Obviously, the smallest cluster size – dimers – needs to be studied first to thouroughly assess the method accuracy. For this, different levels of theory from force fields to *ab initio* via semiempirics and DFT are applied to this problem. Additionally, electronic properties have been calculated both for the monomer building block as well as for three different dimer compositions. The calculated IR and NMR spectra provide hints how an aggregation of KA can be detected in experiment.

6.2 Own Contribution

Both the calculations for the global optimization of Kanamycin dimers and the post-processing calculations of higher level energies and electronic properties have been carried out by the first author.

The project also required interfaces to the AMBER, NAMD and MOPAC program packages which have been implemented by the first author in the context of the initial development of the OGOLEM framework.

6.3 Publication

Authors Johannes M. Dieterich and Bernd Hartke

Institut für Physikalische Chemie
Christian-Albrechts-Universität
Olshausenstraße 40
24098 Kiel, Germany

Ulrich Gerstel and Jens-Michael Schröder

Klinik für Dermatologie,
Universitätsklinikum Schleswig-Holstein,
Arnold-Heller-Str. 3,
24105 Kiel, Germany

Title Aggregation of Kanamycin A:
dimer formation with physiological cations

Submitted July 27, 2010

Accepted

Publication Data *J. Mol. Model.*

Noname manuscript No.
(will be inserted by the editor)

Aggregation of Kanamycin A: dimer formation with physiological cations

Johannes M. Dieterich · Ulrich Gerstel · Jens-Michael Schröder · Bernd Hartke

the date of receipt and acceptance should be inserted later

Abstract Global cluster geometry optimization has focused so far on clusters of atoms or of compact molecules. We are demonstrating here that present-day techniques also allow to globally optimize clusters of extended, flexible molecules, and that such studies have immediate relevance to experiment. For example, recent experimental findings point to production of larger clusters of an aminoglycoside closely related to Kanamycin A (KA), together with certain preferred physiological cations, by *Pseudomonas aeruginosa*. The present study provides first theoretical support for KA clustering, with a close examination of the monomer, the bare dimer, and dimers with sodium and potassium cations, employing global cluster structure optimization, in conjunction with force fields, semiempirical methods, DFT and ab-initio approaches. Interestingly, already at this stage the theoretical findings support the experimental observation that sodium cations are preferred over potassium cations in KA clusters, due to fundamentally different cationic embedding. Theoretically predicted NMR and IR spectra for these species indicate that it should be possible to experimentally detect the aggregation state and even the cationic embedding mode in such clusters.

Keywords global cluster structure optimization · genetic algorithms · evolutionary computation · aminoglycoside clustering · pathogen-associated molecules

J. M. Dieterich
Institut für Physikalische Chemie, Christian-Albrechts-Universität, Olshausenstraße 40, 24098 Kiel, Germany

U. Gerstel
Department of Dermatology, University-Hospital Schleswig-Holstein, Campus Kiel, Arnold-Heller-Str. 3, House 19, D-24105 Kiel, Germany

J.-M. Schröder
Department of Dermatology, University-Hospital Schleswig-Holstein, Campus Kiel, Arnold-Heller-Str. 3, House 19, D-24105 Kiel, Germany

B. Hartke
Institut für Physikalische Chemie, Christian-Albrechts-Universität, Olshausenstraße 40, 24098 Kiel, Germany E-mail: hartke@phc.uni-kiel.de

1 Introduction

Clusters are recognized as important objects of scientific study in chemistry and physics [1–3]. Theoretical determination of their most probable structures, however, is a difficult task [4], requiring stochastic-heuristic global optimization algorithms for clusters of interesting sizes. Despite considerable progress in this area [5], most studies still deal with clusters of atoms, or with clusters of small, rigid molecules that do not change their outer shape strongly when turned around, as evidenced e.g. by database entries [6]. Notably, the situation is different in ab-initio crystal structure prediction, where it has become common to treat more difficult flexible molecules [7–9]. In this contribution, we demonstrate that global cluster structure optimization can also deal with larger, flexible molecules.

In addition, we show that such cluster studies do have considerable direct practical relevance. To this end, we have selected clusters of Kanamycin A (KA) as application example. To describe their practical relevance, we briefly digress into medicinal biochemistry in the following paragraph.

The adaptive immune system of higher multicellular organisms, developing antibodies against antigens presented by pathogens, is supported by the evolutionary older innate immune system. Human skin appears to have a "chemical barrier" function, recognizing bacterial colonization via so-called pathogen-associated molecules (PAMs) which trigger various defense mechanisms [10]. Certain PAMs induce production of cytokines in epithelial cells, leading to inflammatory reactions. Other PAMs induce antimicrobial peptides (AMPs) that kill bacteria without inflammatory side reactions, e.g., by forming pores in bacterial cell membranes [10]. Recently, Schröder and coworkers discovered in a mucoid clinical isolated of *Pseudomonas aeruginosa* potent AMP human beta-defensin-2 (hBD-2)-inducing activity of yet unknown origin [11]. Preliminary purification experiments revealed the existence of an hBD-2-inducing factor by *Pseudomonas aeruginosa*, which did not induce proinflammatory cytokines in epithelial cells, thus excluding the origin of the known Toll-like-receptor(TLR)-5-binding bacterial flagellin. In initial attempts to purify the hBD-2-inducer they found in several hBD-2-inducing activity-containing high performance liquid chromatography (HPLC) fractions the aminoglycoside Kanamycin A (KA), or an isomer of it. Nanospray ESI-MS and ^{1}H-NMR have lead to the current working hypothesis that in these active fractions KA may occur in stable dimers and higher aggregates (up to 26 units) together with certain physiological cations, preferentially sodium.

The absolute configuration of the KA monomer was determined via Xray crystallography of the monosulfate-monohydrate species [12]. However, literature on aggregation of KA or related aminoglycosides is extremely sparse, both on the experimental and on the theoretical side. A 1:1 aggregate of KA with Cu^{2+} could be identified via NMR and EPR [13]. A 2:1 complex could be detected in an electrochemical study [14]. Earlier, similar aggregates of the related compound Gentamicin were found with various methods [15]. One study [16] reported formation of long KA fibers on negatively charged surfaces. Other than that, no experimental evidence of KA aggregation has come to our attention. Theoretical studies on KA are rarer still: One group has performed molecular dynamcis and docking studies involving aminoglycosid monomers and RNA [17, 18], using standard force fields and scoring functions. Apparently only in two cases electronic structure theory methods were applied to KA: One of them again in the context of KA binding to RNA [19], the other one for the isolated monomer in the gas phase [20]. Both studies, however, only employed HF calculations with small basis sets.

Theoretical studies on the aggregation of KA or similar aminoglycosides appear to be non-existent.

In a previous paper [21], two of the present authors described the recent development of the OGOLEM program suite for global structure optimization of arbitrarily mixed clusters of flexible molecules. As exemplary application, that work already contained first results of applying this program suite to KA aggregation with Na^+, using the GAFF force field, identifying a 2:1 aggregate as particularly stable. In the present article, we would like to put this application on a more solid foundation, by (1) comparing different levels of theory for the KA monomer, (2) an in-depth analysis and comparison of $(KA)_2M^+$, M=Na,K, and (3) providing contact points to experiment via calculation of NMR and IR spectra, pointing out experimentally accessible signatures of aggregation.

For systems of biochemical importance such as the present one, it certainly is mandatory to also include an implicit or even explicit solvent description. To ease the burden of the present initial steps, however, we defer this task to the next stage of our studies. A methodical reason for this is that it is easier to understand aggregation of a ternary system (KA, cations, water) by first examining the pure main ingredient (KA) and its interaction with the second ingredient (cations), as we do here. Furthermore, experiments indicate that the cation-aminoglycoside clusters are perfectly stable without any solvent molecules under the rather stringent mass-spectrometry conditions.

2 Methods

Our OGOLEM program suite has been presented in full detail in a recent publication [21]. Therefore we just highlight the features and techniques used in this work.

OGOLEM is loosely based on genetic algorithms as described in Ref. [22], using a pool variant as described in Ref. [23], and has been constructed to be universal by design. This includes both a universality in the allowed type and composition of building blocks as well as in the methods used to evaluate energies of structures. For this work, the interfaces towards AMBER[24] and MOPAC[25] have been used, allowing for energy evaluations on classical mechanical and semiempirical levels.

All global optimization calculations mentioned in this paper have been carried out on the classical mechanical AMBER/GAFF level of theory with implicitly relaxed building blocks. As starting structures, KA monomers optimized with B3LYP/TZVP have been used. OGOLEM then created randomized starting dimers including an ion, if applicable. These starting structures where locally optimized using GAFF and added to the genetic pool. The genetic pool has then been used for mating of two structures, where one was picked randomly and the other fitness weighted. These structures where then crossed using the *genotype*, real-number-based *Germany* operator and mutated with a 5% probability. To reduce the number of local optimizations, OGOLEM then checks the resulting children structures for physical sensibility using a collision (CD) and dissociation detection (DD) based on distance criteria and graphs. These are designed to "localize" the search space around possible minimum funnels, drastically reducing search space size. Crossing and mutation are attempted again if either CD or DD flag the structure as unphysical. Only children structures that are neither clashing nor dissociated are then again locally optimized.

Only the fitter of the two children structures is then returned to the genetic pool and only added if it does not violate a fitness diversity criterion, designed to reduce the risk of premature convergence of the genetic pool.

These primitive global optimization steps are then repeated, yielding a pool of minima.

To assess the quality of GAFF for this class of applications, a representative pool for each system containing 1000 structures was then re-optimized using MOPAC/PM3. The resulting structures were again ranked by energy, structures which converged into the same minimum were eliminated and for the best ten structures of each system, the energy was again evaluated using MOLPRO[26] and the DF-LMP2/cc-pVDZ level of theory.

The lowest-energy structure of each system can then be considered to be a good candidate for the global minimum structure.

For these resulting, lowest energy structures, electronic properties were calculated using Orca [27] and the RI-BP86/TZVP level of theory, namely NMR and IR spectra. We are fully aware of the shortcomings of GGA-based DFT for long-ranged interactions, as appearing in clustered systems. Considering the performance penalty of post-HF *ab initio* methods, we need to trade accuracy for speed. Additionally, it can be argued that the dominant intra- and intermolecular interactions in pure KA clusters can be expected to be hydrogen bonds, for which GGA-DFT has a reasonable chance of providing qualitatively correct results. KA clusters with cations like Na^+ and K^+ can be expected to simplify the situation further, due to additional, strong interactions between the cationic charge and the partial charges on the KA molecules, which should be described fairly well by GGA-DFT.

3 Results and Discussion

In order to highlight the effects of aggregation, we first present results for the KA monomer, followed by results for the KA dimer systems, without and with sodium or potassium ions.

3.1 Kanamycin A monomer

The starting point for any kind of attempt towards the global optimization of a cluster is a proper characterization of the building block in its monomer form. While this is relatively trivial for atomic and well-known, small molecular building blocks, it is getting difficult with bigger molecules such as KA. We therefore present both results on the local optimization of a monomer unit as well as monomer properties such as NMR and IR spectra, as a reference for the dimer results to be obtained later.

Starting with the task of local optimization, we compare results for the local optimization of the monomer structure using the semiempirical PM3 method, RI-BP86 and B3LYP with the TZVP basis set as well as RI-MP/def2-TZVPP and the GAFF force field. Since weak long-range forces should not have too big of an impact on the monomer structure, we consider density functional theory to be an appropriate level of theory for comparative purposes.

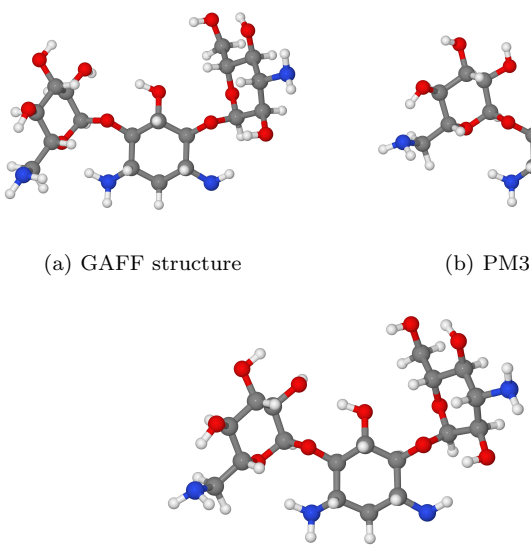
As can be seen in Tab. 1, there is no geometrical difference between the choice of a GGA functional like BP86 and the hybrid B3LYP, the RMSD value being below 0.1 Å.

	PM3	RI-BP86/TZVP	B3LYP/TZVP	RI-MP2/def2-TZVPP
GAFF	5.697	5.724	5.702	4.777
PM3		0.238	0.291	0.186
RI-BP86			0.062	0.145
B3LYP				0.119

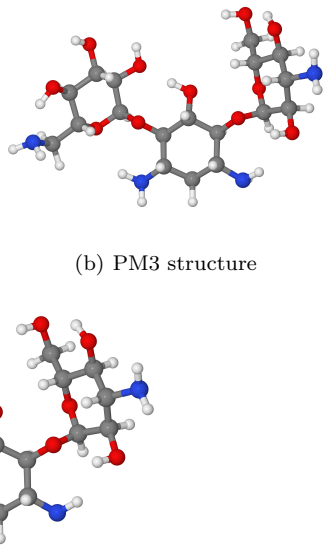
Table 1 RMSD values of the resulting local minima of the monomer. All values in Å.

PM3 does increase the RMSD value to 0.238 and 0.291 Å in comparison to RI-BP96 and B3LYP respectively. The RMSD to the MP2 reference are below 0.2 Å for both DFT methods and PM3, clearly proving that all three hold as valid descriptions of the monomer. Bigger differences arise to the AMBER/GAFF level of theory, bringing the RMSD values up to 5.7 Å. This is neither surprising nor a reason to worry if GAFF is a valid description of KA. Of course, a classical-mechanical description of any molecular or atomic system neglects important quantum mechanical contributions resulting in a noticable error. Furthermore, a seemingly large value of a single numerical measure like the RMSD value is not necessarily a reliable indicator for model breakdown: Obviously, the RMSD value cannot discern between chemically relevant distortions and minor deviations in uninteresting bond length or angles.

Therefore, for better illustration, the monomer structures of GAFF, PM3 and MP2 are depicted in Fig. 1. As can be seen, the qualitative differences between these struc-



(a) GAFF structure



(b) PM3 structure

(c) RI-MP2 structure

Fig. 1 Minimum structures using different levels of theory. Graphical representations of these structures and all following were produced with Jmol[28] and POV-Ray[29].

tures is still negligible, despite the RMSD values of up to 5.7Å. None of the presented deviations allows the conclusion that any of the methods would be not suitable for the local optimization of the monomer molecule. This is of special importance since our ultimate target is the optimization of larger aggregates, using cheap descriptions like force fields. Any method already failing in the monomer process of course would have to be discarded instantly.

As indicated in the introduction, there already is NMR data on (possibly large) aggregates of KA in experiment. For comparison with our future theoretical studies on larger KA aggregates, we have also calculated NMR chemical shifts for the KA monomer and dimers. For this purpose, we have both used the hybrid B3LYP and the GGA BP86 functional in the ORCA program package, predicting NMR peaks in the hydrogen and carbon spectra. Here and in the following, a complete listing of all data can be found in the electronic supplementary information.

For comparative purposes, we carried out several test runs to assess the impact of certain variations in the theoretical approach, in particular, we tested different functionals (GAA vs. hybrid), different basis sets (IGLOIII vs. TZVP) and different geometries (reoptimized vs. non-reoptimized). The full set of chemical shifts can be found in the supplementary information for both hydrogen and carbon. A representative collection can be found in Tab. 2 for hydrogen shifts and Tab. 3 for carbon shifts. For the hydrogen shifts, the replacement of the hybrid B3LYP functional with the GGA BP86 functional causes a shift of the peaks to high field. On the other hand, the substitution of the specialized IGLOIII basis set with the general-purpose TZVP basis set causes a shift to low field. In the case of hydrogen shifts, this yields a good error compensation, causing the resulting RI-BP86/TZVP peaks to be in excellent agreement with the B3LYP/IGLOIII peaks (no deviation bigger than 0.3 ppm). Carrying the chemical shift calculations out on the PM3 geometry causes deviations up to 1 ppm, or approx. 4%, indicating that shielding is less sensitive towards non-minimum structures than other properties like vibrational spectra. This indicates that for larger KA aggregates, qualitative ^{1}H-NMR spectra could be obtained from force field optimized structures without reoptimization at the DFT level. According to further test calculations, we consider these results to be robust under all tested circumstances.

Core	*B3LYP IGLOIII reopt*	*RI-BP86 IGLOIII reopt*	*RI-BP86 TZVP reopt*	*RI-BP86 TZVP PM3 geometry*
2	30.2	29.8	30.2	30.5
15	27.6	27.1	27.7	27.1
32	28.4	28.0	28.3	27.8
34	29.2	28.7	29.1	30.0
67	27.9	27.5	27.8	27.4

Table 2 Representative hydrogen shifts of the KA monomer. All results in ppm. Reference: hydrogen peak of chloroform 23.9 ppm (B3LYP/IGLOIII), 23.2 ppm (RI-BP86/IGLOIII), 24.4 ppm (RI-BP86/TZVP).

In the carbon shifts, similar trends are observed. The substitution B3LYP to BP86 causes a shift to high field, the substitution IGLOIII to TZVP causes a shift to low field. Unfortunately, the combined RI-BP86/TZVP method does not profit from the error compensation encountered in the case of hydrogen shifts. The shift to low field is

signifiantly bigger than the shift to high field, in total causing low-field shifted peaks. Since the chloroform reference shows the same behaviour (low-field shift of 15.6 ppm against B3LYP/IGLOIII), the overall robustness of the RI-BP86/TZVP level of theory is reasonable. The chemical shifts calculated on the PM3 structure now introduce a deviation of up to 20.2 ppm or approx. 14%. Although this is a bigger deviation than in the case of hydrogen shifts, the general agreement and robustness is acceptable.

Core	*B3LYP IGLOIII reopt*	*RI-BP86 IGLOIII reopt*	*RI-BP86 TZVP reopt*	*RI-BP86 TZVP PM3 geometry*
4	128.1	124.0	162.3	162.3
22	70.0	62.7	106.8	103.8
26	100.4	95.6	135.4	129.3
36	139.2	135.6	173.3	170.2

Table 3 Representative carbon shifts of the KA monomer. All results in ppm. Reference: carbon peak of chloroform 73.2 ppm (B3LYP/IGLOIII), 63.5 ppm (RI-BP86/IGLOIII), 98.8 ppm (RI-BP86/TZVP).

Another standard analytical tool are infrared spectra. Since their computational generation is relatively sensitive towards both the shape of the hypersurface in close vicinity to the minimum structure as well as the structural shape, this is also a rather good benchmark for the applied level of theory. In Fig. 2, a comparison is shown between different predictions using the standard BP86 GGA functional and the hybrid B3LYP functional. Trying to estimate the accuracy of the ORCA default settings for the spectral prediction, we thightened the SCF convergence threshold from $5 \cdot 10^{-6}$ E_h to $1 \cdot 10^{-6}$ E_h, enabled central differences to be calculated, reduced the numeric increment to 0.001 a_0 and refined the DFT grid. Introducing a performance penalty of a factor of approx 6.5, these settings should allow for rather converged spectra within the applied level of theory. Comparing the so-obtained spetra with the one with default settings, one can see that the default settings are well sufficient for the KA system, changing only slightly the shape of individual peaks.

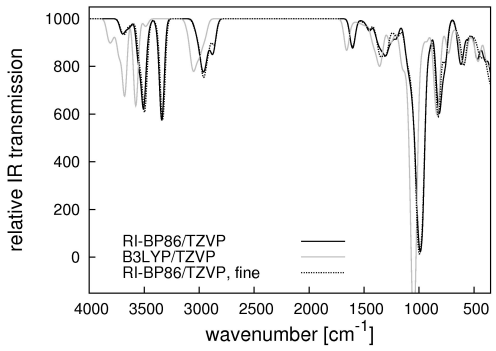


Fig. 2 Predicted infrared spectra of the KA monomer.

The next step then of course needs to be the comparision with another functional. In this case, the B3LYP functional can probably be considered to resemble a somewhat superior choice. Comparing the B3LYP spectra obtained with default settings on a B3LYP optimized geometry to the one obtained with RI-BP, one clearly sees a shift of peaks. Albeit this shift is clearly visible, it is not changing peak orders or shapes in a qualitative way, therefore allowing the conclusion that the BP86 functional allows for a qualitatively correct desription in the context of density functional theory whilst cutting the computational cost by a factor of approx. 14.5 when additionally applying the density fitting approximation to BP86.

Drawing the conclusion that the RI-BP86/TZVP description is sufficient for our purposes, the prediction of Raman spectra is computationally feasible. The monomer spectrum obtained with ORCAs default settings can be found in Fig. 3.

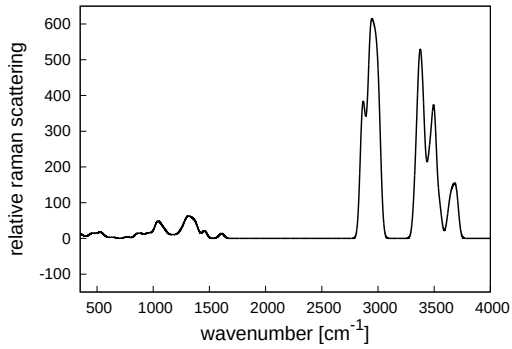


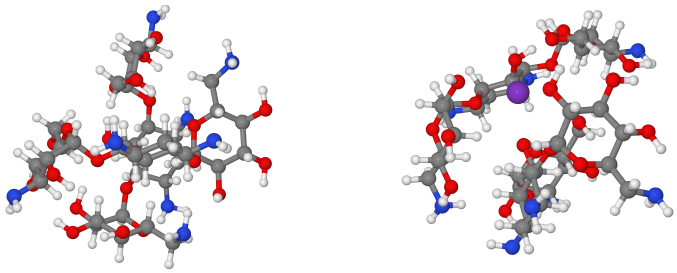
Fig. 3 Predicted Raman spectrum of the KA monomer, at the RI-BP86/TZVP level of theory.

The results presented here are meant as a reference for the effects of aggregation. They definitely have shortcomings stemming from the level of theory (DFT) and numerical approximations (numerical spectra) but comparisons shows that they seem to be sufficiently stable.

3.2 Kanamycin A dimer

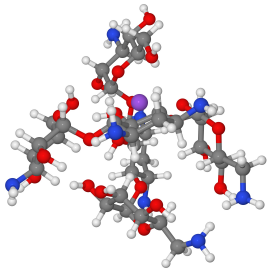
The global optimization of the KA dimer with different physiological cations was carried out using OGOLEM and AMBER with the GAFF force field. Multiple successive runs were carried out, employing a number of global optimization iterations large enough to yield the same result in most of them. Based on this the assumption can be made that a converged result and a good candidate structure for the global minimum was obtained. These global minimum candidate structures are displayed in Fig. 4, for the bare KA dimer and for the KA dimer with a sodium and a potassium cation, respectively.

A noticable difference is encountered. While the dimer without ions and with the sodium ion forms a crossed structure with the ion (if applicable) centered in between the two monomer units, the dimer with the potassium ion forms a more open structure, exposing the ion to the surface. On the one hand, this is an intriguing result since experimental data point to a preference for sodium over potassium in KA aggregates



(a) $(KA)_2$

(b) $(KA)_2K^+$



(c) $(KA)_2Na^+$

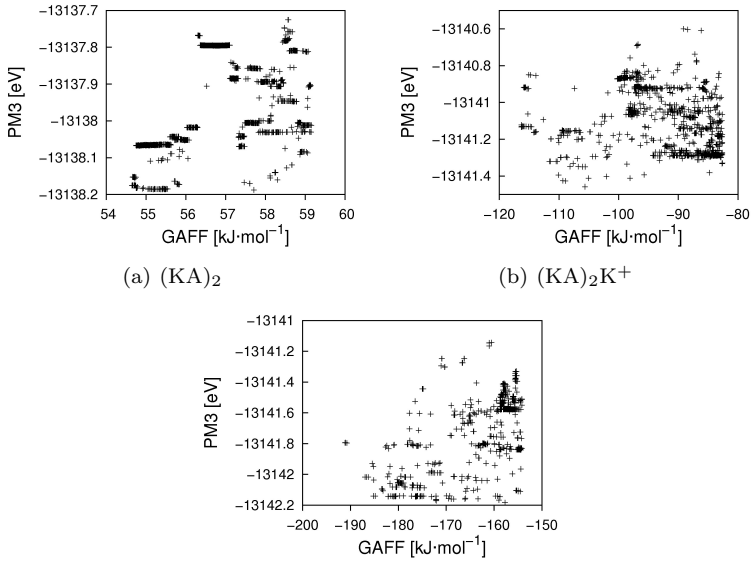
Fig. 4 Minimum structures of the KA dimer on the GAFF level of theory.

with physiological ions. On the other hand, this result obviously cannot be taken as is, since there exists no knowledge on whether or not the chosen level of theory provides even qualitative agreement with higher levels of theory for these systems.

Therefore, a representative final pool from the GAFF runs is taken completely, containing a thousand different minimum structures for each system, and reoptimized using MOPACs PM3 implementation. Correlations between the obtained PM3 and GAFF energies can be found in Fig. 5.

While some remnants of qualitative agreement are visible, the overall correlation is in all cases far from optimal. Additionally, as can be seen from the horizontal accumulations of points (most pronounced in the KA dimer without ion), a lot of GAFF minima seem to disappear on the PM3 hypersurface, leaving a smaller total number of minima. Contrary to first impression, these findings do not rule out the use of this methodical ladder for this system. Correlations of similarly poor quality have been successfully used earlier in studies of other systems; a rare example where bad correlations were admitted and shown was a study on silicon clusters [30]. In such a situation, the key to success is to use a very large number of low-level results as input for the higher-level method, as we do here. Of course, a more correct way to improve matters would be a recalibration of the lower-level method. We reserve this step for future work.

The lowest PM3 structure of each system can be found in Fig. 6. Obviously, the dimer system without ion changes significantly between the GAFF and the PM3 level of theory. The other two systems do not change qualitatively, allowing for the conclusion that despite the unfavorable correlation of energy values found above the GAFF level of theory provides some qualitative agreement with the PM3 level of theory for the best structures. Nevertheless, apparently some caution is in order.



(a) $(KA)_2$ (b) $(KA)_2K^+$

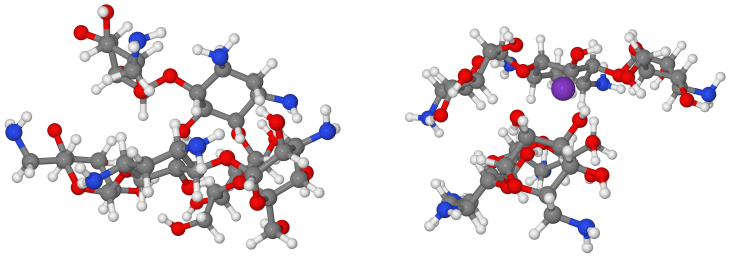
(c) $(KA)_2Na^+$

Fig. 5 Correlations between the GAFF and PM3 energies of optimized structures.

The resulting minimum structures were ranked by PM3 energy and from each system, the lowest ten minimum structures with substantial structural difference were picked manually. The energies of these structures were then recalculated with the linear scaling DF-LMP2/cc-pVDZ level of theory. We are fully aware of the errors introduced by local approximations and the limited basis set size. However, these are the biggest possible calculations for us to-date. As an additional bonus, this local method eliminates basis set superposition error (BSSE) by design, which enhances the reliability of the results for clustered molecules.

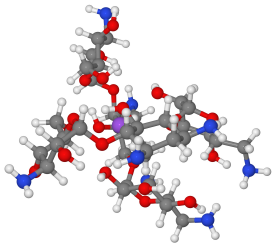
As can be seen from the data depicted in Fig. 7, and in comparison to the GAFF-PM3 comparisons presented above, the correlation between PM3 and LMP2 is rather good for $(KA)_2Na^+$, moderate for $(KA)_2$ and non-optimal in the $(KA)_2K^+$ case. Of course, with only 10 data points the statistics is much worse than for the GAFF–PM3 comparison above. Nevertheless, the conclusion seems to be that PM3 may be usable as predictor for ab-initio results, provided some caution is exercised, as explained in detail above.

This situation is illustrated even better when checking the best LMP2 structures in Fig. 8 in a qualitative manner. Just as in the PM3 and GAFF case, the sodium ion is tightly encapsulated by the two KA monomers interacting to form a hydrogen-bonded network of OH groups around the cation. In contrast, such a tight and extended OH group network is not possible around the larger potassium ion. Instead, in this case the two KA molecules group themselves more loosely around the cation. Actually, in comparison to the sodium cation, the potassium cation is not in the center of the cluster but exposed to its surface. With this observation being robust across strongly different levels of theory, we consider this to be a major clustering tendency of the KA system, worth further investigations targeted at the relative stability of KA aggregates



(a) $(KA)_2$ (GAFF rank 688)

(b) $(KA)_2K^+$ (GAFF rank 100)



(c) $(KA)_2Na^+$ (GAFF rank 583)

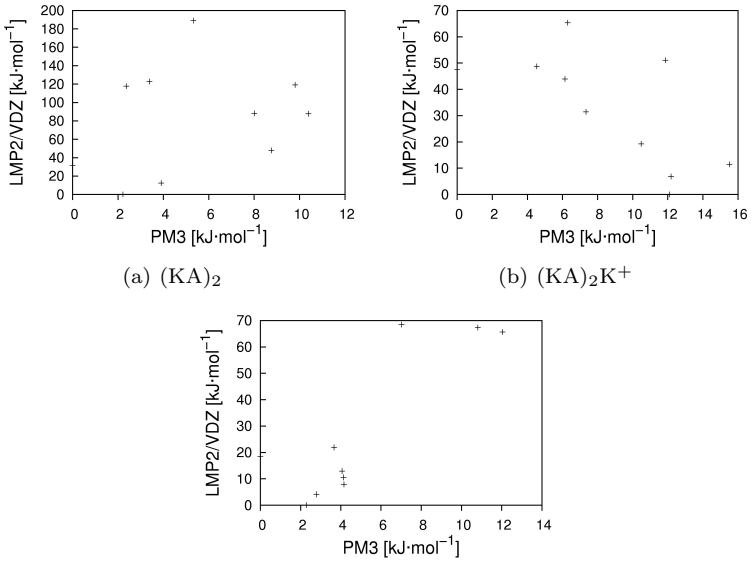
Fig. 6 Minimum structures of the KA dimer on the PM3 level of theory.

of various sizes with ions of different types and numbers. Similar differences in coordinative propensities between these two cations have been noted frequently [31–33]. One simple reason (among others) for this different behavior is the different energy cost for re-orientation [33] in the interaction between the cation and its OH-group coordination environment. Interestingly, one central experimental finding on KA aggregates is that they indeed clearly prefer sodium cations over potassium cations.

Since we want to provide hints for the experiments on how to detect clustering of these molecules, spectral data is of utmost importance. Through easy dilution experiments, an NMR study of aggregation of these systems should be possible. Therefore, the robust chemical shifts are a good and computational feasible choice for prediction of spectral data.

Core	*KA*	$(KA)_2$	$(KA)_2K^+$	$(KA)_2Na^+$
2	30.2	30.1	29.7	29.9
21	29.7	25.3	23.8	28.5
69	30.8	26.5	28.5	26.9
80	27.6	27.4	27.7	27.5
82	26.1	25.0	23.9	22.9
90	29.7	29.2	26.2	25.8
136	27.8	27.6	26.9	26.8
138	30.8	25.1	29.6	25.5

Table 4 Representative hydrogen chemical shifts of the dimers. All values in ppm, calculated with RI-BP86/TZVP on reoptimized structures.



(a) $(KA)_2$ (b) $(KA)_2K^+$

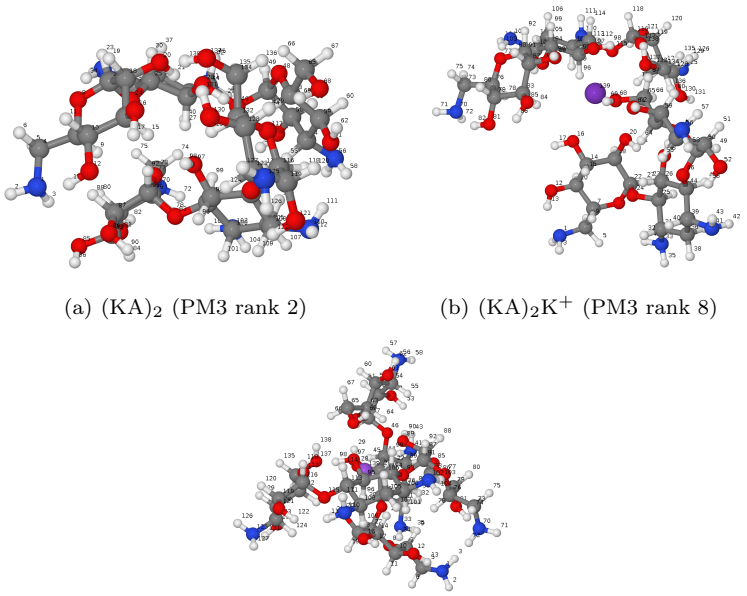
(c) $(KA)_2Na^+$

Fig. 7 Correlations between PM3 and DF-LMP2/cc-pVDZ energies of optimized PM3 structures. Energies referenced to the lowest energy structure.

Core	*KA*	$(KA)_2$	$(KA)_2K^+$	$(KA)_2Na^+$
4	162.3	161.7	161.9	162.5
14	138.4	130.8	133.4	133.3
25	113.0	119.4	116.0	116.2
39	156.6	152.4	156.9	156.5
44	123.5	123.3	110.2	111.0
63	143.5	141.3	136.7	129.8
65	142.1	142.3	142.5	143.9
94	113.0	128.7	114.1	108.4
95	135.4	127.8	135.4	136.7
113	123.5	123.0	109.7	109.4
116	109.1	108.3	102.2	101.6
132	143.5	136.9	132.8	127.8
134	142.1	143.6	143.6	145.9

Table 5 Representative carbon chemical shifts of the dimers. All values in ppm, calculated with RI-BP86/TZVP on reoptimized structures.

In Tabs. 4 and 5, representative results of the calculation of chemical shifts are compiled for all studied dimers and, as reference, for the monomer. The full set of chemical shifts is available from the supplementary information. Starting with the $(KA)_2$ dimer, the biggest changes are observed for 14C, 25C, 94C, 95C and 132C in the carbon shifts. These are all in close range to the other building block and therefore subject to interaction with it. In the hydrogen shifts, the biggest changes are observed for 21H which forms a hydrogen bond to 28O, 69H forming a hydrogen bond to 117O and 138H forming a hydrogen bond to 20O. This is in good agreement with



(a) $(KA)_2$ (PM3 rank 2)

(b) $(KA)_2K^+$ (PM3 rank 8)

(c) $(KA)_2Na^+$ (PM3 rank 2)

Fig. 8 Minimum structures of the KA dimer on the DF-LMP2/cc-pVDZ level of theory.

the observed networks of hydrogen bonds, making the NMR chemical shifts a good indicator of aggregation.

For the $(KA)_2K^+$ system, the biggest changes in carbon shifts are observed for 44C, 113C and 132C which are in close vicinity to the potassium ion. In the hydrogen shifts, the biggest differences occur for the 21H again forming a hydrogen bond to 28O, 69H being subject to attraction of both 61O and 89O as well as 90H forming a hydrogen bond to 61O.

The $(KA)_2Na^+$ system behaves similar to the $(KA)_2K^+$ system, showing the biggest differences for 44C, 63C, 113C and 132C, which are all subject to interaction with the sodium ion. The biggest differences in hydrogen shifts occur for 69H, forming a hydrogen bond to 89O, 82H interacting with 70N, 90H interacting with both 20O and 28O, and 138H being both in the immediate vicinity to the sodium ion and to 68O.

Similarly clear signatures of aggregation are also visible in the IR spectra. Fig. 9 displays an overview of our simulated IR spectra for the bare KA dimer, and for $(KA)_2Na^+$ and $(KA)_2K^+$, respectively, in comparison to the KA monomer spectrum which was already shown in Fig. 2. At first sight, all four IR spectra appear to be very similar. However, a closeup of the OH fingerprint region (cf. Fig. 10) reveals striking differences. The three strong peaks at 3338 cm^{-1}, 3495 cm^{-1} and 3551 cm^{-1} in the monomer spectrum correspond to stretching vibrations of the three OH groups that can form hydrogen bonds to neighboring OH or NH_2 groups in the optimal KA monomer structure. The strongly differing peak positions correspond to the different environments: One OH group at one terminal ring is hydrogen-bonded to a neighboring OH group; it corresponds to the peak at 3551 cm^{-1}. The second OH group sits on the

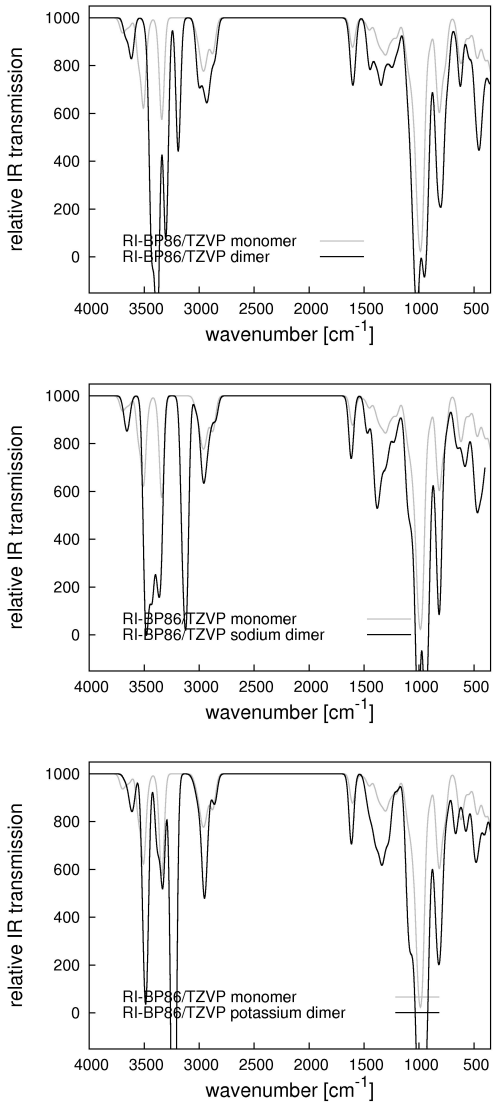
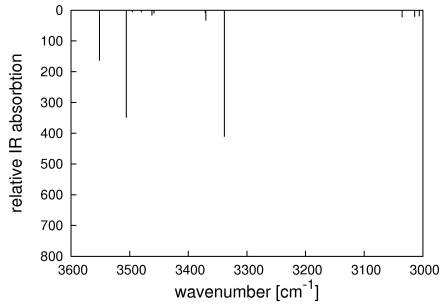


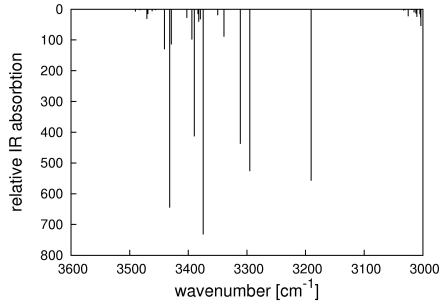
Fig. 9 Predicted infrared spectra of the bare KA dimer (top), and the KA dimers with one sodium cation (middle) and one potassium cation (bottom), respectively.

middle ring and forms a hydrogen bond to an OH group at a terminal ring; it is responsible for the peak at 3495 cm^{-1}. The third OH group is at the other terminal ring and hydrogen-bonds to an NH_2 group there, leading to the peak at 3338 cm^{-1}.

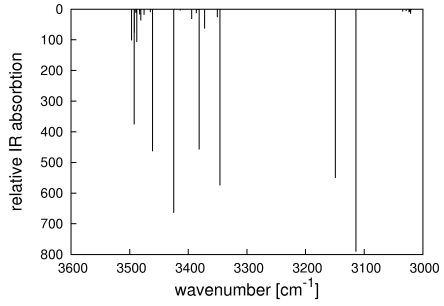
In the KA dimer, the signals from the two terminal OH groups survive but are shifted to 3190 cm^{-1} and 3311 cm^{-1}, respectively, due to changed monomer-internal hydrogen-bond surroundings, which in turn are induced by the relatively close packing of the two monomers in the dimer. Interestingly, the remaining four strongest peaks actually are two peak pairs (in symmetric and antisymmetric versions) involving strong



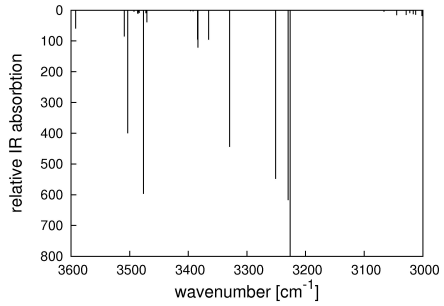
(a) (KA)



(b) $(KA)_2$



(c) $(KA)_2Na^+$



(d) $(KA)_2K^+$

Fig. 10 OH-stretch fingerprint region of predicted IR spectra. Panel (a): KA monomer, (b): $(KA)_2$, (c): $(KA)_2Na^+$, (d): $(KA)_2K^+$.

contributions from two or three OH groups hydrogen-bonded to each other and sitting on different monomers. In other words, they provide a specific signature of the KA–KA aggregation.

Even more interestingly, aggregation signatures in the dimer complexes with cations exhibit both similarities and differences, with respect to the bare dimer. The oxygen atoms of the three characteristic monomer OH groups are actively involved in encapsulating the sodium cation in the $(KA)_2Na^+$ complex. Therefore, the corresponding KA monomer IR peaks are strongly affected by the aggregation: Due to the sodium ion in the immediate neighborhood, the KA monomer peak at 3338 cm^{-1} is strongly redshifted in the complex and reappears as the peak pair at 3114 cm^{-1} and 3149 cm^{-1} (one for each KA in the complex). The mid-ring OH group peak also splits up and appears redshifted at 3346 cm^{-1} and 3381 cm^{-1}. The remaining three strong peaks in the $(KA)_2Na^+$ spectrum (at 3425 cm^{-1}, 3461 cm^{-1} and 3492 cm^{-1}) involve the remaining OH groups of both KA molecules, completing the immediate coordination surroundings of the sodium cation. All three peaks involve strong contributions from two or even three OH groups, and remarkably these always come from both KA molecules. Thus, in addition to the redshift of the peaks mentioned above, in particular these latter three peaks provide a clear IR signature of how the two KA monomers encapsulate the sodium cation: (1) They do this exclusively via the OH groups. The NH_2 groups point to the outside of the cluster, far remote from the cation and with correspondingly small IR intensities. In fact, the tiny background signals in the very same spectral region (3300–3500 cm^{-1}) are almost exclusively due to symmetric and antisymmetric strech vibrations of dangling NH_2 groups. (2) These OH groups form an extensively connected hydrogen-bonded network tightly packed around the sodium cation, as evidenced by multiple, strong OH-stretch combination bands in this spectral region. Dangling OH groups pointing to the cluster outside are responsible for much smaller signals above 3600 cm^{-1}. (The other end of the depicted range, at 3050 cm^{-1} and below, exclusively belongs to CH-stretch vibrations.)

The spectrum of the $(KA)_2K^+$ complex offers a few identical features: Small signals of dangling OH stretch vibrations above 3600 cm^{-1}, tiny peaks corresponding to symmetric and antisymmetric strech vibrations of dangling NH_2 groups in the range 3300–3500 cm^{-1}, and CH-stretch setting in at 3050 cm^{-1}. The most prominent peaks, however, show a strikingly different pattern, compared to the KA monomer, the KA dimer and the $(KA)_2Na^+$ cases, due to a combination of two effects: (1) Normal modes that largely resemble KA monomer ones show a smaller redshift. For example, the peaks at 3229 cm^{-1} and 3250 cm^{-1} essentially correspond to the monomer peak at 3338 cm^{-1}. Thus, they are redshifted by about 100 cm^{-1}, only about half of the amount in the Na^+ case. This may be attributable to the potassium ion being larger and less tightly packed within the KA dimer complex. (2) The remaining normal modes in the fingerprint region for hydrogen-bonded OH stretches are of very different character, compared to the sodium cation case. Since the two KA molecules are less tightly packed, the OH groups closest to the potassium cation do not manage to form an extensively connected hydrogen bonded network. This is visible in the normal modes corresponding to the remaining strong IR peaks: Many of them have significant contributions only from one OH bond, and if a second OH bond significantly contributes it does not belong to the closer neighborhood of the cation anymore. Thus, while it may be difficult to differentiate between the bare dimer and $(KA)_2Na^+$ without close support from theory, it may be possible to detect the significantly less tight packing in $(KA)_2K^+$ without detailed analysis, merely from the absence of strong peaks in

the interval between 3300 cm^{-1} and 3500 cm^{-1} that corresponds to OH-combination bands in the hydrogen-bond network spanning the two monomers and the cation.

In summary, as discussed above, both the NMR and the IR results provide good hints on, and reflect the structural changes occuring in, the different aggregation structures of KA with different ions. Therefore, we consider this to give valuable advice to experimental studies on the KA system.

4 Summary and Outlook

Larger clusters of an aminoglycoside closely related to KA, together with certain physiological cations, have been found in recent experiments on innate immune system response of the human skin. It was unknown, however, if KA forms resonably stable clusters at all, let alone what their properties could be. In this article, we have provided the necessary ground work to launch a large-scale study to provide the badly needed theoretical support for this line of work.

Specifically, with benchmark calculations for the KA monomer, and global optimization studies of the KA dimer and of the systems $(KA)_2Na^+$ and $(KA)_2K^+$, we have tested the performance of the standard GAFF force field and the standard PM3 semiempirical method against DFT and MP2. The obtained method correlations are only partly satisfactory, indicating the need to depart from standard low-level methods and to attempt system-specific recalibrations for future studies of these systems. Nevertheless, a robust finding across all three methods is that Na^+ induces the KA dimer to take on a well-ordered, chelate-like shape around the cation, forming a stable aggregate and shielding the cation from the environment. This does not happen for the potassium cation, where only a loose, unshielded aggregate is formed. This correlates nicely with the experimental finding that Na^+ is preferred over K^+ in these systems.

To provide further convenient contact points with experiment, we have also calculated NMR and IR data for the same systems. Detailed analysis of the results reveals in both cases clear and experimentally accessible indicators for KA-KA and KA-cation aggregation, including the possibility to differentiate between tightly encapsulated and loosely bound structures.

In order to improve the correlations between different calculational levels, future work will address a force field recalibration against the ab-initio data provided here. Our findings indicate that improving the description of the potassium ion should be the most important and perhaps already sufficient ingredient. Besides implicit and/or explicit solvent modeling already mentioned as obligatory extension in the introduction, further important goals to intensify contact with experiment are larger KA aggregates, also including other physiological cations (both experimentally preferred ones like Cu^{2+} and other obvious possibilities like Mg^{2+}, Ca^{2+} or NH_4^+ that do not seem to fit into the KA clusters), and comparisons to minor constitutional isomers of KA (again following experimental indications).

Acknowledgements

B.H. would like to thank the German Research Foundation (DFG) for strong financial support via grant Ha-2498/10. J.M.D. and B.H. thank the North-German Supercomputing Alliance (HLRN) for a generous grant of computer time, and would like to

acknowledge the friendly and competent efforts of the HLRN support staff. J.M.D. would like to acknowledge Mats Eriksson for hints on AMBER's (fixed) input formatting.

References

1. E.E.B. Campbell, M. Larsson (eds.). *The Physics and Chemistry of Clusters*, no. 117 in Proceedings of Nobel Symposium (World Scientific, 2000)
2. R.L. Johnston, *Atomic and Molecular Clusters* (Taylor and Francis, 2002)
3. C.R.A. Catlow, S.T. Bromley, S. Hamad, M. Mora-Fonz, A.A. Sokol, S.M. Woodley, Phys. Chem. Chem. Phys. **12**, 786 (2010)
4. D.J. Wales, *Energy Landscapes* (Cambridge University Press, 2003)
5. B. Hartke, Angew. Chem. Int. Ed. **41**, 1468 (2002)
6. The cambridge cluster database. http://www-wales.ch.cam.ac.uk/CCD.html
7. C.H. Görbitz, B. Dalhus, G.M. Day, Phys. Chem. Chem. Phys. **12**, 8466 (2010)
8. S. Kim, A.M. Orendt, M.B. Ferraro, J.C. Facelli, J. Comput. Chem. **30**, 1973 (2009)
9. P.G. Karamertzanis, A.V. Kazantsev, N. Issa, G.W.A. Welch, C.S. Adjiman, C.C. Panetlides, S.L. Price, J. Chem. Theor. Comput. **5**, 1432 (2009)
10. J.M. Schröder, J. Harder, Cell. Mol. Life Sci. **63**, 469 (2006)
11. J. Harder, U. Meyer-Hoffert, L.M. Teran, L. Schwichtenberg, J. Bartels, S. Maune, J.M. Schröder, Am. J. Respir. Cell Mol. Biol. **22**, 714 (2000)
12. Y.A. Puius, T.H. Stievater, T. Srikrishnan, Carbohydr. Res. **341**, 2871 (2006)
13. N. D'Amelia, E. Gaggelli, N. Gaggelli, E. Molteni, M.C. Baratto, G. Valensin, M. Jezowska-Bojczuk, W. Szczepanik, Dalton Trans. p. 363 (2004)
14. X.Q. Lu, M. Zhang, J.W. Kang, X.Q. Wang, L. Zhuo, H.D. Liu, J. Inorg. Biochem. **98**, 582 (2004)
15. W. Lesniak, W.R. Harris, J.Y. Kravitz, J. Schacht, V.L. Pecoraro, Inorg. Chem. **42**, 1420 (2003)
16. M. Kopaczynska, M. Lauer, A. Schulz, T. Wang, A. Schaefer, J.H. Fuhrhop, Langmuir **20**, 9270 (2004)
17. T. Hermann, E. Westhof, J. Med. Chem. **42**, 1250 (1999)
18. N. Moitessier, E. Westhof, S. Henssian, J. Med. Chem. **49**, 1023 (2006)
19. L. Huang, L. Massa, J. Karle, Proc. Nat. Amer. Soc. **104**, 4261 (2007)
20. M. Monajjemi, M. Heshmata, H.H. Haeria, Biochem. (Moscow) **71**, S113 (2006)
21. J.M. Dieterich, B. Hartke, Mol. Phys. **108**, 279 (2010)
22. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Kluwer Academic Publishers, 1989)
23. B. Bandow, B. Hartke, J. Phys. Chem. A **110**, 5809 (2006)
24. D.A. Case, T.A. Darden, T.E. Cheatham, C.L. Simmerling, J. Wang, R.E. Duke, R. Luo, M. Crowley, R.C. Walker, W. Zhang, K.M. Merz, B. Wang, S. Hayik, A. Roitberg, G. Seabra, I. Kolossváry, K.F. Wong, F. Paesani, J. Vanicek, X. Wu, S.R. Brozell, T. Steinbrecher, H. Gohlke, L. Yang, C. Tan, J. Mongan, V. Hornak, G. Cui, D.H. Mathews, M.G. Seetin, C. Sagui, V. Babin, P.A. Kollman. Amber 10 (2008)
25. J.J.P. Stewart. Mopac2009, version 10.055l
26. H.J. Werner, P.J. Knowles, R. Lindh, F.R. Manby, M. Schütz, et al. Molpro, development version, a package of ab initio programs (2010)
27. Orca: an ab initio, dft and semiempirical electronic structure package.
28. Jmol: an open-source java viewer for chemical structures in 3d. http://www.jmol.org
29. Pov-ray - the persistence of vision raytracer. www.povray.org
30. K.A. Jackson, M. Horoi, I. Chaudhuri, T. Frauenheim, K.A. Jackson, Phys. Rev. Lett. **93**, 013401 (2004)
31. T. Dudev, C. Lim, J. Am. Chem. Soc. **132**, 2321 (2010)
32. M. Eriksson, T.K. Lindhorst, B. Hartke, J. Chem. Phys. **128**, 105105 (2008)
33. F. Schulz, B. Hartke, Chem. Phys. Chem. **3**, 98 (2002)

CHAPTER

7

Design of Switchable Molecules

We always strain at the limits of our ability to comprehend the artifacts we construct – and that's true for software and for skyscrapers.

James Gosling

7.1 Scope of the Project

As pointed out, GAs are a promising method to target not only the previously described non-discrete but also discrete optimization problems. Among those are, in the chemical context, problems of molecular design. In general, molecular design is the optimization of a molecule for a specific application. This project targeted the design of switchable molecules. Attempts have been made to design switchable molecules [148], but none of the attempts targeted this multidimensional problem using global optimization techniques. These have been successfully applied to other problems of molecular design [53, 59, 149, 150].

The ultimate target of this project is to provide an easy to use, multithreading program which optimizes the excitation energies of switchable molecules up to user-definable values. These targets could be reached, implementing a program that tunes molecular switches to wanted wavelengths, given enough chemical flexibility in the allowed substitution sites. This has been demonstrated within this project by tuning the bridged azobenzene backbone to absorb at the wavelengths of cheap industrial lasers. Still there is room for future enhancements in features, both in the easy accessing of more backbones as well as in tuning more properties than just the excitation energies.

Other important properties like the quantum yield have been studied within this project by means of post-processing calculations based on surface-hopping and classical molecular dynamics.

7.2 Own Contribution

Both the implementation of a new program part, `ogolem.switches`, and the calculations for the global optimization of switches have been carried out by the second author. The implementations included a *taboo* algorithm, topological checks and interfaces to the MNDO and MOPAC program packages. The total implementation effort amounted to more than 4500 SLOC to-date.

All surface-hopping and molecular dynamics calculations for post-processing purposes have been carried out by the first author, N. O. Carstensen.

7.3 Publication

Authors Niss O. Carstensen, Johannes M. Dieterich and Bernd Hartke

Institut für Physikalische Chemie
Christian-Albrechts-Universität
Olshausenstraße 40
24098 Kiel, Germany

Title Design of optimally switchable molecules by genetic algorithms

Submitted July 2, 2010

Accepted November 10, 2010

Publication Data *Phys. Chem. Chem. Phys.* DOI: 10.1039/C0CP01065K (2010)

7.4 Additional Information

7.4.1 Outlook

Some ideas directly come to mind for future enhancement of the `ogolem.switches` program part. The first step is to keep standard backbones of switchable molecules ready within the program. These include the studied bridged azobenzene backbone and could include well-known backbones such as azobenzene, stilbene and fulgides. These backbones might require more refined topological checks to distinguish between the different isomers which should be implemented. Obviously, the possibility to use custom backbones should be maintained.

Once these features are included, the next logical step is to allow for an automatic tuning not only of the substitution sites but also of the backbones. This might be a non-trivial task if the topological checks are heavily backbone dependent. In the easiest case this might only mean different optimal dihedral angles, in a more difficult case this could as well mean different topological checks.

Additionally, it might be worthwhile to allow for a higher degree of flexibility in the substitution patterns, allowing the user to specify individually for each substitution site which substituents are allowed. This might be even more of importance once the program is used by synthetic chemists with a more advanced knowledge on which substitution patterns are synthesizable and which are not.

A more refined fitness function might also be of interest, taking into account more properties than just the stabilities and excitation energies of the different isomers.

Ultimately, arbitrary properties of arbitrary molecules should be optimized. With the development of `ogolem.switches`, most parts required to do so are already implemented. A generalization is therefore targeted as the next logical step.

CHAPTER

8

Summary

> The trouble with programmers is that you can never tell what a programmer is doing until it's too late.
>
> Seymour Cray

A new framework, OGOLEM, for the global optimization of chemical problems was developed. The development was carried out in a strictly object-oriented fashion, causing the framework to allow for global optimizations of arbitrary clusters carried out on almost arbitrary levels of theory. Genetic algorithms were chosen as the global optimization technique since they provide very efficient and flexible means not only for non-discrete but also for discrete optimization problems. Parallelizability was a major concern during the development. All program parts are at least parallelized employing Java threads, the optimization of clusters has been additionally parallelized using MPI. In the approximation of Amdahl's law, the parallelized fraction of the program could be proven to be larger than 99.6% using MPI-type parallelization [114][1].

[1] Unpublished results on a Sun T5240 massively parallel server show the same for the shared memory

The developed framework can be used to optimize cluster structures of arbitrary compositions. Additionally, a program package for the optimization of parameters has been implemented, allowing for global optimization on system-specific reparametrized levels of theory. Additionally, the applicability of genetic algorithms for the discrete problem of molecular design – in this case the design of optimally switchable molecules – has been used.

The most important features of the framework where demonstrated by applications. These applications included a benchmark of the program performance. Testing standard benchmark functions, the performance of the framework was found to be good. Also, the solution of these benchmark function was found to scale subquadratic with the dimensionality in all tested cases. Considering that normal real-world applications tend to show a difficulty that scales at least $\mathcal{O}(N^3)$ with the problem dimensionality, a new class of benchmarks has been tested, GRUNGE [151]. Future work on this benchmark class will be carried out, trying to demonstrate its difficulty.

Since the framework does allow for arbitrary cluster compositions by design, highly mixed Lennard-Jones clusters have been studied. These studies included a parametrization of a LJ(6,16,2)-type force field against highest level *ab initio* reference data. By studying the LJ_{38} cluster where already in the homoatomic case different structural types are energetically very close to each other, composition-induced structural transitions could be located [152].

In a cooperation with experiment, the aggregation of Kanamycin A, a pathogen associated molecule (PAM), was studied. Starting with dimers including physiological cations, first insights into these systems could be given. Employing a combination of a standard force field, semiempirics and higher-level *ab initio* calculations, the dimer aggregation could be studied. Calculations of spectra were carried out, showing that the aggregation has a significant in two standard analysis methods, IR and NMR spectra [153]. Future work on these problems will very likely need to include a system-specific force field reproducing higher level calculations.

Optimally switchable molecule were designed, based on the bridged azobenzene backbone. By optimizing the vertical excitation energies to the wavelengths of readily available laser pointers, the capabilities of this program part were tested. It could be shown

case using 128 threads on the *CoolThreads*™ architecture.

that given enough flexibility in the allowed substitution sites, the implemented algorithms are capable of optimizing the vertical excitation energies almost exactly to the specified optima [154].

The developed framework hopefully proves useful in other applications. Since the program is universal by design, no restrictions on either system or level of theory exist. This offers a multitude of possibilities for future applications. These might include further studies on all the projects carried out within this work. Namely, the mixed Lennard-Jones clusters could be studied using force fields employing higher-order terms, e.g. three-body contributions. The aggregation of Kanamycin A will require further studies of bigger systems. For these, the force field used should be reparametrized to better reproduce accurate data. The mentioned GRUNGE class of benchmark functions should be made available to other working groups to allow for performance comparisions of global optimization techniques on the basis of a more real-world difficulty. The molecular design of switchable molecules will be further extended to support higher levels of theory for calculating vertical excitation energies as well as more refined fitness functions. Additionally, the logial next step would be to take environmental effects into account. With very few exceptions, the global optimization of both switchable molecules and clusters is carried out *in vacuo*. Therefore, it might be worthwile to allow for an easy solvation – probably of the implicit type[2] – or optimization inside protein pockets and/or zeolithes and/or on surfaces through e.g. hybrid QM/MM methods.

Ultimately, the target of this work is to provide an easy to use framework for the global optimization of chemical problems. Since this is a moving target, constant future work is required to keep track of it. This obviously also includes a critical assessment of program design decisions.

[2]Through some interfaces to other program packages a COSMO type solvation [155] is utilizable.

Acknowledgments

> And then he was gone in a flash of white light.
>
> *The White Light*
> Rudy Rucker

I would like to thank Bernd Hartke for accepting me in his group. Not only has he been a source of absolutely insane – in the most positive sense – ideas, but also he has proven to me numerous times that he is a wonderful human being and an old-fashioned scientist, interested in knowledge and not self-affirmation. I will keep his positive character traits in best memory: honesty and diplomacy. The latter he kept on trying to teach me. Unfortunately without much success...

I would like to thank Roy Johnston for accepting to be both referee for this work and examinator in the disputation.

All the past and present members of the group I would like to thank for the unique working atmosphere, which at no time hit a global minimum.

Mats Eriksson I would like to thank for numerous espressi, beers, evenings and dinners

spend together. We've been considered a couple or brothers more than once. I do not know which of the two embarrasses him more...

Ole Carstensen not only for the really nice cooperation but also for forgiving me my sometimes pressuring character.

Sascha 'Sushi' Frick I'd like to thank for funny mails, great hints on shirts and awesome looking video clips of genetic operators.

Wulf Thimm for his humor and his unbelievable ability to get to the heart of everything, causing great joy not only for me.

The various people that provided projects, ideas and/or special access to resources I would like to acknowledge (in random order): Dr. U. Gerstel, Prof. Dr. J.-M. Schröder, Prof. Dr. W. Bensch and Prof. Dr. F. Neese.

Thanks to the projects/institutions/people that made this work possible in one way or the other (in random order): the FreeBSD project, the openJDK project, Sun Microsystems, the Jmol project, the POV-ray project, the Linux project, the LaTeX project, HLRN, DFG, the University of Kiel (and its Institute for Physical Chemistry) and all the others.

My other friends (without names since I am doomed to forget someone) from Germany, Portugal, Sweden and Finland I would like to thank for existing and accepting most of my (numerous) flaws, for example my non-existing remembrance of dates...

Special thanks to a special person in my life.

Last but not least, I want to thank my family. Without them I would not even exist. On top, they provided me with the best chances for life one can get.

Bibliography

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Kluwer Academic Publishers, 1989.

[2] E. Fischer, *Ber. Dt. Chem. Ges.* **27**, 2985 (1894).

[3] D. Koshland, *Proc. Natl. Acad. Sci.* **44**, 98 (1958).

[4] A. R. Leach, B. K. Shoichet, and C. E. Peishoff, *J. Med. Chem.* **49**, 5851 (2006).

[5] L. T. Wille and J. Vennik, *J. Phys. A: Math. Gen.* **18**, L419 (1985).

[6] G. W. Greenwood, *Z. Phys. Chem.* **211**, 105 (1995).

[7] M. Hazewinkel, Encyclopaedia of Mathematics, `http://eom.springer.de`, 2010.

[8] D. H. Ackley, *A connectionist machine for genetic hillclimbing*, Kluwer Academic Publishers, 1987.

[9] C. G. Broyden, *J. Inst. Math. App.* **6**, 76 (1970).

[10] R. Fletcher, *Comp. Jour.* **13**, 317 (1970).

[11] R. Fletcher, *Practical methods of optimization*, John Wiley & Sons, 1987.

[12] D. Goldfarb, *Math. Comput.* **24**, 23 (1970).

[13] D. F. Shanno, *Math. Comput.* **24**, 647 (1970).

[14] D. F. Shanno, *Math. Comput.* **24**, 657 (1970).

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 3rd edition, 2007.

[16] R. P. Brent, *Algorithms for Minimization without Derivatives*, Dover Publications, Inc., 2002.

[17] D. H. Besset, *Object-Oriented Implementations of Numerical Methods.*, Academic Press, 2001.

[18] P. W. Hemker, *NUMAL, Numerical procedures in ALGOL 60*, MC Syllabus 47, Mathematical Centre, Amsterdam, 1980.

[19] H. T. Lau, *A Numerical Library in Java for Scientists & Engineers*, CRC Press, 2003.

[20] A. Srivastav, personal communication.

[21] B. Hartke, *J. Comput. Chem.* **20**, 1752 (1999).

[22] B. Bandow and B. Hartke, *J. Phys. Chem. A* **110**, 5809 (2006).

[23] D. M. Deaven and K. M. Ho, *Phys. Rev. Lett.* **75**, 288 (1995).

[24] D. M. Deaven, N. Tit, J. R. Morris, and K. M. Ho, *Chem. Phys. Lett.* **265**, 195 (1996).

[25] H. Takeuchi, *J. Chem. Inf. Model.* **46**, 2066 (2006).

[26] H. Takeuchi, *J. Chem. Inf. Model.* **47**, 104 (2007).

[27] H. Takeuchi, *J. Chem. Inf. Model.* **48**, 2226 (2008).

[28] X. Shao, L. Cheng, and W. Cai, *J. Comput. Chem.* **25**, 1693 (2004).

[29] X. Yang, W. Cai, and X. Shao, *J. Comput. Chem.* **28**, 1427 (2007).

[30] X. Shao, X. Yang, and W. Cai, *J. Comput. Chem.* **29**, 1772 (2008).

[31] S. Y. Chong and M. Tremayne, *Chem. Commun.* , 4078 (2006).

[32] I. Rechenberg, *Evolutionsstrategien: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.*, Frommann-Holzboog, 1973.

[33] H.-P. Schwefel, *Evolutionsstrategie und numerische Optimierung*, PhD thesis, Technische Universität Berlin, Germany, 1975.

[34] J. R. Koza, Technical Report STAN-CS-90-1314, Technical report, Stanford University Computer Science Department, 1990.

[35] L. Fogel, A. Owens, and M. Walsh, *Artificial intelligence through simulated evolution*, John Wiley and Sons Inc., 1966.

[36] V. Minzu and L. Beldiman, *Engineer. App. Art. Intell.* **20**, 993 (2007).

[37] C. Patvardhan, A. Nayaran, and A. Srivastav, Enhanced quantum evolutionary algorithm for difficult knapsack problems., in *Second International Conference on Pattern Recognition and Machine Intelligence, Premi 2007*, edited by A. Ghosh, R. De, and S. Pal, 2007.

[38] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, *Science* **220**, 671 (1983).

[39] D. J. Wales and H. A. Scheraga, *Science* **285**, 1368 (1999).

[40] S. Goedecker, *J. Chem. Phys.* **120**, 9911 (2004).

[41] S. E. Schönborn, S. Goedecker, and S. Roy, *J. Chem. Phys.* **130**, 144108 (2009).

[42] G. Ciuprina, D. Ioan, and I. Munteanu, *IEEE Trans. Magnet.* **38**, 1037 (2002).

[43] J. Kennedy and R. Eberhart, Particle Swarm Optimization, in *Proc. IEEE Int. Conf. Neural Networks*, p. 1942, 1995.

[44] S. T. Call, D. Y. Zubarev, and A. I. Boldyrev, *J. Comput. Chem.* **28**, 1177 (2007).

[45] R. Soltani, F. Jolai, and M. Zandieh, *Exp. Sys. App.* **37**, 5951 (2010).

[46] A. J. Page, T. M. Keane, and T. J. Naughton, *J. Parallel Distrib. Comput.* **70**, 758 (2010).

[47] D. H. Jun and K. El-Rayes, *Autom. Construct.* **19**, 109 (2010).

[48] H. C. W. Lau, T. M. Chan, W. T. Tsui, and W. K. Page, *IEEE Trans. Autom. Sci. Eng.* **7**, 383 (2010).

[49] S. H. Zegordi, I. N. K. Abadi, and M. A. B. Nia, *Comput. Indus. Eng.* **58**, 373 (2010).

[50] M. Watanabe, M. Furukawa, A. Mizoe, and T. Watanabe, *IEEE Trans. Indus. Electr.* **48**, 724 (2001).

[51] G. Ascia, V. Catania, and M. Palesi, *IEEE Trans. Evol. Comp.* **8**, 329 (2004).

[52] K. Jeevan, G. A. Quadir, K. N. Seetharamu, and I. A. Azid, *Microelectr. Inter.* **22**, 3 (2005).

[53] K. Mitra, *Int. Mater. Rev.* **53**, 275 (2008).

[54] D. J. Wales, *Energy Landscapes*, Cambridge University Press, 2004.

[55] R. L. Johnston, *Atomic and Molecular Clusters*, Taylor & Francis, 2002.

[56] H. M. Cartwright, B. Hartke, K. D. M. Harris, R. L. Johnston, S. Habershob, S. M. Woodley, V. J. Gillet, and R. Unger, *Applications of Evolutionary Computation in Chemistry*, Springer, 2004.

[57] B. Hartke, *Angew. Chem.* **114**, 1534 (2002).

[58] B. Hartke, M. Schütz, and H.-J. Werner, *Chem. Phys.* **239**, 561 (1998).

[59] T. Nagata, *J. Organomet. Chem.* **692**, 225 (2007).

[60] D. Hecht and G. B. Fogel, *J. Chem. Inf. Model.* **49**, 1105 (2009).

[61] S. M. Woodley, P. D. Battle, J. D. Gale, and C. R. A. Catlow, *Phys. Chem. Chem. Phys.* **1**, 2535 (1999).

[62] Z. Zhou, V. Siegler, E. Y. Cheung, S. Habershon, K. D. M. Harris, and R. L. Johnston, *Chem. Phys. Chem.* **8**, 650 (2007).

[63] K. D. M. Harris, M. Tremayne, and B. M. Kariuki, *Angew. Chem.* **113**, 1674 (2001).

[64] S. Habershon, Z. Zhou, G. W. Turner, B. M. Kariuki, E. Y. Cheung, A. J. Hanson, E. Tedesco, R. L. Johnston, and K. D. M. Harris, EAGER: A Computer Program for Direct-Space Structure Solution from Powder X-ray Diffraction Data.

[65] G. H. F. Diercksen, B. T. Sutcliffe, and A. Veillard, *Computational Techniques in Quantum Chemistry*, Reidel, Boston, 1975.

[66] M. Born and R. Oppenheimer, *Annalen d. Physik* **84**, 457 (1927).

[67] J. C. Slater, *Phys. Rev.* **34**, 1293 (1929).

[68] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry*, Dover Publications Inc., 1996.

[69] F. Jensen, *Introduction to Computational Chemistry*, John Wiley and Sons, 2004.

[70] D. Cremer and Z. He, *J. Phys. Chem.* **100**, 6173 (1996).

[71] R. J. Bartlett, *Ann. Rev. Phys. Chem.* **32**, 359 (1981).

[72] K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, *Chem. Phys. Lett.* **157**, 479 (1989).

[73] J. A. Pople, M. Head-Gordon, and K. Raghavachari, *J. Chem. Phys.* **87**, 5968 (1987).

[74] P. Y. Ayala and G. E. Scuseria, *J. Chem. Phys.* **110**, 3660 (1999).

[75] G. E. Scuseria and P. Y. Ayala, *J. Chem. Phys.* **111**, 8330 (1999).

[76] P. E. Maslen and M. Head-Gordon, *Chem. Phys. Lett.* **283**, 102 (1998).

[77] P. E. Maslen and M. Head-Gordon, *J. Chem. Phys.* **109**, 7093 (1998).

[78] S. Saebø and P. Pulay, *Chem. Phys. Lett.* **113**, 13 (1985).

[79] N. Flocke and R. J. Bartlett, *J. Chem. Phys.* **121**, 10935 (2004).

[80] P. Pulay, *Chem. Phys. Lett.* **100**, 151 (1983).

[81] T. Kato, *Comm. Pure Appl. Math.* **10**, 151 (1957).

[82] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. M. Jr., D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, *J. Am. Chem. Soc.* **117**, 5179 (1995).

[83] P. M. Morse, *Phys. Rev.* **34**, 57 (1929).

[84] J. Pople and D. Beveridge, *Approximate Molecular Orbital Theory*, McGraw-Hill, 1970.

[85] J. J. P. Stewart, *J. Mol. Model.* **7**, 765 (2009).

[86] M. J. S. Dewar, E. G. Zoebisch, E. F. Healy, and J. J. P. Stewart, *J. Am. Chem. Soc.* **107**, 3902 (1985).

[87] J. J. P. Stewart, *J. Comput. Chem.* **10**, 209 (1989).

[88] J. J. P. Stewart, *J. Comput. Chem.* **10**, 221 (1989).

[89] J. J. P. Stewart, *J. Mol. Model.* **10**, 155 (2004).

[90] J. J. P. Stewart, *J. Mol. Model.* **13**, 1173 (2007).

[91] J. J. P. Stewart, *J. Mol. Model.* **10**, 6 (2004).

[92] P. Hohenberg and W. Kohn, *Phys. Rev.* **136**, B864 (1964).

[93] R. A. da Mata, *Local Correlation Methods in Classical and Quantum Mechanics Hybrid Schemes*, PhD thesis, Universität Stuttgart, 2007.

[94] W. Kohn and L. J. Sham, *Phys. Rev.* **140**, A1133 (1965).

[95] A. D. Becke, *J. Chem. Phys.* **98**, 5648 (1964).

[96] A. D. Becke, *Phys. Rev. A* **38**, 3098 (1988).

[97] J. C. Slater, *Phys. Rev.* **81**, 385 (1951).

[98] S. H. Vosko, L. Wilk, and M. Nusair, *Can. J. Chem.* **58**, 1200 (1980).

[99] C. Lee, W. Yang, and R. G. Parr, *Phys. Rev. B* **37**, 785 (1988).

[100] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, The MIT Press, 2001.

[101] D. E. Knuth, *The Art of Scientific Computing*, Addison-Wesley, 2009.

[102] D. Ritchie, C Programming Language History, `http://www.livinginternet.com/i/iw_unix_c.htm`, 2010.

[103] B. D. Eubanks, *Wicked Cool Java*, No Starch Press, 2005.

[104] J. Bloch, *Efficient Java*, Addison-Wesley, 2009.

[105] The LLVM Compiler Infrastructure Project, `http://llvm.org`.

[106] S. J. Metsker and W. C. Wake, *Design Patterns in Java*, Addison-Wesley, 2006.

[107] G. Amdahl, *AFIPS Conference Proceedings* **30**, 483 (1967).

[108] Java HotSpot Garbage Collection, `http://java.sun.com/javase/technologies/hotspot/gc/g1_intro.jsp`.

[109] The MPI Standard, `http://www.mcs.anl.gov/research/projects/mpi/standard.html`.

[110] Orca: an ab initio, DFT and semiempirical electronic structure package.

[111] R. L. Johnston, *Dalton Trans.* , 4193 (2003).

[112] C. R. A. Catlow, S. T. Bromley, S. Hamad, M. Mora-Fonz, A. A. Sokol, and S. M. Woodley, *Phys. Chem. Chem. Phys.* **12**, 786 (2010).

[113] F. Calvo, *Comp. Mat. Sci.* **45**, 8 (2009).

[114] J. M. Dieterich and B. Hartke, *Mol. Phys.* **108**, 279 (2010).

[115] D. J. Wales and J. P. K. Doye, *J. Phys. Chem. A* **101**, 5111 (1997).

[116] J. Lee, I.-H. Lee, and J. Lee, *Phys. Rev. Lett.* **91**, 080201 (2003).

[117] The Cambridge Cluster Database, `http://www-wales.ch.cam.ac.uk/CCD.html`.

[118] J. P. K. Doye and L. Meyer, *Phys. Rev. Lett.* **95**, 063401 (2005).

[119] S. M. Cleary and H. R. Mayne, *Chem. Phys. Lett.* **418**, 79 (2005).

[120] F. Calvo and E. Yurtsever, *Phys. Rev. B* **70**, 045423 (2004).

[121] V. K. de Souza and D. J. Wales, *J. Chem. Phys.* **130**, 194508 (2009).

[122] D. Parodi and R. Ferrando, *Phys. Lett. A* **367**, 215 (2007).

[123] L. O. Paz-Borbón, R. L. Johnston, G. Barcano, and A. Fortunelli, *J. Chem. Phys.* **128**, 134517 (2008).

[124] L. O. Paz-Borbón, T. V. Mortimer-Jones, R. L. Johnston, and A. Posada-Amarillas, *Phys. Chem. Chem. Phys.* **9**, 5202 (2007).

[125] Y. Xiao and D. E. Williams, *Chem. Phys. Lett.* **215**, 17 (1993).

[126] B. Hartke, *Phys. Chem. Chem. Phys.* **5**, 275 (2003).

[127] S. Kazachenko and A. J. Thakkar, *Chem. Phys. Lett.* **476**, 120 (2009).

[128] B. S. González, J. Hernández-Rojas, J. Bretón, and J. Gomez-Llorente, *J. Phys. Chem C* **112**, 16497 (2008).

[129] J. Hernández-Rojas, J. Breton, J. Gomez-Llorente, and D. J. Wales, *J. Chem. Phys. B* **110**, 13357 (2006).

[130] M. P. Hodges and D. J. Wales, *Chem. Phys. Lett.* **324**, 279 (2000).

[131] B. Hartke, *J. Chem. Phys.* **130**, 024905 (2009).

[132] F. Schulz and B. Hartke, *Chem. Phys. Chem.* **3**, 98 (2002).

[133] F. Schulz and B. Hartke, *Phys. Chem. Chem. Phys.* **5**, 5021 (2003).

[134] F. Schulz and B. Hartke, *Theor. Chem. Acc.* **114**, 357 (2005).

[135] F. Calvo, G. Torchet, and M.-F. de Feraudy, *J. Chem. Phys.* **111**, 4650 (1999).

[136] F. Calvo and G. Torchet, *J. Cryst. Growth* **299**, 374 (2007).

[137] J. P. K. Doye and D. J. Wales, *Chem. Phys. Lett.* **262**, 167 (1999).

[138] J. P. K. Doye, D. J. Wales, W. Branz, and F. Calvo, *Phys. Rev. B* **64**, 235409 (2001).

[139] D. A. Wheeler, sloccount, `http://www.dwheeler.com/sloccount`.

[140] S. Warshall, *J. ACM* **9**, 11 (1962).

[141] G. Chartrand, *Introductory Graph Theory*, Dover Publications, Inc., 1985.

[142] T. Bäck, *Evolutionary algorithms in theory and practice*, Oxford University Press, 1996.

[143] A. Törn and A. Zilinskas, *Global Optimization, Lecture Notes in Computer Science No. 350*, Springer Verlag, 1980.

[144] D. S. H. Mühlenbein and J. Born, *Parallel Computing* **17**, 619 (1991).

[145] H.-P. Schwefel, *Numerical optimization of computer models*, Wiley, 1981.

[146] P. Schwerdtfeger, N. Gaston, R. P. Krawczyk, R. Tonner, and G. E. Moyano, *Phys. Rev. B* **73**, 064112 (2006).

[147] J.-M. Schröder and J. Harder, *Cell. Mol. Life Sci.* **63**, 469 (2006).

[148] N. Adami, D. Fazzi, A. Bianco, C. Bertarelli, and C. Castiglioni, *J. Photochem. Photobiol. A: Chemistry* (2010), in press, doi: 10.1016/j.jphotochem.2010.06.009.

[149] M. C. Durrant, *Chem. Eur. J.* **13**, 2406 (2007).

[150] R. Huenerbein, F. Neese, and S. Grimme, poster presented at the 45th Symposium on Theoretical Chemistry, Neuss, Germany, 2009.

[151] J. M. Dieterich and B. Hartke, *J. Theor. Comput. Chem.* (2010), submitted.

[152] J. M. Dieterich and B. Hartke, *J. Comput. Chem.* (2010), DOI: 10.1002/jcc.21721.

[153] J. M. Dieterich, U. Gerstel, J.-M. Schröder, and B. Hartke, *J. Mol. Model.* (2010), submitted.

[154] N. O. Carstensen, J. M. Dieterich, and B. Hartke, *Phys. Chem. Chem. Phys.* (2010), DOI: 10.1039/C0CP01065K.

[155] A. Klamt and G. Schüürmann, *J. Chem. Soc., Perkin Trans. 2* , 799 (1993).

Declaration

I hereby declare that the work presented in this thesis was done by me, under the supervision of Prof. Dr. Bernd Hartke, with no other help than the referenced sources in the text. This is my first dissertation and the work has never been used in any other dissertation attempts.
The dissertation complies to the good scientific practice rules as proposed by the German Research Foundation (DFG).

Kiel, 20.07.2010 _

Johannes M. Dieterich

Curriculum Vitae

	Johannes Manfred Dieterich
Date of birth	25. 06. 1985
Birth place	Berlin-Zehlendorf, Germany
Nationality	German
Address	Waitzstr. 76
	24118 Kiel

Education

2003 – 2008	Diploma Studies in Chemistry, University of Stuttgart (Germany)
2006	Erasmus Stay at the Institute for Organic Chemistry, Åbo Akademi, Turku (Finland) under the supervision of Prof. Dr. R. Leino.
2008	Diploma Thesis in Theoretical Chemistry: *QM/MM Study of an Aldehyde Oxidoreductase* under the supervision of Prof. Dr. H.-J. Werner
2008 – 2010	Ph. D. Student at Christian-Albrechts-Universität zu Kiel under the supervision of Prof. Dr. B. Hartke

Publications

[1] J. M. Dieterich, H.-J. Werner, R. A. Mata, S. Metz, W. Thiel, *J. Chem. Phys.* **132**, 035101, (2010).

[2] J. M. Dieterich and B. Hartke, *Mol. Phys.* **108**, 279, (2010).

[3] J. M. Dieterich and B. Hartke, *J. Theor. Comput. Chem.*, submitted (November 29, 2010)

[4] N. O. Carstensen, J. M. Dieterich and B. Hartke, *Phys. Chem. Chem. Phys.*, DOI: 10.1039/C0CP01065K (2010)

[5] J. M. Dieterich, U. Gerstel, J.-M. Schröder and B. Hartke, *J. Mol. Model.*, submitted (July 27, 2010)

[6] J. M. Dieterich and B. Hartke, *J. Comput. Chem.*, DOI: 10.1002/jcc.21721 (2010)

Talks

Sep 2009 Talk at the Structure Prediction of Clusters Workshop, London

Jan 2010 Talk at the Institute for Physical Chemistry, Göttingen

Posters

Mar 2009 Poster at the Winter School on Theoretical Chemistry, Jülich

Sep 2009 Poster at the Symposium on Theoretical Chemistry, Neuss