A Comparative Study of Parallel Metaheuristics for Protein Structure Prediction on the Computational Grid

Alexandru-Adrian Tantar, Nouredine Melab, El-Ghazali Talbi
Laboratoire d'Informatique Fondamentale de Lille
LIFL/CNRS UMR 8022, DOLPHIN Project - INRIA Futurs, Cité Scientifique
59655 - Villeneuve d'Ascq Cedex, France
E-mail: {tantar, melab, talbi}@lifl.fr

Abstract

A comparative study of parallel metaheuristics executed in grid environments is proposed, having as case study a genetic algorithm, a simulated annealing algorithm and a random search method. The random search method was constructed in order to offer a lower bound for the comparison. Furthermore, a conjugated gradient local search method is employed for each of the algorithms, at different points on the execution path. The algorithms are evaluated using the protein structure prediction problem, the benchmark instances consisting of the tryptophan-cage protein (Brookhaven Protein Data Bank ID 1L2Y) and α-cyclodextrin. The algorithms are designed to benefit from the grid environment although having no particular optimization for the specified benchmarks. The presented results are obtained by running the algorithms independently and, in a second time, in conjunction with the conjugated gradient search method. Experimentations were performed on a nation-wide grid reuniting five distinct administrative domains and cumulating 400 CPUs. The complexity of the protein structure prediction problem remains prohibitive as far as large proteins are concerned, making the use of parallel computing on the computational grid essential for its efficient resolution.

I. Introduction

With the evolution of high-performance and high-throughput distributed computing and with the proliferation of nuclear magnetic resonance (NMR) data, we are at the dawns of a new era in molecular research and pharmaceutical drug design. A focus is set by current research on molecular structure prediction, molecular folding and molecular docking. Computational modeling and prediction offer an alternative to laboratory experimentation, unfeasible for large size molecules. A viable approach for addressing the implied complexity matters is grid computing, nowadays admitted as a powerful way for achieving high performance on computational-intensive applications.

The protein structure prediction problem, further referred to as PSP, is one of the particularly interesting challenges of parallel computing on the computational grid. The problem consists in determining the ground-state conformation of a specified protein, given its amino-acid sequence - the *primary structure*. The ground-state conformation term designates the associated tridimensional native form, referred to as zero energy *tertiary structure*. Second in the above enumeration, molecular folding represents a closely related problem, the desired outcome being the pathways followed along the folding process in a molecule. Third, molecular docking describes the complexed macromolecule resulting from the binding of two separate folded molecules, exerting geometrical and chemical complementarity. From a computational standpoint, *in silico* docking simulates molecular recognition, although not relating to the molecular pathways of the process but to the final complexed result. Having been studied for more than a decade and of particular interest, protein-protein docking, as a particular case, is fundamental in understanding biomolecular processes, interactions between antibodies and antigens, intra-cellular signaling modulation mechanisms, inhibitor design, macromolecular interactions, etc.

The importance of the PSP problem is determined by the ubiquitousness of proteins in the living organisms, applications of computational protein structure prediction directing to computer assisted drug design and computer assisted molecular design. From a structural point of view, proteins are complex organic compounds composed of amino-acid residues chains joined by peptide bonds - please refer to Fig. 1 (the graphical representations of the *tryptophan-cage* protein included in the figure were created using UCSF Chimera [32]). Proteins are involved in immune response mechanisms, enzymatic activity, signal transduction, etc. Due to the intrinsic relation between the structure of a molecule and its functionality, the problem implies important consequences in medicine and biology related fields.

An extended referential resource for protein structural data may be accessed through the Brookhaven Protein Data Bank[1] [31]. For a comprehensive introductory article on the structure of proteins and related notions and aspects, consult [8]. Also, for a glossary of terms, see [35].

The current article is developed with the support of PPF bioinformatics (Univ-Lille1) within the context of the **DOCK - Conformational Sampling and Docking on Grids** project, sustained by ANR (Agence Nationale de la Recherche - http://www.gip-anr.fr), under the coordination of Prof. El-Ghazali Talbi and reuniting LIFL (USTL-CNRS-INRIA), IBL (CNRS-INSERM) and CEA DSV/DRDC.

1-4244-0910-1/07/$20.00 ©2007 IEEE.

[1] http://www.rcsb.org - Brookhaven Protein Data Bank; offers geometrical structural data for a large number of proteins

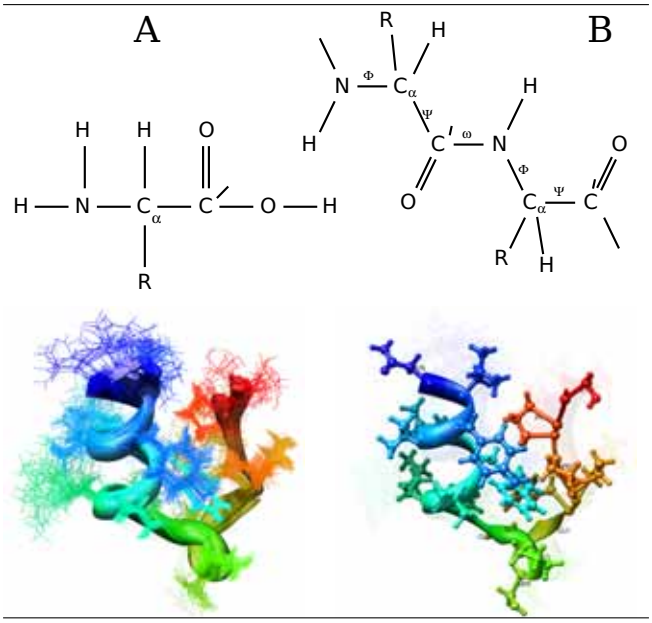


Fig. 1. **First row**: structure of an amino-acid - **A** - $NC_{\alpha}C$ backbone structure; **B** - polymeric structure; ω, Φ and Ψ relate to dihedral angles; R designates the specific amino acid's *side chain* characteristic. **Second row**: tryptophan-cage protein (PDB ID 1L2Y) - **A** - multiple near-native conformations; **B** - ribbon-ball&stick representation of a single conformation.

Referring to complexity aspects of protein structure prediction, as an example, for a reduced size molecule composed of 40 residues, a number of 10^{40} conformations must be taken into account when considering, in average, 10 conformations per residue. Furthermore, if a number of 10^{14} conformations per second is explored, a time of more than 10^{18} years is needed for finding the native-state conformation. For example, for the *[met]-enkephalin* pentapeptide, composed of 75 atoms and having five amino-acids, *Tyr-Gly-Gly-Phe-Met*, and 22 variable backbone dihedral angles, a number of 10^{11} local optima is estimated. Detailed aspects concerning complexity matters were discussed in [25][26], leading to the mention of the *Levinthal's paradox* [7] which states that, despite numerous pathways, *in vivo* molecular folding for example, has a time scale magnitude of several milliseconds. Notes on molecular structure prediction complexity may be found in [24]. As a conclusion, no simulation or resolution is possible unless extensive computational power is applied - it may be inferred that no polynomial time resolution is achievable if no or less *a priori* knowledge is employed.

A. Comparative studies in the literature

Different comparative studies were developed in the literature, considering various areas of interest. Addressing the diverse parametric elements modeling the behavior of conformation sampling methods, the work of [12] discusses the impact of variation operators and local search hybrids employed for flexible ligand docking evolutionary algorithms. The performance of the evolutionary algorithms is evaluated by varying different algorithm specific settings, *e.g.* population size, mutation, recombination and local search operators. The presented results lead to an interesting conclusion, indicating that local search operators determine the EAs to be more prone to getting trapped in local minima, hence being unable to sample an extensive conformational domain. As a counterpart, an annealing scheme for variance control in the mutation operator is identified as being an important efficiency determinant component. In this context, the search for different annealing schemes is underlined as an interesting area for further research.

Another comprehensive study was conducted by [13], comparing a Monte Carlo simulated annealing, a common genetic algorithm and a Lamarckian genetic algorithm. The article discusses the various aspects related to the compared algorithms, problem concepts, empirical free energy function definition, etc. Overall, it is concluded over the efficiency of the Lamarckian genetic algorithm under study, indicating it as the candidate for the case of ligands with an increased number of degrees of freedom. An interesting comparison study is also presented in [14], the authors evaluating a random search procedure for flexible molecular docking and four heuristic search algorithms, namely, genetic algorithms, evolutionary programming, simulated annealing and tabu search. The study is performed on five test cases, using basic implementations of the mentioned heuristics.

B. Article overview

The herein proposed study takes into consideration a hierarchical parallel genetic algorithm and a simulated annealing algorithm. A random search method was also constructed in order to offer a comparison basis. Each of the algorithms is discussed in the following chapters, experimentation setup details being also offered in the results section. As compared to existing research analyses, the herein proposed study aims in opposing and comparing the considered algorithms under a grid environment. Augmented computational resources allow for higher and more complex algorithmic constructions thus rendering possible the design of hierarchical concurrent and parallel approaches. The obtained results offer the base for creating higher level algorithms and patterns combining different strategies.

Although briefly presented the underlying technical aspects of the used framework, as well as the grid environment, were not detailed as being out of scope for our study, allowing for a better focus on the algorithms. Under the same considerations, no in-depth details were included for the employed force field. An important element to be mentioned would be that, although efforts were made to compare the algorithms in a consistent manner, there is no base in opposing the algorithms for the general case - the interest relies on isolating the strengths of each approach. Part of the notions and definitions presented in this paper may also be found in our previous work [16][17], although not related to the herein presented study and addressing bicriterion resolution aspects.

The remainder of the paper is organized as follows: a brief insight of the field is proposed in Section 2 indicating the main directions for solving the PSP problem. Section 3 sketches each of the employed algorithms, offering also details over the parallel models that were used. In Section 4, the ParadisEO-G4 framework is described, along with the subsidiary underlying middleware, Globus-4, the final part of the corresponding section sketching the general implementation aspects. In Section 5, experimentation results are given with an introductory presentation of the GRID5000 computational grid. Section 6 comprises the conclusions.

II. Protein Structure Prediction Insight

The inter-atomic interactions to be considered for the protein structure prediction problem are a resultant of electrostatic forces, entropy, hydrophobic characteristics, hydrogen bonding, etc. Precise energy determination also relies on the solvent effect enclosed in the dielectric constant ϵ and in a continuum model based term. A trade-off is accepted in practice, opposing accuracy against the approximation level, varying from exact, physically correct mathematical formalisms to purely-empirical approaches. The main categories to be mentioned are *de novo, ab initio* electronic structure calculations, semi-empirical methods and molecular mechanics based models. Hybrid and layered approaches were also designed, in order to reduce the amount of performed calculus in the detriment of accuracy.

The mathematical model accurately describing molecular systems is formulated upon the Schrödinger equation, which makes use of molecular wavefunctions for modeling the spatio-temporal probability distribution of constituent particles [9]. There should be noted that, though offering the most accurate approximation, the *Schrödinger* equation cannot be accurately solved for more than two interacting particles. For resolution related aspects, please consult [33], [34].

Extended explanations for the herein exposed directions are available *via* [9][10][11][8].

Ab initio (first principles) calculations rely on quantum mechanics for determining different molecular characteristics, comprising no approximations and with no *a priori* required experimental data. Molecular orbital methods make use of *basis functions* for solving the *Schrödinger* equation. The high computational complexity of the formalism restricts their appliance area to systems composed of tens of atoms.

Semi-empirical methods substitute computationally expensive segments by approximating *ab initio* techniques. A decrease in the time required for calculus is obtained by employing simplified models for electron-electron interactions: *extended Hückel model*, *neglect of differential overlap*, *neglect of diatomic differential overlap*, etc.

Empirical methods rely upon molecular dynamics (*classical mechanics based methods*), and were introduced by Alder and Wainwright [21][22]. After more than a decade protein simulations were initiated on bovine pancreatic trypsin inhibitor - BPTI [23]. Empirical methods often represent the only applicable methods for large molecular systems, namely, proteins and polymers. Empirical methods do not make use of the quantum mechanics formalism, relying solely upon classical Newtonian mechanics, *i.e.* Newton's second law - the equation of motion. As to the basis of the considered approach, we should mention that, according to recent results [27][28], empirical methods exceed *ab initio* methods. Conceptually, molecular dynamics models do not dissociate atoms into electrons and nuclei but regard them as indivisible entities.

Also, hybrid and layered methods exist [18][19][20], connecting several methods through various computing architectures, in an attempt to obtain accurate results at low computational costs, and, consequently, in a reduced period of time.

III. Parallel Metaheuristics for Solving the PSP

A. Encoding of the conformations

The algorithmic resolution of the PSP, in heuristic context, is directed through the exploration of the molecular energy surface. The sampling process is performed by altering the backbone structure in order to obtain different structural variations.

Different encoding approaches were considered in literature, the trivial approach considering the direct coding of atomic Cartesian coordinates [29]. The main disadvantage of direct coding is the fact that it requires filtering and correcting mechanisms, inducing non-negligible affected times. Moreover, by using amino-acid based codings [30], hydrophobic/hydrophilic models were developed. In addition, several variations exist, making use of all-heavy-atom coordinates, C_α coordinates or backbone atom coordinates, where amino-acids are approximated by their centroids.

For the herein described method, an indirect, less error-prone, torsional angle based representation was preferred, knowing that, for a given molecule, there exists an associated sequence of atoms. More specifically, each individual is coded as a vector of torsion angle values - Fig. 2.

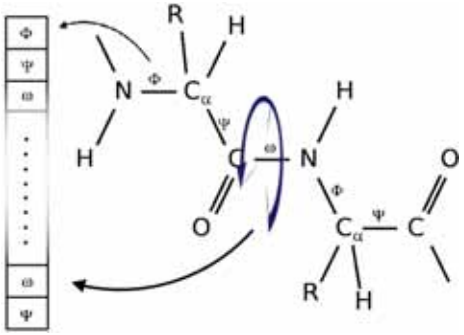


Fig. 2. Chromosome encoding based on specifying the backbone torsional angles.

The defined number of torsion angles represents the degree of flexibility. Apart from torsion angles which move less than a specified parameter, all torsions are rotatable. Rotations are performed in integer increments, energy quantification of covalent bonds and non-bonded atoms interactions being used as optimality evaluation criterion.

B. Scoring function

The scoring function is computed by making use of bonded atoms energy and non-bonded atoms energy through an independently developed force field function. The quantification of energy is performed by using empirical molecular mechanics, as depicted in Table III-B. An extensive discussion on force fields designed for protein simulations, with in-depth details, is offered in the article of [15]. The first part of the mentioned work covers the evolution of the force fields, starting from the 1980s and discussing various formulations which include the *Amber*, *CHARMM* and *OPLS* force fields.

TABLE I

SCORING FUNCTION QUANTIFYING THE INTER-ATOMIC INTERACTIONS.

$E =$	$\sum_{bonds}$	$K_b(b-b_0)^2$
$+$	$\sum_{bondangle}$	$K_\theta(\theta-\theta_0)^2$
$+$	$\sum_{torsion}$	$K_\phi(1-\cos n(\phi-\phi_0))$
$+$	$\sum_{Van\ der\ Waals}$	$\frac{K_{ij}^a}{d_{ij}^{12}} - \frac{K_{ij}^b}{d_{ij}^6}$
$+$	$\sum_{Coulomb}$	$\frac{q_i q_j}{4\pi\varepsilon d_{ij}}$
$+$	$\sum_{desolvation}$	$\frac{Kq_i^2V_j+q_j^2V_i}{d_{ij}^4}$

The involved factors model oscillating entities, the interatomic forces being conceptually simulated by considering interconnecting springs between atoms. A specific constant is associated with each type of interaction, notationally denoted by K_{inter}. An optimal value for the considered entity (bond, angle, torsion) is introduced in the equation as reference for the variance magnitude - $(A - A_0)$. A stands for the experimentation value, while A_0 specifies the natural, experimentally observed value. More specific, b represents the bond length, θ the bond angle, ϕ the torsion angle and q_a, d_{ij} and V_p the electrostatic charge associated with a given atom, the distance between the i and the j atoms and a volumetric measure for the p atom respectively. Although part of the designed algorithms, we considered out of scope for the current study to enter into further details concerning the employed force field. The use of empirical force fields has the drawback of offering results which are not directly comparable with results obtained through another differently-parameterized force field. This inconvenient is avoided by *ab initio* techniques although at the price of high computational demands for calculating the energy of the conformations.

An example of $\alpha - cyclodextrin$ energy surface is given in Fig. 3.

The set of corresponding molecular conformations was obtained by modifying a specified near-optimal initial conformation. Two torsional angles were chosen at random, for each of the two torsional angles, values between 0 and 360 being enumerated, in 10 degrees increments, all the other

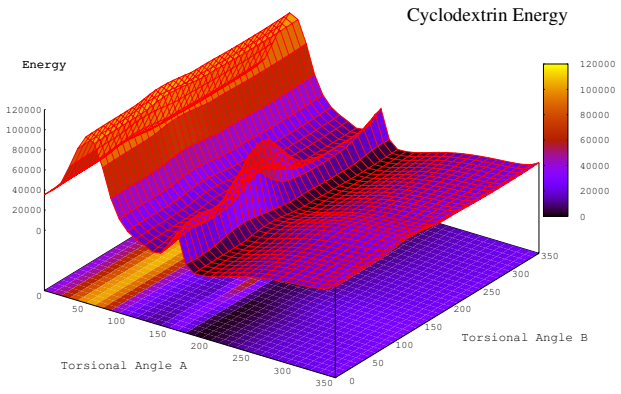


Fig. 3. Energy surface for $\alpha - cyclodextrin$. High energy points are depicted in light colors, the low energy points being identified by the dark areas.

torsional angles being maintained rigid. The lighter areas on the obtained surface correspond to high-energy conformations. Furthermore, an energy-map representation is given, in the XY-plane - only the dark regions are meaningful. The hyper-surface, generated by varying the entire set of torsional angles has an extremely rough landscape, with a large number of local optima.

C. Conjugated Gradient Local Search

The developed methods may benefit from relying on a hybrid architecture, combining, for example, a genetic algorithm with a conjugated gradient-based local search method - thus, a *Lamarckian* optimization technique is constructed.

The exploration and the intensification capabilities of the exploration algorithms, do not suffice as paradigm, when addressing rough molecular energy function landscapes. Small variations of the torsion angle values may generate extremely different individuals, with respect to the fitness function. As a consequence, a nearly optimal configuration, considering the torsion angle values, may have a very high energy value, and thus, it may not be taken into account for the next generations.

In order to correct the above exposed problem, a conjugated-gradient based method may be applied for local search, alleviating the drawbacks determined by the conformation of the landscape. Fig. 4 was obtained by applying the local search technique for each of the conformations that were previously used for generating the $\alpha - cyclodextrin$ energy surface in Fig. 3.

D. Genetic Algorithm

Evolutionary algorithms are stochastic search iterative techniques, with a large area of appliance - epistatic, multimodal, multicriterion and highly constrained problems [4]. Stochastic operators are applied for evolving the initial randomly generated population, in an iterative manner. Each generation undergoes a selection process, the individuals

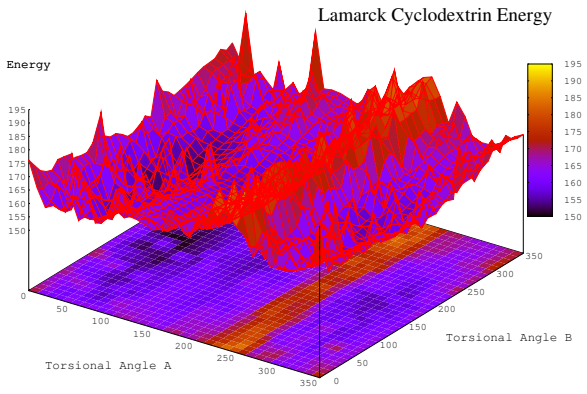


Fig. 4. Energy surface obtained after applying a Lamarck local search on the initial set of conformations.

being evaluated by employing a problem specific fitness function.

Algorithm III-D. EA pseudo-code.

Generate($P(0)$);
t := 0;
while not Termination_Criterion($P(t)$) **do**
  Evaluate($P(t)$);
  $P'(t)$:= Selection($P(t)$);
  $P'(t)$:= Apply_Reproduction_Ops($P'(t)$);
  $P(t+1)$:= Replace($P(t)$, $P'(t)$);
  $t := t+1$;
endwhile

The pseudo-code in Alg. III-D exposes the generic components of an EA. The main subclasses of EAs are the genetic algorithms, evolutionary programming, evolution strategies, etc.

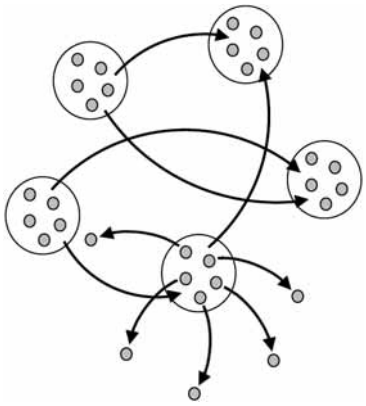
Genetic Algorithms (GAs) are Darwinian-evolution inspired, population-based metaheuristics that allow a powerful exploration of the conformational space. However, they have limited search intensification capabilities, which are essential for neighborhood-based improvement (the neighborhood of a solution refers to part of the problem's landscape). A random population of individuals is evolved in generations through different strategies in order for convergence to be achieved. The *genotype* represents the raw encoding of individuals while the *phenotype* encloses the coded features. For each generation, individuals are selected on a fitness basis, genotype alteration being performed by means of crossover and mutation operators. Applying the genetic operators has as result the modification of the population's structure as to intensify exploration inside a delimited segment or for diversification purposes.

The presented GA is parallelized in a hierarchical manner. First, several GAs cooperate by exchanging their genetic material (parallel island model [4]). Second, as the fitness function of each GA is time-intensive the fitness evaluation phase of the GA is parallelized (parallel evaluation of the population model [4]). These two models are provided in a transparent way through the ParadisEO-G4 framework [1][2], dedicated to the reusable design of parallel hybrid metaheuristics on computational grids.

The granularity of the problem, as counterpart term for the computationally expensive fitness evaluations, biased the resolution pattern towards a parallel, cooperative island-model approach. As a consequence, several populations evolve on a master machine, fitness function evaluations being distributed on remotely available computing units. We have to note that the evaluation of the fitness function consists of several stages, including the calculation of Cartesian atomic coordinates, inter-atomic distances determination, etc.

Complexity may also be addressed by developing specialized operators in conjunction with hybrid and parallel algorithms. The parallel affinity of the EAs represents a feature determined by their intrinsic population-based nature. The main parallel models are the island cooperative model, the parallel evaluation of the population and the distributed evaluation of a single solution.

For a complete overview on parallel and grid specific metaheuristcs refer to [1][3][4][5].



Island deployment model and parallel evaluation of the individuals: multiple algorithms are executed independently in a concurrent manner, migrations of the individuals occurring on a periodic basis. Parallel evaluation of the population is also employed by each algorithm - for simplicity the figure depicts only one algorithm performing the parallel evaluation step.

The designed genetic algorithm follows the above exposed pseudo-code, including in addition two levels of parallelism - island model and parallel evaluation of the population. An island model was adopted in the design, several independent algorithms being executed concurrently, the algorithms exchanging individuals at a predefined number of iterations - emigrants and immigrants. The exchange is performed in an asynchronous manner, *i.e.* no synchronization between the execution/generations of the algorithms is imposed - an important aspect to consider when there is an interest in having algorithms converging at different rates. The emigrant conformations are selected through a stochastic tournament technique, the integration of the immigrant individuals being performed by replacing the worst individuals in the population. At each migration phase, one third of the population is selected for the exchange. In addition, at each generation,

the best obtained conformation till that point is preserved.

In order to exploit the local search capabilities of the conjugated gradient local search method, two different sets of operators were designed. For each operator type, crossover and mutation, a simple approach was followed - a two-point crossover and, respectively, a one-point torsion angle mutation. In addition, operators that apply the local search method on the outcome offsprings were designed. As an example, for the second case, the crossover generates two new conformations starting from two specified parents, the offsprings being optimized by applying the local search method. Similarly, for the mutation operator, after applying a random torsion angle variation on a random chosen angle, the local search method is applied. One potential drawback of this technique resides in the fact that it may lead to a premature convergence of the algorithm, thus a careful balancing of the two sets of operators being required, allowing in the same time for diversity and convergence.

The migrating individuals contribute to maintaining diversity while assuring for the coordinated convergence of all the islands.

E. Simulated Annealing Algorithm

Simulated annealing algorithms are solution-based meta-heuristics and were introduced as a generalization of Metropolis Monte-Carlo techniques, for simulating the evolution of a solid in the process of annealing - refer to Alg. III-E for a simple pseudo-code example. The system starts from an initial disordered state gradually following a cooling schedule, maintaining the thermodynamic equilibrium. Modifications of the current state are accepted on a Boltzmann probability distribution, *i.e.* the acceptance probability being computed according to a previous found state. Simulated annealing algorithms have a performance guarantee of finding the global optimum provided an idealistic long enough schedule is offered. The algorithm may act alternatively as a global search or as a local search method, depending on the schedule.

Algorithm III-E. SA pseudo-code.

```
Generate(S_0);
k := 0;
while T(k) > T_threshold do
    for s := 1 to nbSamples do
        S_rand := randomMove(S_0);
        ΔE := eval(S_rand) − eval(S_0);

        if ΔE < 0 then
                S_0 := S_rand;
        else
                S_0 := S_rand with prob. 1.0 / (1.0 + e^(ΔE/T(k)))
        endif
    endfor
    k := k + 1;
endwhile
```

The main drawback of the simulated annealing algorithm consists in the fact that it is difficult to parallelize without breaking the underlying philosophy, resulting in high computational-demanding methods. As a counterpart, there is no optimality guarantee proof for the genetic algorithm. For our study, a limited number of samples were generated at each step of the schedule, the samples being evaluated in parallel. Furthermore, a synchronous multi-start model is employed for launching several SA algorithms in parallel on a random generated set of initial solutions, at each step of the schedule, a specified number of sampled conformations being evaluated in parallel. The overall best found value is considered as final result. Although more complex simulated annealing variants may be constructed, for our purposes a minimalist version was preferred as to not induce an artificial bias between the compared algorithms.

Another problem to be considered for the simulated annealing algorithm would be the design of a cooling schedule to be followed. For our case a simple exponential decreasing schedule was considered, at each iteration of the algorithm, the temperature being reduced by multiplication by a fixed constant. In this case, the initial temperature must also be attentively selected. More sophisticated variants of simulated annealing algorithms render the method less sensitive to the different parameters involved, *i.e* adaptive versions, etc. For the scope of the proposed study, addressing parallelism issues, the algorithm was developed in a basic form.

F. Random search method

The developed random search method (pseudo-code example in Alg. III-F) does not comport any optimization - the only aspect to be mentioned is the parallel evaluation of randomly generated set of conformations, in an iterative manner. Evaluating the conformations in parallel for this case only reduces to alleviating the walltime of the computation. The overall best found conformations is considered as final result.

Algorithm III-F. Random search method pseudo-code.

```
for s := 1 to nbIterations do
    P_rand := generateRandomConf(nbConformations);
    evaluateInParallel(P_rand);
    updateOverallBestFoundSolution(P_rand);
endfor
```

IV. ParadisEO-G4 based implementation

ParadisEO[2] is a framework dedicated to the reusable design of parallel hybrid meta-heuristics by providing a broad range of features, including EAs, local search methods, parallel and distributed models, different hybridization mechanisms, etc. The rich content and utility of ParadisEO increases its usefulness.

ParadisEO is a C++ LGPL white-box open source framework, based on a clear conceptual separation of the meta-heuristics from the problems they are intended to solve. This separation, and the large variety of implemented optimization features, allow a maximum code and design reuse. Changing existing components and adding new ones can be easily done, without impacting the rest of the application.

ParadisEO is one of the rare frameworks that provide the most common parallel and distributed models, portable

[2]http://paradiseo.gforge.inria.fr

on distributed-memory machines and shared-memory multiprocessors, as they are implemented using standard libraries such as MPI, PVM and PThreads. The models can be exploited in a transparent way - one has just to instantiate its associated ParadisEO components. The user has the possibility of choosing, by a simple instantiation, the MPI or the PVM for the communication layer. The models have been validated on academic and industrial problems, and the experimental results demonstrate their efficiency [5]. The architecture of ParadisEO-G4 is layered as it is illustrated in Fig. 5.

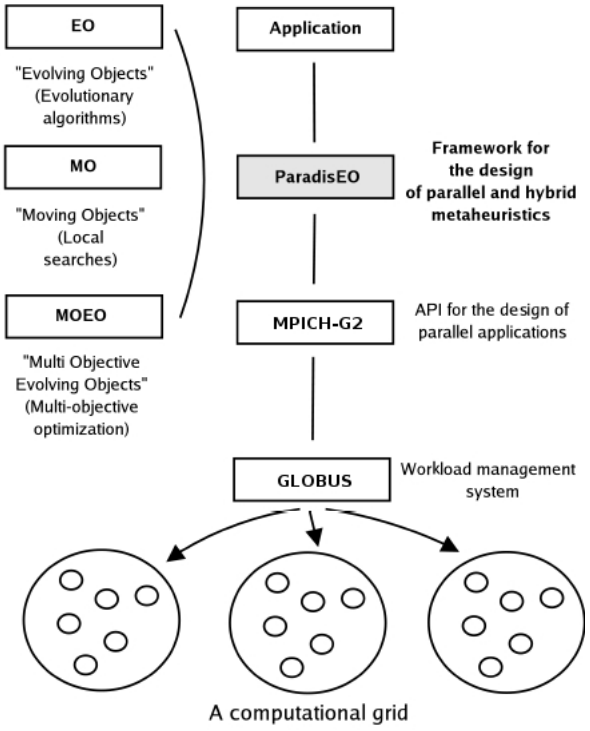


Fig. 5. A layered architecture of ParadisEO-G4.

From a top-down view, the first level supplies the optimization problems to be solved using the framework. The second level represents the ParadisEO framework, including optimization solvers, embedding single and multicriterion meta-heuristics (evolutionary algorithms and local searches). The third level provides interfaces for MPICH-G2 based programming. The fourth and lowest level supplies communication and resource management services.

The implementation relies on invariant elements provided by the ParadisEO-G4 framework, providing support for the insular model approach, as well as for distributed and parallel aspects concerning the parallel population evaluation. In this context, deployment related aspects are transparent, the focus being oriented on the application-specific elements.

The main steps to be performed, in order to configure the environment and to deploy the algorithm, consist in specifying the individuals encoding, the specific operators and the fitness function. Furthermore, elements concerning selection mechanisms and replacement strategies must be specified, along with configuration parameters (number of individuals, number of generations etc).

V. Experiments and Results

The underlying support for performing the experiments was GRID5000, a French nation-wide experimental grid, connecting several sites which host clusters of PCs interconnected by RENATER[3] (the French academic network). GRID5000 is promoted by CNRS, INRIA and several universities[4] - please refer to [6] for details.

At this time the GRID is gathering more than 2600 processors with around 2.5 Tb of cumulated memory and 100 Tb of non-volatile storage capacity. Inter-connections sustain communications of 2.5 Gbps (10 Gbps soon). The target point to be achieved is a marker-stone of 5000 processors for 2007, regrouping nine centers: Bordeaux, Grenoble, Lille, Lyon, Nancy, Orsay, Rennes, Sophia-Antipolis, Toulouse.

The runs were executed on five sites, namely Lille, Nice - Sophia Antipolis, Lyon, Nancy and Rennes, cumulating an overall of 400 CPUs. The GRID is designed to allow a per reservation utilization of the resources - no interferences may occur during the experiments, the allocation of the resources being associated only with the user which requested the reservation. The demanded resources are completely available during the entire experimentation time, unless *in extremis* exceptional events occur. Detailed information regarding all the involved functional aspects may be accessed on the GRID5000 web site: https://www.grid5000.fr. Graphical results are presented in Fig. 6, Fig. 7 and Fig. 8 - as previously mentioned, the benchmark instances were the tryptophan-cage mini-protein (PDB ID 1L2Y) and α-cyclodextrin (which is not a protein but it represents a special interest due to its toroidal structure and its applications in drugs development).

A. Configuration of the algorithms

1) Genetic Algorithm: The genetic algorithm has been designed to iterate for 100 generations, the size of the population being maintained to a constant number of 300 individuals. As mentioned in section III-D, two sets of operators were employed with the following probabilities: 0.95 - crossover operator, 0.15 - local search crossover, 0.05 - mutation operator and 0.05 - local search mutation. In addition, overall probabilities were associated for each type of operators: 1.0 and 0.1 for the crossover and the mutation, respectively. This results, for example, in a probability of 0.95 for applying the simple crossover operator and of 0.15 for applying the local search crossover, given the overall 1.0 associated probability - a simple roulette wheel decision is performed. A stochastic tournament selection strategy was enclosed in the algorithm for selecting offsprings out of the population to be evolved. The replacement phase for back-integrating the offsprings is performed by using also a stochastic tournament strategy.

[3]Réseau National de Télécommunications pour la Technologie, l'Enseignement et la Recherche - http://www.renater.fr

[4]CNRS - http://www.cnrs.fr/index.html; INRIA - http://www.inria.fr.

Another element with important consequences over the algorithm is the asynchronous migration parameterization - frequent migrations may result in a premature convergence while distant migrations fall in the opposite case (the algorithms having independent evolutions). For our case, one sixth of the population migrates at each five generations, in asynchronous manner (migrations occur at different times, depending on the evolution of the algorithm). A stochastic tournament selection strategy is being applied for selecting the emigrant individuals while the immigrant discard the worse individuals in the target population. Migrations are performed inside a ring topology, each algorithm having a source island for receiving individuals and a destination island for sending the emigrant individuals.

2) Simulated Annealing: The main problem for implementing the simulated annealing algorithm in order to offer a consistent comparison base was to assure that the same number of evaluations is performed as for the genetic algorithm. The cooling schedule is given by an interpolation curve defined through a pre-defined set of control points. The resulting curve was defined to mimic to some extent an exponential schedule. The initial temperature was set to 1000, the final threshold being set to 0.1 - each of the extremes is defined by an associated control point. In addition, a fixed number of steps is considered, each step having an associated temperature given by the interpolation curve schedule. At each step ten random moves are performed by modifying the solution found at the given step - each of the generated iterations is evaluated in parallel. Furthermore, three instances of the simulated algorithm are launched in parallel, the final solution being given as the overall optimal solution.

3) Random Search: No special settings were defined for the random search method - the only constraint consists in having the same number of evaluations as for the genetic algorithm. The method is being executed in an iterative manner, at each iteration a pre-defined number of conformations being randomly generated (300 conformations for our case), to be evaluated in parallel in a master-slave model.

B. Experimental outcomes

An important improvement is obtained by applying the gradient local search method - the main disadvantage of the method consists in the fact that it blocks easily in local optima points. As a comparison, for the α-cyclodextrin, the set of solutions found by genetic algorithm hybridized with the gradient method had an average of 201.37 $kcalmol^{-1}$ (stdev. 21.82 $kcalmol^{-1}$), with a maximum of 243.05 $kcalmol^{-1}$ and a minimum of 161.69 $kcalmol^{-1}$ while the genetic algorithm alone gave a set of solutions with an average of 3790.56 $kcalmol^{-1}$ (stdev. 708.54 $kcalmol^{-1}$) and a maximum, minimum of 5845.27 $kcalmol^{-1}$, 2470 $kcalmol^{-1}$, respectively. For both approaches, the maximum number of generations was maintained at 100 - for each case, the number of steps for the gradient method was set to 30. A number of 30 independent executions were performed for the hybridized GA as well as for the GA alone. Furthermore, the runs of the simulated annealing algorithm had as result a set of solutions with an average of 2359.26 $kcalmol^{-1}$ (stdev. 281.12 $kcalmol^{-1}$), with a maximum of 4593 $kcalmol^{-1}$ and a minimum of 1029.14 $kcalmol^{-1}$. All the solutions obtained for the α-cyclodextrin by using the hybridized GA were bellow the native energy - not a guarantee for a correct solution but rather a measure of the exploration capabilities of the algorithm. This was not the case for the *tryptophan-cage* protein for which the algorithm did not descend to a native-energy level. The average computational load for the computers performing the evaluation phase varied in the range of 90%-99% - at the opposite end, the computer executing the island genetic algorithm remained in a range which did not pass 3% of the CPU power.

In Fig. 6, a graphical comparison is offered for the genetic algorithm and the simulated annealing algorithm - the results obtained for the later one were stretched in order to be overlayed inside the same interval, thus offering an overall perspective. Only the best individuals (for each of the algorithms deployed in the insular model) were considered, at each generation; for the simulated annealing algorithm only the improvement points were represented. For the presented case, the genetic algorithm was hybridized with the conjugated gradient method, an elitist approach being adopted. The conformations obtained at the end of the SA runs were further optimized by applying the gradient search method, resulting in improvements in the range of 5%-10%. As multiple genetic algorithms are being concurrently executed as part of the island model, the improvements for each of the algorithms are being depicted as points, an overall interpolation evolution curve unifying the results. In the same manner, the improvements for the simulated annealing algorithms are depicted as blue points. The second snapshot in Fig. 6 represents a fitness decorrelation measure for the island-executed GAs over all the generations, offering an overview of the fitness dynamics along the execution path. The decorrelation measurements offers information regarding the convergence rate of the algorithm thus given the possibility of deciding on the different parameters of the algorithm - it may be observed the tendency of converging towards the upper limit near the 100 generation.

A typical run of a genetic algorithm for the α-cyclodextrin is exposed in Fig. 7. Furthermore, Fig. 8 includes results for the α-cyclodextrin for the simulated annealing algorithm, in the first row, and the results for the random search method, in the second row. Neither of the algorithms manages to obtain a close result, as opposed to the native conformation, in the fixed time-frame / number of conformations, for the 1L2Y conformation (with 46.446 kcal mol^{-1}). Nonetheless, for α-cyclodextrin, with native-conformation energy of 242.4 kcal mol^{-1}, the genetic algorithm descends below this energy, reaching at conformations with energy as low as 160 kcal mol^{-1}.

Although an artifact of the force field parametrization, the obtained low-energy α-cyclodextrin conformations result comes to sustain the chosen approach as a viable exploration technique.

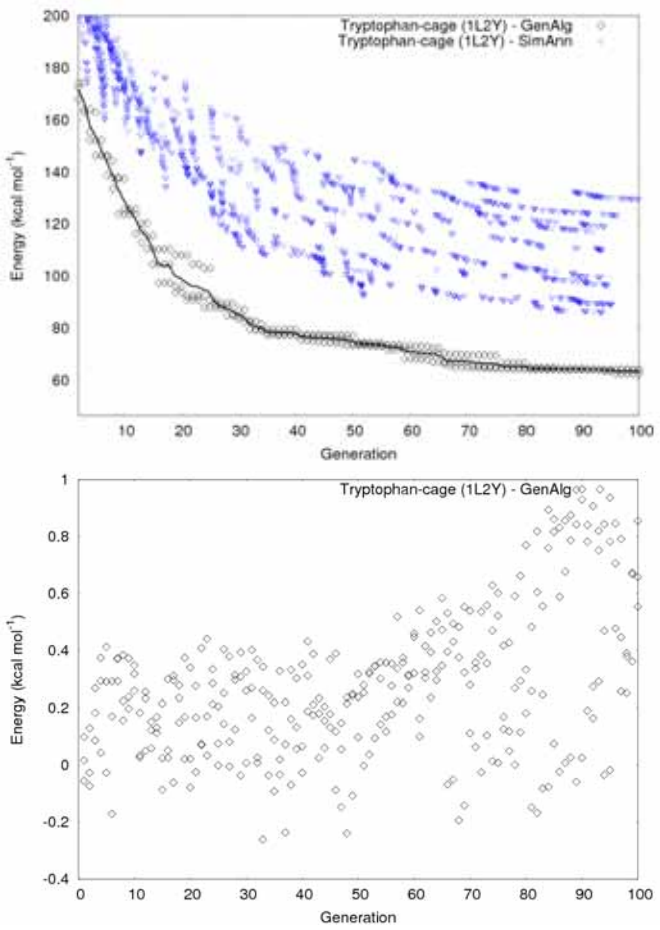


Fig. 6. Experimental results - execution of an island-model genetic algorithm hybridized with a conjugated gradient local search method. As an overlay, on the first picture, the search paths of several synchronously started simulated annealing algorithms - only the improvement points were depicted. The second picture offers an image of the GA fitness decorrelation over the generations.

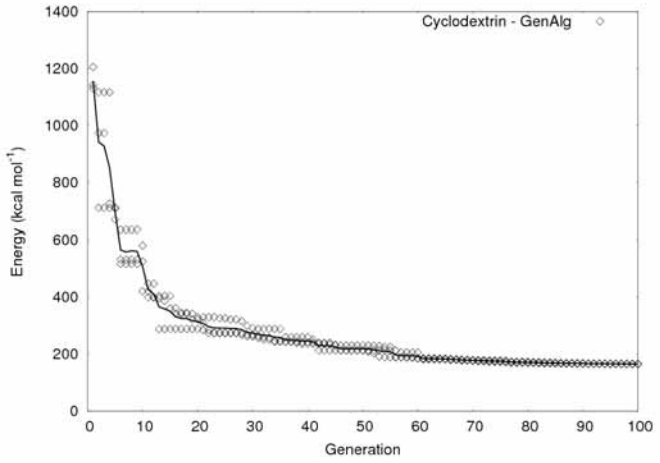


Fig. 7. Experimental results for α-cyclodextrine, using the genetic algorithm combined with the gradient local search method.

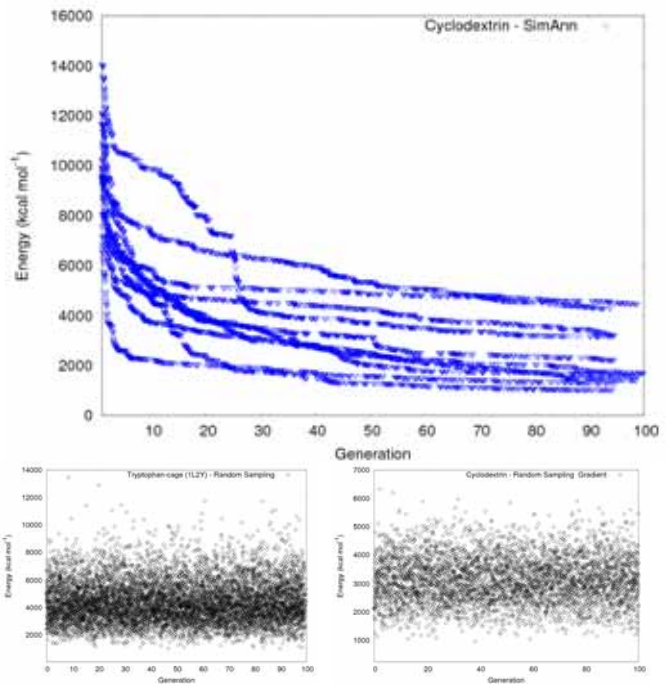


Fig. 8. Experimental results: **first row** - results for α-cyclodextrine using the simulated annealing algorithm; **second row** - random search method results for 1L2Y (no gradient search applied) and α-cyclodextrine (combined with the gradient search), respectively.

The second row in Fig. 8 offers the graphical representation for the random method results - the 1L2Y protein, no gradient applied, and α-cyclodextrin, combined with the conjugated gradient search method. As expected, much worse results are obtained - the method was used only for comparison purposes, with no real practical application.

VI. Conclusions and Future Work

For the case of our study, the genetic algorithm proved to better behave on the considered benchmarks. A few things to be mentioned would be that, in the given time frame, the genetic algorithm alone, without the conjugated gradient local search method, does not outperform the simulated annealing algorithm. The sequential nature of simulated annealing algorithm as well as its underlying philosophy allows for little intrinsic parallelism. Furthermore, for the simulated annealing algorithm, hybridization schemes with the gradient method are not straight-forward paths to follow. Depending on the implementation template, the algorithm tends to get easily trapped in local minima or becomes subject to moves only from one local optimum to another, allowing for less or no uphill changes.

The immediate idea to follow would be to hybridize the genetic algorithm with the simulated annealing algorithm. The first offers a variate range of conformations to serve as starting points for further sampling while the second is less prone to getting easily trapped in local minima, unlike the gradient local search method. Considering in addition the nature of the execution environment, hierarchical multi-stage parallel models may be envisaged, combining the insular

model with the parallel evaluation of the conformations and the synchronous multi-start model.

REFERENCES

[1] S. Cahon, N. Melab, E.-G. Talbi, *An Enabling Framework for Parallel Optimization on the Computational Grid, Proc.* 5^{th} *IEEE/ACM Intl. Symposium on Cluster Computing and the Grid (CCGRID'2005)*, Cardiff, UK, 9-12 May, 2005.

[2] N. Melab, S. Cahon, E.-G. Talbi, *Grid Computing for Parallel Bioinspired Algorithms*, Journal of Parallel and Distributed Computing (JPDC), Elsevier Science, Vol. 66(8), pp. 1052-1061, Aug. 2006.

[3] E.-G. Talbi, *A Taxonomy of Hybrid Metaheuristics*, Kluwer Academic Publishers, Journal of Heuristics, 8:541-564, 2002.

[4] E. Alba and G. Luque, E.-G. Talbi and N. Melab, *Metaheuristics and parallelism*, Edited by E. Alba, J. Willey and Sons, 2005.

[5] S. Cahon, N. Melab, E.-G. Talbi, *ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics*, Kluwer Academic Publishers, Journal of Heuristics, 10:357-380, May 2004.

[6] Raphael Bolze, Franck Cappello, Eddy Caron, Michel Dayde, Frederic Desprez, Emmanuel Jeannot, Yvon Jegou, Stephane Lanteri, Julien Leduc, Nouredine Melab, Guillaume Mornet, Raymond Namyst, Pascale Primet, Benjamin Quetier, Olivier Richard, El-Ghazali Talbi, Irea Touche, *Grid'5000: A Large Scale and Highly Reconfigurable Experimental Grid Testbed*, International Journal of High Performance Computing Applications, Vol. 20, No. 4, 481-494, 2006.

[7] Cyrus Levinthal, In Proc. *How to Fold Graciously*, Mossbauer Spectroscopy in Biological Systems, (ed. J. T. P. DeBrunner and E. Munck), pp. 22-24, University of Illinois Press, Proceedings of a meeting held at Allerton House, Monticello, Illinois, 1969.

[8] Arnold Neumaier, *Molecular Modelling of Proteins and Mathematical Prediction of Protein Structure*, SIAM Review, 39:407-460, 1997.

[9] H. Dorsett and A. White, *Overview of Molecular Modelling and Ab Initio Molecular Orbital Methods Suitable for Use with Energetic Materials*, Department of Defense, Weapons Systems Division, Aeronautical and Maritime Research Laboratory, DSTO-GD-0253, Salisbury South Australia, September 2000.

[10] A. White, F.J. Zerilli and H.D. Jones, *Ab Initio Calculation of Intermolecular Potential Parameters for Gaseous Decomposition Products of Energetic Materials*, Department of Defense, Energetic Materials Research and Technology Department, Naval Surface Warfare Center, DSTO-TR-1016, Melbourne Victoria 3001 Australia, August 2000.

[11] Paul Sherwood, *Hybrid Quantum Mechanics/Molecular Mechanics Approaches*, Modern Methods and Algorithms of Quantum Chemistry, Proceedings, Second Edition, J. Grotendorst (Ed.), John von Neumann Institute for Computing, Jülich, NIC Series, Vol.3, ISBN 3-00-005834-6, pp. 285-305, 2000.

[12] Thomsen, Rene, *Flexible ligand docking using evolutionary algorithms: investigating the effects of variation operators and local search hybrids*, Biosystems, Vol. 72, pp. 57–73, Nov. 2003.

[13] Morris, Garrett M. and Goodsell, David S. and Halliday, Robert S. and Huey, Ruth and Hart, William E. and Belew, Richard K. and Olson, Arthur J., *Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function*, Journal of Computational Chemistry, Vol. 19, pp. 1639–1662, Jan. 1999.

[14] David R. Westhead and David E. Clark and Christopher W. Murray, *A comparison of heuristic search algorithms for molecular docking.*, Journal of Computer-Aided Molecular Design, Vol. 11, pp. 209–228, 1997.

[15] Jay W. Ponder and David A. Case, *Force Fields for Protein Simulations.*, Advances in Protein Chemistry, Vol. 66, pp. 27–85, 2003.

[16] Alexandru-Adrian Tantar and Nouredine Melab and El-Ghazali Talbi and Bernard Toursel, *Solving the Protein Folding Problem with a Bicriterion Genetic Algorithm on the Grid.*, CCGRID, Vol. 2, ISBN 0-7695-2585-7, 43, 2006.

[17] A.-A. Tantar, N. Melab, E.-G. Talbi, B. Parent and D. Horvath, *A parallel hybrid genetic algorithm for protein structure prediction on the computational grid.*, Future Generation Computer Systems, Vol. 23, pp. 398–409, March 2007.

[18] Thom Vreven, Keiji Morokuma, Ödön Farkas, H. Bernhard Schlegel, Michael J. Frisch, *Geometry optimization with QM/MM, ONIOM, and other combined methods. I. Microiterations and constraints*, Wiley Periodicals Inc., J. Comput. Chem. 24: 760-769, 2003.

[19] Hideaki Kikuchi, Rajiv K. Kalia, Aiichiro Nakano, Priya Vashishta, Hiroshi Iyetomi, Shuji Ogata, Takahisa Kouno, Fuyuki Shimojo, Kenji Tsuruta, Subhash Saini, *Collaborative Simulation Grid: Multiscale Quantum-Mechanical/Classical Atomistic Simulations on Distributed PC Clusters in the US and Japan*, IEEE, 2002.

[20] Aiichiro Nakano, Rajiv K. Kalia, Priya Vashishta, Timothy J. Campbell, Shuji Ogata, Fuyuki Shimojo, Subhash Saini, *Scalable atomistic simulation algorithms for materials research*, SC2001 November 2001, Denver (c) 2001 ACM.

[21] B.J. Alder and Wainwright, *Phase Transition for a Hard Sphere System*, T.E.*J. Chem. Phys.* 27, 1208, 1957.

[22] B.J. Alder and Wainwright, *Studies in Molecular Dynamics I: General Method*, T.E.*J. Chem. Phys.* 31, 459, 1959.

[23] J.A. McCammon, B.R. Gelin, M. Karplus, *Nature (Lond.)* 267, 585, 1977.

[24] J. Thomas Ngo, Joe Marks, *Computational Complexity of a Problem in Molecular-Structure Prediction*, *Protein Engineering*, 5(4):313-321, 1992.

[25] Pierluigi Crescenzi and Deborah Goldman and Christos H. Papadimitriou and Antonio Piccolboni and Mihalis Yannakakis, *On the complexity of protein folding*, Journal of Computational Biology, vol. 5 no. 3, pp. 423-466, 1998.

[26] Pierre-Yves Calland, *On the structural complexity of a protein*, Protein Engineering, vol. 16 no. 2, pp. 79-86, 2003.

[27] E.E. Lattman, CASP4 *Proteins*, 44:399, 2001.

[28] R. Bonneau, J. Tsui, I. Ruczinski, D. Chivian, C.M.E. Strauss, D. Baker Rosetta in CASP4: progress in ab-initio protein structure prediction. *Proteins*, 45:119-126, 2001.

[29] A. Rabow, H. Scheraga, *Protein Science*, 5:1800-1815, 1996.

[30] N. Krasnogor, W. Hart, J. Smith, D. Pelta, *Protein structure prediction problem with evolutionary algorithms*, In Proc. of the Genetic and Evolutionary Computation Conference, 1999.

[31] F.C. Bernstein, T.F. Koetzle, G.J. Williams, E. Meyer, M.D. Bryce, J.R. Rogers, O. Kennard, T. Shikanouchi and M. Tasumi, *The protein data bank: A computer-based archival file for macromolecular structures*, J. Mol. Biol. 112 (1977), pp. 535-542.

[32] Pettersen, E.F., Goddard, T.D., Huang, C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C., and Ferrin, T.E., *UCSF Chimera - A Visualization System for Exploratory Research and Analysis*, J. Comput. Chem. 25 (2004), pp. 1605–1612.

[33] A.L. Islas, C.M. Schober, *Multi-symplectic integration methods for generalized Schrödinger equations*, Future Generation Computer Systems 19 (2003) 403-413.

[34] Brian E. Moore, Sebastian Reich, *Multi-symplectic integration methods for Hamiltonian PDEs*, Future Generation Systems 19 (2003) 395-402.

[35] H. Van de Waterbeemd, R.E. Carter, G. Grassy, H. Kubinyi, Y.C. Martin, M.S. Tute, P. Willett, *Glossary of terms used in computational drug design*, Pure & Appl. Chem., Vol. 69, No. 5, pp. 1137-1152, Great Britain, 1997.