A Grid-Based Hybrid Hierarchical Genetic Algorithm for Protein Structure Prediction

Alexandru-Adrian Tantar, Nouredine Melab, and El-Ghazali Talbi

Abstract. A hybrid hierarchical conformational sampling evolutionary algorithm is presented in this chapter, relying on different parallelization models. After first reviewing general conformational sampling aspects, *e.g.* existing approaches, complexity matters, force field functions, a focus is considered for the protein structure prediction problem. Furthermore, having as basis the highly multimodal nature of the energy landscape structure, a hybrid evolutionary approach is defined, enclosing conjugate gradient and adaptive simulated annealing enforced components. An insular model is employed, the conformational sampling process being conducted on a collaborative basis. Nonetheless, although low energy conformations were obtained, no close to native conformations were attained. Consequently, a higher complexity hierarchical paradigm has been constructed, with incentive following results.

1 Introduction

Entitled as *a silent revolution* in a recollection of the last century preeminent discoveries [37, 21], contemporary computational biology extends over mathematical modeling, molecular biology and computer science, comprising inter-linked scientific research disciplines. *In silico* conformational modeling and simulation, although computationally expensive, ascertained significant advancements in the entire life sciences spectrum [50, 47]. Conclusive examples may be found by reminding the completion of the human genome mapping, attained this decade, Human Genome Project [56, 36], the Folding@Home project [42, 49] fighting cancer, and Alzheimer's disease, etc. Nonetheless, no advancement on the current state

Alexandru-Adrian Tantar · Nouredine Melab · El-Ghazali Talbi
INRIA Lille - Nord Europe, Project-Team DOLPHIN, Room 211 bis,
Building A, 40, Avenue Halley, Parc Scientifique de la Haute-Borne,
59655 Villeneuve d'Ascq Cedex, France
e-mail: `Alexandru-Adrian.Tantar@inria.fr`,
`{Nouredine.Melab,El-Ghazali.Talbi}@lifl.fr`

F. Fernández de Vega, E. Cantú-Paz (Eds.): Paral. and Distrib. Comp. Intel., SCI 269, pp. 291–319.
springerlink.com

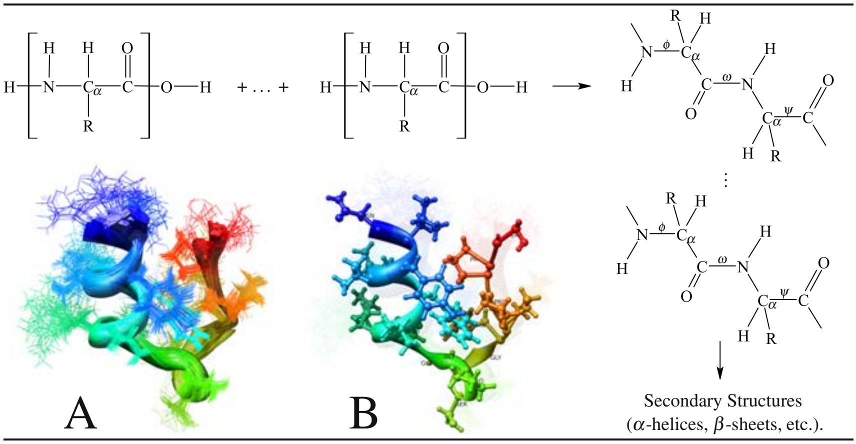


Fig. 1. First row: a $NC_{\alpha}C$ back-bone structure resulting as a combination of multiple amino-acids; secondary and ternary structures follow. R designates the specific amino acid's *side chain* characteristic, ω, Φ and Ψ relate to dihedral angles. **Second row, A & B**: tryptophan-cage protein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

of the art is possible unless extensive grid computing is employed. At the core of avant-garde conformational sampling and molecular dynamics simulations, grid computing nowadays offers an unprecedented *sine qua non* computational support [20, 35], in this context, connecting the computational biology and computer science domains.

The foundations of this chapter address *ab initio* conformational sampling [40], having Protein Structure Prediction, further referred to as PSP, as a reference topic. Of particular interest for the parallel grid computing domain, the problem consists in determining the *ground-state* conformation of a specified protein, given its amino-acids sequence – the *primary structure*. In this context, the ground-state conformation term designates the associated tridimensional native form, referred to as *zero energy structure*. From a structural point of view, proteins are complex organic compounds composed of amino-acid residue chains joined by peptide bonds – for a graphical illustration, please refer to Fig. 1. Assenting to a concise definition, conformational sampling entails the exploration of an exponentially large space of possible configurations [41, 13, 9], derived on the basis of an extensive number of degrees of liberty, which define the flexibility of the under study conformation. An energetically stable configuration has to be computationally predicted with the support of an underlying, generally highly multimodal, force field function [45]. Of quintessential impact and reinforced by the *in vivo* realm ubiquitousness of proteins, the intrinsic relation connecting the structure of a protein and the corresponding biological function determines fundamental consequences for computer assisted drug design, the understanding of immune response mechanisms, etc. In addition, computational modeling and prediction offer an alternative to laboratory *in vitro* experimentation, unfeasible for large domain analysis.

For the herein presented conformational sampling study a paradigm combining an Evolutionary Algorithm (EA) and an Adaptive Simulated Annealing (ASA) technique is considered – to be further detailed in the following sections. A study comprising different local search algorithms and outlining the efficacy of the ASA method on several benchmark conformations was previously presented in [55]. In addition, an extensive analysis of different intensification and diversification operators has been presented in [54]. EAs are stochastic search iterative techniques, with a large area of application – epistatic, multimodal, multicriterion and highly constrained problems [8]. A direct subclass of the EAs, 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 in this context to a part of the problem's landscape). At the opposite extreme, the class of the different Simulated Annealing [34] algorithms presented in the literature, further denoted as *SA*s, offers weak ergodicity optimization techniques capable of dealing with multimodal functions of a large nonlinearity and discontinuity degree. Simulated annealing algorithms were developed by Kirkpatrick [34] as a generalization of the Metropolis Monte Carlo techniques [39], including as extension a temperature schedule which offers an improved control over the acceptance rate. The underlying paradigm simulates metal recrystallization in the process of annealing, the entropy of an initially disordered system being adiabatically reduced to low entropy states while maintaining at each step a thermodynamic equilibrium. The SAs represent a viable alternative to gradient based local search methods, being less prone to getting trapped in local minima. Furthermore, the implementation of an SA algorithm does not impose complex development constraints – as a counterpart and as opposed to EAs, simulated annealing techniques are extensively sequential in their nature thus being difficult to parallelize.

Furthermore, the currently available computational resources allow for higher complexity algorithmic constructions, rendering possible the design of hierarchical parallel and distributed approaches. Nonetheless, a complex algorithmic underlying layer has to be unfolded in order to effectively exploit the existing computational resources. A transparent deployment has to be ensured, endorsing large-scale distributed applications to be expanded over geographically dispersed clusters. The parallel construction of the here considered approaches is sustained by an MPI [23] based version of ParadisEO [7, 8], a framework dedicated to the reusable design of parallel hybrid meta-heuristics. A broad range of features is provided by the framework, including EAs support, local search methods, parallel and distributed models, hybridization mechanisms, etc. For a complete overview of the existing dedicated frameworks on parallel and grid specific metaheuristcs refer to [10, 8, 51, 1, 7].

The contents of this study inscribe in the context of *ANR Dock – Conformational Sampling and Docking on Computational Grids*[1], designated under the *Docking@Grid* acronym, a French National Research Agency three years funded project, scheduled to end by fall 2009. Encompassing distinct areas of expertise, the foundations of the project are set on the complementarity of the participant research teams and laboratories, specifically, (1) DOLPHIN, INRIA Lille – Nord Europe, Fundamental Computer Science Laboratory of Lille, LIFL, (2) Biology Institute of Lille, IBL CNRS/INSERM and (3) Life Sciences Division, CEA/iRTSV – Grenoble. As a final phase of the project, an *in vitro* biological validation of the attained results will be conducted under the competences of the Life Sciences Division, CEA. Note that all presented experimentations were performed on Grid'5000, a nation-wide computational grid, consisting of almost 5000 computational cores, shared in a network of nine academic centers. Conformational sampling results are reported on the basis of a large number of deployments, with up to almost 1000 computational cores.

The remainder of this chapter is organized as follows. An introduction discussing in brief protein structure prediction aspects is offered in Section 2, followed by an incremental presentation of the considered algorithmic components in Section 3. Encoding and evaluation function details are discussed, the formal basis of a conjugate gradient and of an adaptive simulated annealing algorithm being illustrated. A first hybrid parallel approach is afterwards introduced, implementation and execution environment details being also presented. As part of Section 4 the employed benchmark conformations are outlined, finally, experimental outcomes being discussed. As entailed by the drawn conclusions, a hierarchical parallel algorithm is proposed, addressing minima characterization issues – definition details and results are given. Conclusions and further directions are finally drawn.

2 Protein Structure Prediction

As outlined in the introduction, the PSP problem consists in determining the ground-state conformation of a specified protein, given its amino-acids sequence. 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 modeling solvent derived effects through dielectric constants and continuum model based terms – a more detailed, force field oriented discussion is presented in a following section. 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 [16, 58, 40].

Accurate mathematical models, describing molecular systems, are formulated upon the *Schrödinger* equation [16], which makes use of molecular wavefunctions

[1] http://dockinggrid.gforge.inria.fr

for modeling the spatio-temporal probability distribution of the constituent entities. Nonetheless, although offering the most accurate approximation, the *Schrödinger* equation cannot be solved exactly for more than two interacting particles [16, 58]. At the opposite extreme, *empirical methods* rely upon molecular dynamics (*classical mechanics based methods*), and were introduced by Alder and Wainwright [2, 3]. Empirical methods do not make use of the quantum mechanics formalism, relying solely upon classical Newtonian mechanics, *i.e.* Newton's second law, and often represent the only applicable methods for large molecular systems, namely, proteins and polymers. After more than a decade protein simulations were initiated on bovine pancreatic trypsin inhibitor – BPTI [38].

Considering complexity aspects, 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 determining the ground-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 [13, 9]. As a conclusion, no simulation or resolution is possible unless extensive computational resources are used – it may be inferred that no polynomial time resolution is achievable if no or less *a priori* knowledge is employed.

For a comprehensive introductory article on the structure of proteins and related aspects please consult [40, 12]; a glossary of terms is also available in [57]. In addition, an extended referential resource for protein structural data may be accessed through the Brookhaven Protein Data Bank[2] [4].

3 A Parallel Hybrid Metaheuristic for the PSP

The exploration and intensification capabilities of the EAs do not suffice as paradigm, when addressing rough molecular energy function landscapes. Small variations of a torsional angle value may generate extremely different individuals, with respect to the fitness function. As a consequence, a nearly optimal configuration, considering the torsional angle values, may have a high energy value, and thus, it may not be taken into account for the future iterations of the algorithm. In order to correct the above exposed problem, a local search based method may be applied as a refinement step, alleviating the drawbacks determined by the conformation of the landscape – thus, a *Lamarckian* optimization technique is constructed.

3.1 Encoding of the Conformations and the Force Field Function

The algorithmic resolution of the PSP, in heuristic context, is directed through the exploration of the molecular energy surface. The sampling process is performed

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

by altering the structure of the under study conformation, *i.e.* backbone structure, associated torsional angles, etc., in order to obtain different structural variations. With implications over the sampling methodology, different encodings have been mentioned in literature. The trivial approach would consist of using a direct coding of the atomic Cartesian coordinates [46]. Nonetheless, as a main disadvantage of direct encoding based representations, filtering and correcting mechanisms are required, inducing a non-negligible overhead. Different other models were developed, including, for example, all-heavy-atom coordinates, C_α coordinates or backbone and residue atoms coordinates representations, hydrophobic/hydrophilic models [15], etc. For the herein described method, an indirect, less error-prone, torsional angle based representation has been preferred. More specifically, each conformation is coded as a vector of torsional angles values, denoted in the following as γ, $\gamma \overset{def}{=} (\gamma_1, \gamma_2, \cdots, \gamma_N)$, $\alpha_i \prec \gamma_i \prec \beta_i$, where N represents the liberty degree of the conformation and α_i, β_i stand as the lower and upper limits of the γ_i encoding value, $1 \le i \le N$. For a graphical illustration, please refer to Fig. 2.

$$
\begin{aligned}
E = & \sum_{bonds} K_b(b-b_0)^2 \\
+ & \sum_{angles} K_\theta(\theta-\theta_0)^2 \\
+ & \sum_{torsions} K_\phi(1-\cos n(\phi-\phi_0)) \\
+ & \sum_{Van\ der\ Waals} \frac{K^a_{ij}}{d^{12}_{ij}} - \frac{K^b_{ij}}{d^6_{ij}} \\
+ & \sum_{Coulomb} \frac{q_i q_j}{4\pi\varepsilon d_{ij}} \\
+ & \sum_{desolvation} \frac{Kq_i^2 V_j + q_j^2 V_i}{d^4_{ij}}
\end{aligned}
$$

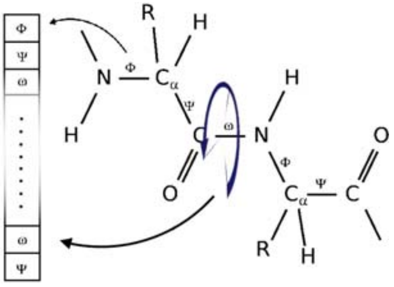


Fig. 2. Scoring function quantifying the inter-atomic interactions

The energy function, hereafter noted as E, is defined by relying on an independently calibrated *Consistent Valence Force Field (CVFF)* [14] based force field. The quantification of energy is performed by using empirical molecular mechanics, as depicted in Fig. 2. As classically employed for empirical force field definitions, a set of specific constants is associated with each interaction type, here denoted by K_b, K_θ, K_ϕ and K^a_{ij} for, respectively, bonds, angles, torsional angles and van der Waals interactions. An optimal value for the considered entity (bond, angle, torsion) is introduced through a corresponding $(A - A_0)$ equation term, where A, A_0 specify the sampled value, respectively the *a priori* experimentally determined optimal value. More specific, for the herein example, b represents bond lengths, θ angular values, ϕ torsional angles and q_a, d_{ij} and V_p the electrostatic charge associated to given atoms, the distance between the i and the j atoms, respectively, a volumetric

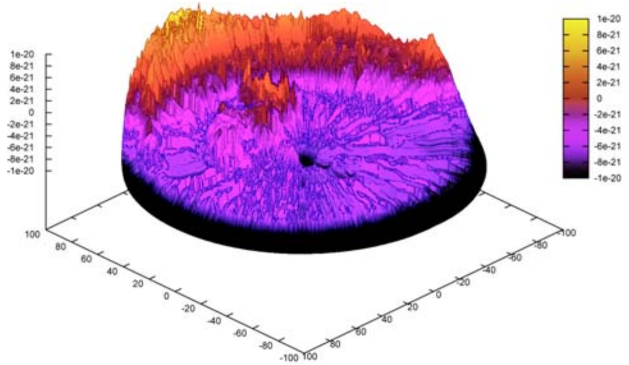


Fig. 3. Free energy surface for the *tryptophan-cage* protein around a deep optimum conformation. High energy points are depicted in light colors, the low energy points resulting in darker areas.

measure for the p atom. No further details are here included as being out of scope for the herein study – please refer to [45] for additional information.

An example of free energy surface representation for the *tryptophan-cage* is given in Fig. 3. The lighter areas of the surface correspond to high-energy conformations. The sampling values used for constructing the representation were computed by employing the Gibbs free energy over an ensemble of locally sampled conformations E_i – refer to Horvath *et al.* [26] for additional references and details:

$$G = -kT\left[\sum \exp\left(-\frac{E_i}{kT}\right)\right]$$

where $k = 1.3806504(24) \times 10^{23}\, J\, K^{-1}$ designates the constant of Boltzmann, offering, in numerical form, a connection between the molecular level and macroscopic observed effects, expressed as an ensemble result. Further, T represents a temperature term, the ensemble being equivalent to approx. 0.6 $kcal\ mol^{-1}$ at $300K$ – introductory notions and references were presented in [26].

An extensive discussion reviewing the force fields designed for protein simulations, with in-depth details, is offered in [45]. The first part of the study covers the evolution of the force fields over the last three decades, discussing various formulations which include the *Amber*, *CHARMM* and *OPLS* force fields.

3.2 Conjugate Gradient Local Search

The conjugate gradient method, an extension of the steepest gradient descent method, has been independently developed in the early 1950's by Eduard Stiefel and Magnus R. Hestenes, with the cooperation of J.B. Rosser, G. Forsythe and L. Paige [25]. Depending on the setup of the parameter values, the method converges to the closest local minimum, hence not being well adapted for the global optimization of large highly multimodal functions. References of early works on more advanced conjugate gradient methods lead to the publications of R. Fletcher, C.M. Reeves, M.J.D. Powell, E. Polak and G. Ribière [17, 18, 44], appeared a decade later.

For the rest of this section a *nonlinear* conjugate gradient approach is considered, simply referred to as *conjugate gradient*. If the force field based energy function E is continuous and differentiable in $\gamma_i \in \gamma, 1 \leq i \leq N$, the ∇E gradient is defined as a vector of partial derivatives:

$$\nabla E = \left[\frac{\partial E}{\partial \gamma_1} \; \frac{\partial E}{\partial \gamma_2} \; \cdots \; \frac{\partial E}{\partial \gamma_N} \right]^T \tag{1}$$

Hence, considering an iterative approach, at each iteration, the current point γ^+ can be updated by setting $\gamma^+ \leftarrow \gamma^+ - \tau_\varepsilon \nabla E_{\gamma^+}$, where the τ_ε step has a *positive, small enough value, adapted for the function under study*, and where ∇E_{γ^+} denotes the gradient vector computed at the γ^+ solution point. Compared to the steepest descent method, the conjugate gradient algorithm considers not only the gradient vector at the current point but also the previous directions. Hence, at each iteration k of the algorithm, the $\gamma^+_{\{k\}}$ solution is updated as follows:

$$\gamma^+_{\{k+1\}} \leftarrow \gamma^+_{\{k\}} - \tau_\varepsilon \delta_k, \; \delta_k \overset{def}{=} \begin{cases} \nabla E_{\gamma^+_{\{k\}}}, \text{ if } k = 0 \\ \nabla E_{\gamma^+_{\{k\}}} - \xi_k \delta_{k-1}, \text{ if } k > 0 \end{cases} \tag{2}$$

The algorithm is mainly based on the ξ_k factor which, in terms of convergence, defines the behavior of the method. Classically employed forms of the ξ_k term are defined as a combination of the previously computed gradient vectors, including different formulations, *e.g.* Fletcher-Reeves, Hestenes-Stiefiel, Polak-Ribière, etc. [17, 18, 44].

The basic pseudo-code of the nonlinear conjugate gradient method is given in Algorithm 1. The first step of the algorithm, for $k = 0$, is similar to the steepest descent method, the following steps relying in addition on the previously computed gradient vectors. For the herein example, the *Fletcher-Reeves* form has been chosen for the ξ_k term. Further, having computed the ξ_k, δ_k terms (lines 3-8), a line search is applied in order to minimize $E(\gamma^+_{\{k\}} - \tau_\varepsilon \delta_k)$ by varying the τ_ε factor. For details on line search algorithms refer to [48]. Different stopping criteria can be chosen – common approaches consider an *a priori* specified threshold for the gradient vectors (*e.g.* the absolute value of all the components of the $\nabla E_{\gamma^+_{\{k\}}}$ gradient vector falling below 1.0e-5) or for the attained improvement. In addition, a maximum number of iterations can be imposed.

The here employed component relies on *analytical gradient formulation*, the exploration being conducted on fine-grain landscape information. As a consequence, the method may not be well adapted for dealing with the conformational sampling landscape particularities, offering nevertheless fine-tuning minimization advantages.

Algorithm 1. Nonlinear Conjugate Gradient Pseudo-Code.

```
Set $k \leftarrow 0$, $\gamma^+_{\{k\}} \leftarrow \gamma$ ($\gamma$, $\gamma^+_{\{k\}}$ represent the current and the best known solution at iteration $k$, respectively)
repeat
    if $k = 0$ then
        Set $\delta_k \leftarrow \nabla E_{\gamma^+_{\{k\}}}$
    else
        Set $\xi_k \leftarrow \frac{\nabla E^T_{\gamma^+_{\{k\}}} \nabla E_{\gamma^+_{\{k\}}}}{\nabla E^T_{\gamma^+_{\{k-1\}}} \nabla E_{\gamma^+_{\{k-1\}}}}$
        Set $\delta_k \leftarrow \nabla E_{\gamma^+_{\{k\}}} - \xi_k \delta_{k-1}$
    end if
    Find $\tau_\varepsilon$ minimizing $E(\gamma^+_{\{k\}} - \tau_\varepsilon \delta_k)$
    Set $\gamma^+_{\{k+1\}} \leftarrow \gamma^+_{\{k\}} - \tau_\varepsilon \delta_k$
    Set $k \leftarrow k + 1$
until $|E_{\gamma^+_{\{k\}}} - E_{\gamma^+_{\{k-1\}}}| < \tau_{prec}$ or $\nabla E_{\gamma^+_{\{k\}}} < \tau_{lb}$.
```

3.3 Adaptive Simulated Annealing Algorithm

Classical SA algorithms [34] rely on a Boltzmann sampling distribution, including as components a probability density function of the state space, $g(\gamma)$, an acceptance probability function $h(\Delta E)$ and an annealing schedule $T(k)$. Gradient information is not employed in classical constructions of the algorithm. The annealing schedule is defined over a number of discrete steps. The acceptance function has the role of quantifying the probability of performing a transition from an E_k energy state to a new state with energy E_{k+1}. Classical definitions make use of the Metropolis criterion [39] which makes use of the Boltzmann probability density function:

$$h(\Delta E) = \frac{e^{-E_{k+1}/T}}{e^{-E_{k+1}/T} + e^{-E_k/T}} = \frac{1}{1 + e^{\Delta E/T}} \cong e^{-\Delta E/T},\ \Delta E = E_{k+1} - E_k \qquad (3)$$

Given a Gaussian-Markov system, with a probability density state space function $g(\Delta\delta) = (2\pi T)^{-N/2} e^{-\|\Delta\delta\|^2/(2T)}$, for an appropriate initial temperature T_0, the global minimum can be found if the temperature is decreased no faster than $T(k) = T_0/\ln k$. Low discrimination between solutions is considered in the initial phases of the algorithm, the method acting like a global search exploration. Near the final phases, local search is performed at low temperatures. Nonetheless, the main difficulty in designing a Boltzmann SA consists in determining the starting temperature as well as an efficient schedule for the problem under study. In practice, a $T_0/\ln k$ schedule does not offer a fast enough annealing. While no longer guaranteeing asymptotic convergence, exponentially decreasing schedules are preferred instead, *e.g.* $T(k) = e^{((c-1)k)} T_0, T(k) = c\ T(k-1), k \geq 1$, with $0 \ll c < 1$, $c \approx 0.98$.

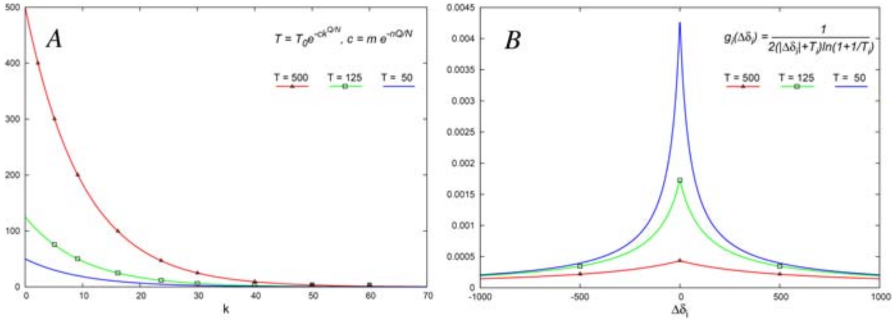


Fig. 4. **A** – Temperature decrease as defined for the ASA T_i schedules. **B** – ASA probability density function.

The Adaptive Simulated Annealing (ASA), an enhanced version of the basic SA algorithm, has been initially presented in the work of Ingber [27, 28, 31, 29, 30]. ASA exploits the characteristics of a specifically designed generating function allowing for an exponentially faster annealing process, as compared to the classical Boltzmann distribution based approach. In addition to employing a temperature parameter for the acceptance function, hereafter noted as T_a, distinct T_{ik_i} parameters and probability density functions are associated to each of the control parameters.

In the following, for simplicity, T_{ik_i} is denoted as T_i, with T_{i0} representing the initial temperature of the T_i schedule. As detailed in Ingber's articles, by considering $T_i \overset{def}{=} T_{i0}e^{(-c_i k_i^{Q_i/N})}$, with $c_i = m_i e^{-n_i Q_i/N}$, asymptotic convergence is attained. The m_i, n_i control parameters can be employed for adjusting and fine-tuning the algorithm for a specific problem. While for $Q_i > 1$ (quenching factors) an accelerated exploration is performed, the asymptotic convergence proof no longer stands, the algorithm being prone to getting trapped in local minima.

The adaptive features of the algorithm are determined by sensitivity derived factors, namely the T_a, T_i temperature schedules, which are employed in deciding over and controlling the acceptance, respectively, generation of new solutions – refer to Fig. 4 for a graphical depiction. The considered factors enclose descriptive information over the structure of the landscape to explore. The ASA generation function is defined over a set of uniform random variables, $u_i \in U[0,1]$, as exposed below:

$$\gamma_i^{k+1} = \gamma_i^k + \delta_i(\beta_i - \alpha_i), \text{ where } \delta_i \text{ is defined as:} \tag{4}$$

$$\delta_i = sgn(u_i - 0.5)\ T_i[\ (1 + 1/T_i)^{|2u_i - 1|} - 1\ ], \delta_i \in [-1, 1] \tag{5}$$

In the herein context α_i, β_i denote the lower, respectively upper limit of the γ_i encoding value. Acceptance is performed according to the Metropolis criterion. After a specified number of accepted solutions, *reannealing* takes place, adjusting the algorithm's parameters. Gradient based sensitivities are used for updating the acceptance temperature, T_a, the T_i temperature schedules and the k_i step indexes. No restriction is imposed on defining different sensitivity measures, other than gradient based ones. Considering γ^+, γ_a the best known solution and the last accepted solution, respectively, at a given step of the algorithm, for each component $\gamma_i^+ \in \gamma^+$, the associated sensitivity s_i is computed, to be employed in the reannealing step:

$$s_i = \left| \frac{\partial E}{\partial \gamma_i^+} \right|, \; s_{max} = \max_{1 \leq i \leq N} s_i, \; T_i' = \frac{s_{max}}{s_i} T_i, \; k_i' = \left[\frac{\ln(T_{i0}/T_i')}{c_i} \right]^N \tag{6}$$

$$T_{a0} = E(\gamma_a), \; T_a = E(\gamma^+), \; k = \left[\frac{\ln(T_{a0}/T_a)}{c} \right]^N \tag{7}$$

The main phases of the ASA method are depicted hereafter in Algorithm 2. The quenching factors Q, Q_i, the initial temperatures T_{a0}, T_{i0}, the m_i, n_i control parameters and the k, k_i step indexes are initialized in the first two lines. Lines 3 and 4 set the best known and the last accepted solutions which, for this step, are identical to the initial solution. The algorithm includes a main exploration loop (lines 5-32) and a secondary internal loop for generating new solutions (lines 6-11). Newly generated solutions are accepted based on the Metropolis criterion (line 13), the reannealing of the temperature schedules (lines 18-24) being performed at a pre-specified number of accepted solutions. At the end of each iteration of the main loop, step indexes and temperature schedules are updated in order to reflect the advancement of the algorithm (lines 26-31). The algorithm finishes after a fixed number of iterations or at a pre-specified threshold of iterations with no improvement.

As opposed to classical SA algorithms, the influence of the initial parameters over the exploration is alleviated, the annealing schedule being adaptively modified as to reflect the current exploration stage. While not directly employed in generating new solution points, gradient information is used for modifying the factors which intervene in the sampling process, consequently avoiding the direct disadvantages of steepest descent gradient based approaches. An improved scaling is offered as factors are independently modified on each dimension.

As a final remark, although including adaptive mechanisms, a large number of fine-tuning parameters are included. The effective calibration of the algorithm does not stand simplified tractableness basis, demanding for advanced parameter optimization. A possible approach, as suggested by Ingber, consists in using the ASA algorithm *per se* as a control parameters optimization component. Nevertheless, considering that performance evaluations require for the algorithm to be executed on one or multiple benchmarks, a high computational impact is implied. Subsequently, parallel support is required, entailing the optimization process to be carried on the support of a scalable distributed algorithm. Therefore, as part of the herein work, a meta-evolutionary algorithm has been employed in order to answer the mentioned concerns [53], given that ASA does not comport a high parallelization affinity.

A detailed description of the ASA algorithm, including comparison, test case studies and applications is available in the work of Ingber [30, 29, 27, 28, 31].

Algorithm 2. ASA Pseudo-Code.

```
Set $c$, $Q$, $k = 0$, $T_{a0} = E(\gamma)$
Set $Q_i$, $m_i$,$n_i$, $c_i = m_i e^{-n_i Q_i/N}$, $k_i = 0$, $T_{i0} = 1.0$, for $1 \le i \le n$
Set $\gamma^+ \leftarrow \gamma$ ($\gamma$, $\gamma^+$ represent the current and the best known solution, respectively)
Set $\gamma_a \leftarrow \gamma$ ($\gamma_a$ represents the last accepted solution)
repeat
    for all $\gamma_i \in \gamma,\ 1 \le i \le N$ do
        repeat
            $\delta_i \leftarrow sgn\left(u_i - \frac{1}{2}\right)\ T_i \left[ \left(1+\frac{1}{T_i}\right)^{|2u_i-1|} - 1 \right]$, $u_i \in U[0,1]$
            $\gamma_i' \leftarrow \gamma_i + \delta_i(\beta_i - \alpha_i)$
        until $\alpha_i < \gamma_i' < \beta_i$
    end for
    $\Delta E \leftarrow E(\gamma') - E(\gamma)$
    if $u < e^{-\Delta E/T_a}$, $u \in U[0,1]$ then
        Accept $\gamma'$ as the current solution: $\gamma \leftarrow \gamma'$, $\gamma_a \leftarrow \gamma'$
        if $E(\gamma') < E(\gamma^+)$ then
            Update the best known solution: $\gamma^+ \leftarrow \gamma'$
        end if
        if reannealing limit reached then
            for all $k_i,\ T_i,\ 1 \le i \le N$ do
                $s_i \leftarrow \left| \frac{\partial E}{\partial \gamma_i^+} \right|,\ \gamma_i^+ \in \gamma^+,\ s_{max} \stackrel{def}{=} \max_{1 \le i \le N} s_i$
                $T_i \leftarrow \frac{s_{max}}{s_i} T_i,\ k_i \leftarrow \left[ \frac{\ln(T_{i0}/T_i')}{c_i} \right]^N$
            end for
            $T_{a0} \leftarrow E(\gamma_a)$, $T_a \leftarrow E(\gamma^+)$, $k \leftarrow \left[ \frac{\ln(T_{a0}/T_a)}{c} \right]^N$
        end if
    end if
    for all $k_i,\ T_i,\ 1 \le i \le N$ do
        $k_i \leftarrow k_i + 1$
        $T_i \leftarrow T_{i0} e^{(-c_i k_i^{Q_i/N})}$
    end for
    $k \leftarrow k + 1$
    $T_a \leftarrow T_{a0} e^{(-ck^{Q/N})}$
until stopping criterion met.
```

3.4 Hybrid Parallel Genetic Algorithm

Evolutionary algorithms rely on a set of intensification vs. diversification directed operators for iteratively evolving an initial randomly generated population. At each iteration of the algorithm (generation), a selection process is conducted, the fitness of each individual being evaluated on a problem specific fitness function, *i.e.* the force field function for the herein case. The pseudo-code in Alg. 3 exposes the generic structure of an EA. Following a broad classification perspective, the main

Algorithm 3. EA Pseudo-Code.

```
$t \leftarrow 0$
Generate($P(0)$)
**while** $\neg$Termination_Criterion($P(t)$) **do**
    Evaluate($P(t)$)
    $P'(t) \leftarrow$ Selection($P(t)$)
    $P'(t) \leftarrow$ Apply_Reproduction_Ops($P'(t)$)
    $P(t+1) \leftarrow$ Replace($P(t), P'(t)$)
    $t \leftarrow t+1$
**end while**
```

subclasses of EAs are the Genetic Algorithms (GAs), Evolutionary Programming, Evolution Strategies, etc. In this context, a *genotype* represents the raw encoding of the individuals while the *phenotype* offers the equivalent representation features. At each generation, the genotype of a selected set of individuals is altered by applying mutation and crossover operators in order to intensify the exploration over an interest region or for diversification purposes as to avoid a premature convergence. Last, offsprings are reinserted in the population according to a pre-specified criterion.

The herein considered GA was parallelized in a hierarchical manner, including, in addition to the exposed basic pseudo-code, three levels of parallelism – the insular model, the parallel evaluation of the population and the synchronous multi-start model. A conceptual simplistic depiction of the different models is offered in Fig. 5. At execution time, a set of identically configured algorithms is deployed, independently evolving a local assigned population whereas fitness evaluations are dispatched on remote worker nodes. A stochastic tournament strategy approach is used for the selection and the replacement phases of the algorithm. Furthermore, in addition to classical simple diversification and intensification operators, *e.g.* random mutation, two-points crossover, each algorithm encloses an analogous set of conjugate-gradient extended operators. The defined alternate set of operators function by first applying the enclosed mutation, respectively, crossover standard mechanisms, the resulting offspring(s) being further refined by the local search component. Embedding the standard and the gradient enhanced version, a combined operator is provided, allowing for a selective, rate dependent, application of the internal sub-operators, *e.g.* allowing for the standard mutation operator to be applied on 90% of the subjected solutions, respectively, for the extended operator on the remaining 10%. An eloquent practical exemplification is found when considering a high-energy barrier surrounded optimum conformation. With no refinement, a close to optimum solution is subject to attain, with a high probability, an elevated fitness energy. Consequently, the solution, although encoding valuable information, exhibits a high probability of being discarded, in the selection process. A balanced design has to be assured, nevertheless, *e.g.* by specifying appropriate operator rates, gradient steps, etc., as to avoid a potential premature convergence of the algorithm. Additionally, a refined local optimum solution stands as a key minima representative over the surrounding high-energy conformations, locally characterizing the afferent landscape region.

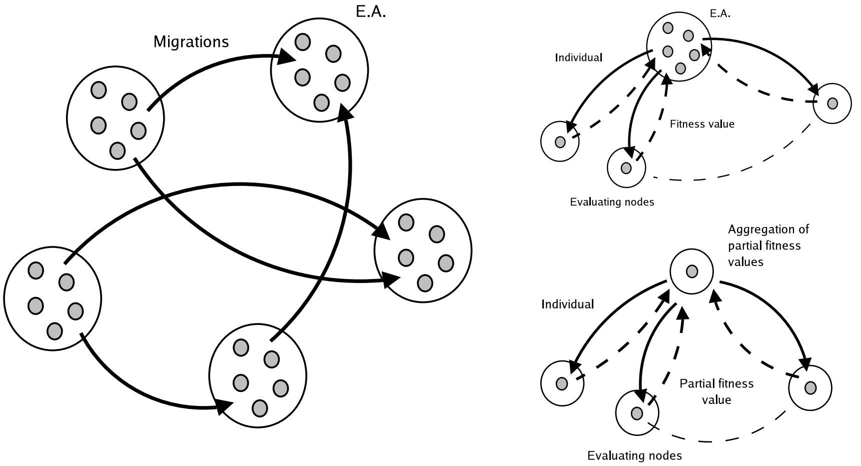


Fig. 5. The three main EAs parallelization models: island (a)synchronous cooperative model – left side of the figure, parallel evaluation of the population and distributed evaluation of a single solution – right side, upper, respectively lower part.

A synchronous ASA multi-start local search refinement phase is additionally interposed, succeeding the completion of a fixed number of iterations. Independent local explorations are simultaneously launched, for each of the to be refined solutions, obtained by random selection out of the local population. Further, allowing for convergence and diversity control, an asynchronous inter-islands exchange of genetic material is performed, at a predefined number of iterations. A cyclic, ring topology model communication pattern is set, *i.e.* accepting only one source and one destination per island. The specified migration model, allows for a coordinated global convergence, as determined by the migration frequency, number of exchanged solutions, etc., whilst reducing the external impact on the local island exploration process. A strong local attractor is required to cycle the entire ring, through multiple selection steps, before attaining global acceptance. Emigrant solutions are retrieved by means of a stochastic tournament selection, at the opposite end, the worst individuals in the target population being replaced by immigrant solutions. Survival of the best individual is assured by a weak-elitism scheme. For each local search refinement and migration phase, one tenth, respectively, one sixth of the population, is subject to undergo the local optimization, respectively, information exchange process.

Note that, except for selection and replacement, all operations, including the local search enhanced operators, are performed in parallel by delegation to worker nodes. A detailed discussion of the ParadisEO framework architecture and the afferent components developed in order to sustain the construction of the herein presented algorithmic model, execution roles, communication topologies, etc., is presented in [7, 8].

3.5 ParadisEO Based Implementation

ParadisEO[3], initially designed and developed by Sebastien Cahon [7, 8], is an extendible open source C++ framework based on a clear conceptual separation of the meta-heuristics from the problems they are intended to solve. The *EO* suffix stands for Evolving Objects, the framework being basically an extension of the Evolving Objects (EO) [33] LGPL C++ open source project, the result of an European joint work [33]. EO includes a paradigm-free Evolutionary Computation library, dedicated to the flexible design of EAs through evolving objects, superseding the most common dialects (Genetic Algorithms, Evolution Strategies, Evolutionary and Genetic Programming).

Furthermore, most common parallel/distributed models, *i.e.* synchronous island model, synchronous multi-start, etc., are provided in the ParadisEO-PEO module (Parallel EO). A portable design over distributed-memory machines and shared-memory multi-processors is offered, relying on standard libraries such as Message Passing Interface (MPI) [23, 24] and POSIX Threads (PThreads) [6]. A transparent exploitation of the enclosed parallel models, in (non) dedicated parallel environments, is assured. Nevertheless, with the continuous evolution of the distributed computing grids and with the perpetuous development of the available computing resources, there is a *sine qua non* requirement to pass beyond the physical design of the grids. Extending the existing framework, in order to offer a grid-enabled ParadisEO implementation, demands for a Grid middleware layer and a Grid Application Programming Interface. Furthermore, an infrastructure interface is required, providing communication and resource management tools. The here adopted approach consists in using the Globus Toolkit [20, 19] computing system as a Grid Infrastructure - an outline is presented in [52].

A layered architecture of the ParadisEO framework is presented in Fig. 6. 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 standard MPI based programming. At this level virtually any standard conforming MPI distribution may be placed as layer. The fourth and lowest level supplies communication and resource management services. A broad range of experimentations were conducted on employing the Globus Toolkit with MPICH/MPICH-G2 [23], MPICH-VMI [43] and OpenMPI [22].

With no exception, all tests have been deployed on the Grid'5000 (https://www.grid5000.fr) French nation-wide experimental computational grid, connecting several sites which host clusters of PCs interconnected by RENATER[4] (the French academic network). At this time, Grid'5000 is gathering more than 4000 computational cores with more than 100 Tb of non-volatile storage capacity, regrouping nine centers: Bordeaux, Grenoble, Lille, Lyon, Nancy, Orsay, Rennes,

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

[4] http://www.renater.fr

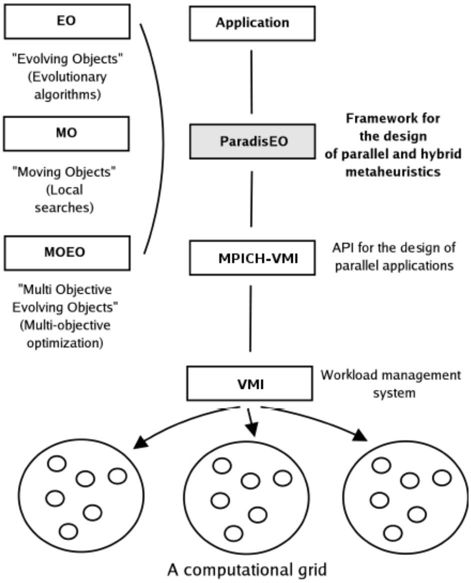

EO
"Evolving Objects" (Evolutionary algorithms)
MO
"Moving Objects" (Local searches)
MOEO
"Multi Objective Evolving Objects" (Multi-objective optimization)
Application
ParadisEO
Framework for the design of parallel and hybrid metaheuristics
MPICH-VMI
API for the design of parallel applications
VMI
Workload management system
A computational grid

Fig. 6. A layered architecture of ParadisEO.

Sophia-Antipolis, Toulouse. Following time dependent requirements and computational resources availability constraints, determined by the shared nature of the environment, experimentations were conducted on most of the Grid'5000 sites. As dictated by a per experiment demand, a varying number of resources has been used, ranging from a reduced number of computational cores, for tuning and prototyping purposes, to up to almost 1000 cores for the actual deployment and testing - see Table 1 for details.

Table 1. Environment details for a conformational sampling experimentation cumulating almost 1000 computational cores, over multiple clusters

Cluster/Site*	**CPUs**	**Cores**	**Architecture Details**
Azur/Sophia	59	118	Dual AMD Opteron^{TM} 2.0GHz/1MB/333MHz, 2GB RAM
Helios/Sophia	53	212	Quad Core AMD Opteron^{TM} 2.2GHz/1MB/400MHz, 4GB RAM
Sol/Sophia	27	108	Quad Core AMD Opteron^{TM} 2.6GHz/1MB/667MHz, 4GB RAM
Sagittaire/Lyon	60	120	Dual AMD Opteron^{TM} 2.0GHz/1MB/400MHz, 2GB RAM
Capricorne/Lyon	51	102	Dual AMD Opteron^{TM} 2.4GHz/1MB/400MHz, 2GB RAM
Orsay/Paris	152	304	Dual AMD Opteron^{TM} 2.4GHz/1MB/NA, 2GB RAM
Overall	**402**	**964**	

4 Experimental Outcomes

4.1 Conformational Sampling Benchmarks

Assessing conformational sampling algorithms requires to set a trade-off over the considered benchmarks. A first aspect to be considered regards complexity matters, *i.e.* reduced size conformations are of no interest (there is no need of determining the structure of a water molecule using computational grid resources) whilst highly complex molecules may be highly computationally restrictive (due to resource constraints, force field calibration limitations, etc.). A second aspect is defined on validation requirements - the crystallographic structure of the benchmark molecule has to be known in order to be able to have performance evaluations.

The herein adopted molecular complexes for the conformational sampling algorithms assessment, are the *tryptophan-cage* (trp-cage - Protein Data Bank ID: 1L2Y), the *tryptophan-zipper* (trp-zipper - Protein Data Bank ID: 1LE1) and the *α-cyclodextrin*. *Tryptophan-cage* and *tryptophan-zipper* belong to the class of mini-proteins presenting particularly fast folding characteristics. *Cyclodextrins*, in α, β or γ conformations, with 6, 7, 8 glucose units, respectively, due to their toroidal structure, are important for drug-stability applications, being used as protectors against micro-environment interactions or as homogeneous distribution stabilizers, etc.

The selected benchmark conformations can be considered, to a certain extent, as being significant and representative as they include different structural patterns, hence, requiring a flexible enough algorithm to predict the different enclosed secondary structures. Refer to Fig. 7 for a graphical representation of the three molecular conformations. An equivalent schematic representation is also exposed in order to better illustrate the structural characteristics of each molecule (as the cyclic structure of α-cyclodextrin). The α-cyclodextrin molecule, while not being a protein, has been included in the study due to its particular cyclic structure. In addition, the addressed conformations, given the number of defined torsional angles, namely 64, 54, 73 angles for α-cyclodextrin, 1LE1, 1L2Y, respectively, offer the advantage of not requiring an extremely expensive energy evaluation computation time.

4.2 Execution Configuration and Outcomes

A ring insular model consisting of three algorithms has been deployed at run-time, each island evolving a fixed-size population of 300 solutions for 300 generations. No specific parameter tuning has been considered, the employed configuration being incrementally constructed in a series of trial executions. As previously outlined, combined mutation and crossover operators have been employed, *e.g.* the classical two-point crossover operator and the conjugate-gradient enhanced version, in mutual exclusive manner, with a 0.85, respectively, a 0.15 rate. Analogously, the mutation operators are applied with equal rates, for the classical and local search extended version, having an overall 0.05 probability. A selection rate of 0.75 has been set, with a 0.95 probability of accepting a better individual over a worse one.

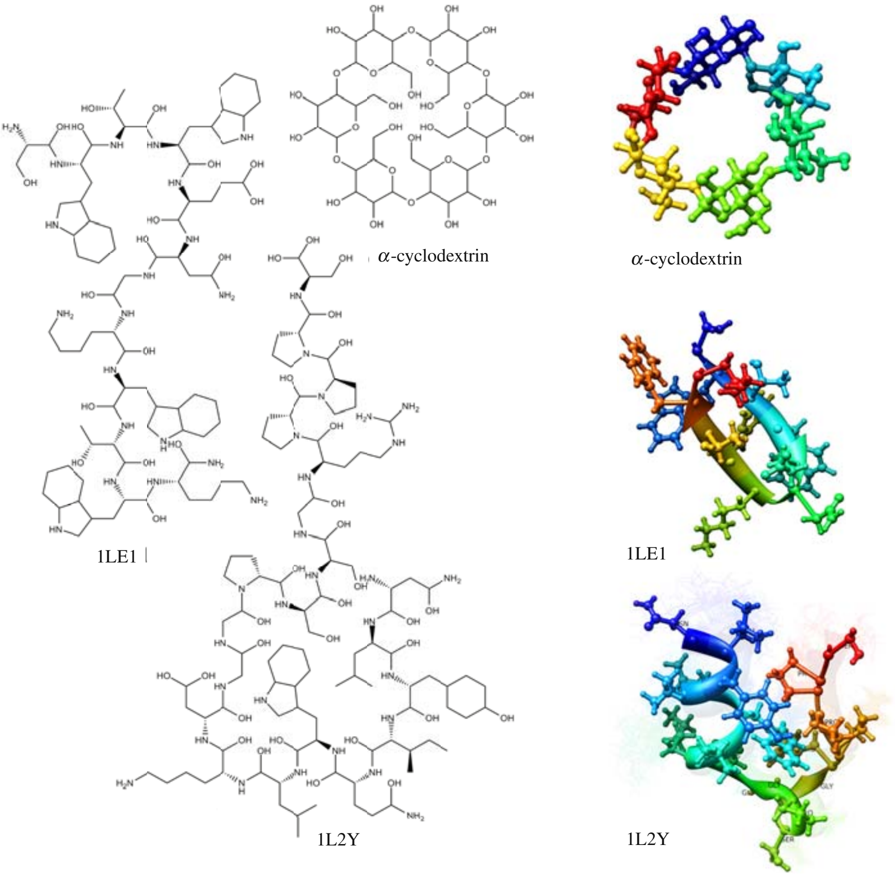


Fig. 7. Structural overview of the considered benchmarks – α-cyclodextrin, tryptophan-zipper (1LE1) and tryptophan-cage (1L2Y).

Although the induced fitness degradation, with a 0.05 probability, worse solutions are accepted in order to exploit the potentially significant enclosed information. Replacement is conducted on similar basis, with a 0.75 probability of discarding a worse solution. The refinement phase has been set to be applied at every five generations, relying exclusively on the ASA component, described in Section 3.3. A fine-grained gradient minimization is additionally carried out on the resulting conformations, exploiting the analytical foundations of the conjugate gradient local search operator. A worse-replacement strategy is used for reinserting the final refined solutions into the initial population.

Another element with important consequences over the convergence of the constructed algorithm is given by the asynchronous migration rate. Frequent migrations may result in a premature convergence while distant migrations fall at the opposite extreme - exploration conducted on distinct algorithms with independent evolution

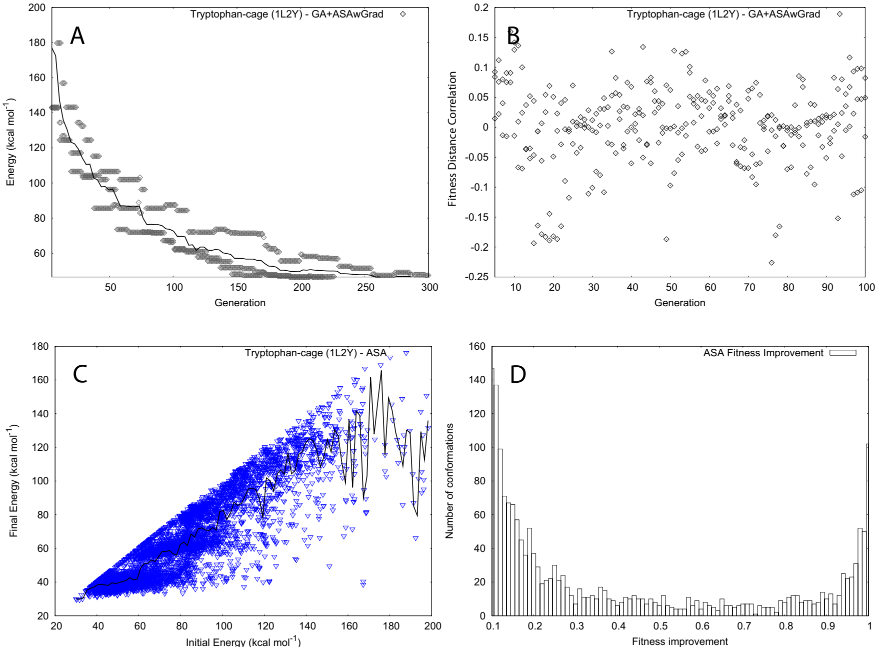


Fig. 8. 1L2Y – An execution example depicting the profile of the island model algorithm (A, B), in the second row, the evolution of the ASA component being captured (C, D).

curves. For the herein case, one sixth of the local population is set to emigrate, in asynchronous manner, at every five generations - migrations may occur at different times, depending on the advancement of each algorithm.

A meta-evolutionary genetic algorithm has been designed for finding an optimal parameterization of the adaptive simulated annealing algorithm. No special strategies or operators were designed, a simple distributed EA being considered; the algorithm has been executed inside the same grid environment. In this case each individual of the meta-algorithm represents an encoding of the different parameterization values – control parameters of the adaptive simulated annealing algorithm, initial temperature, number of accepted solutions determining reannealing, quenching factor, etc. The fitness of each individual has been computed as the average improvement obtained after running the adaptive SA on a set of five known difficult conformations. For each fitness evaluation run, for each of the five conformations, a maximum number of 3000 samplings was set. As an example, one of the chosen resulting parameterizations had a reannealing limit of 111 accepted conformations with 97 sampling points at each temperature and a large quenching factor of 33.16.

For the synchronous multi-start execution, two approaches were considered. The adaptive simulated annealing algorithm is either executed in order to sample 3000 solutions in one run, either 10 short runs with 300 samples each are iteratively launched. In addition, at the end of one ASA run, the outcome conformation is further optimized by applying a 30 step gradient.

As a first remark, after compiling the execution results, the use of conjugate gradient extended operators determined a dramatic improvement. Analyzing, for example, the results obtained by using the genetic algorithm alone, for the α-cyclodextrin conformation, an average of 3790.56 $kcal\ mol^{-1}$ (stdev. 708.54 $kcal\ mol^{-1}$) has been attained, with a maximum, minimum of 5845.27 $kcal\ mol^{-1}$, respectively 2470 $kcal\ mol^{-1}$. At the opposite extreme, the set of solutions found by the gradient hybridized genetic algorithm resulted in an average of 201.37 $kcal\ mol^{-1}$ (stdev. 21.82 $kcal\ mol^{-1}$), with a minimum of 161.69 $kcal\ mol^{-1}$ and a maximum of 243.05 $kcal\ mol^{-1}$. A number of 30 independent executions were performed for the gradient hybridized GA as well as for the GA alone, with no hybridization.

Finally, for all studied benchmarks, the ASA-hybridized GA (best scored conformations) attained a below native reference energy: 28.9 $kcal\ mol^{-1}$ for the 1L2Y protein (reference energy at 46.6 $kcal\ mol^{-1}$), -3.5 $kcal\ mol^{-1}$ for 1LE1 (11.1 $kcal\ mol^{-1}$) and 161.6 $kcal\ mol^{-1}$ for α-cyclodextrin (242.4 $kcal\ mol^{-1}$). Nevertheless, although descending below the energy of the native conformation, the corresponding RMSD (Root Mean Square Deviation) values were constantly outside acceptable limits, with minimum values close to or above 4Å.

A graphical illustration, capturing the island model algorithm evolution, is given in Fig. 8. The depicted examples outline, in a first step, results obtained for the hybrid island based algorithm, while the second part offers an overall perspective of the ASA execution-time improvement rate. For each island, at every generation, the fitness of the best found conformation is depicted (A), a median trend evolution line being traced. Although the algorithms advance at different rates, with several thresholds, convergence is attained near 300 generations. A corresponding fitness distance correlation (FDC) [32] plot is additionally illustrated (B), offering an overview of the fitness dynamics, *e.g.* convergence rate information, over generations fitness variance, etc. An ideal case would consist of a 1.0 FDC value, expressing a perfect correlation between fitness and inter-solutions distance values, while, at the opposite end, a -1.0 FDC value indicates a completely uncorrelated landscape, providing no useful information. A symmetrical spread may be observed (B), with an ascending positive correlation trend, as determined by the advancement of the exploration. Additionally, an outline of the ASA improvement bias is shown in the second row of the figure, traced as a plot exposing initial vs. final energy (C) and, second, as a histogram (D). Approximately one sixth of the refined conformations allowed for an above 10% improvement while only a reduced fraction of 3% resulted in an above 90% improvement (D). The equivalent run-time evolution graph (C), exclusively considering the ASA refinement outcomes, revealed several clusters, attributed to strong attractors determining basins in the conformational landscape (visible at ~ 40.0 $kcal\ mol^{-1}$, 60.0 $kcal\ mol^{-1}$, final energy - C).

As an overall conclusion, first, the hybrid parallel algorithm design incurs strong exploration capabilities, although, second, far from native outcome conformation were returned. Appearing as energy landscape artifacts, with high RMSD - low energy conformations, due to the force field parameterization, the obtained solutions do not stand as valid conformations. As a consequence it can be concluded that a higher level extensive exploration approach is required with a more robust evaluation protocol.

4.3 Advanced Hybrid Hierarchical Parallel Algorithm

As determined by the drawn conclusions, a cluster sampling, domain decomposition oriented algorithm has been considered. A straightforward extension of the representation model has been constructed by considering, for a chromosome, an overlapping associated domain. Defining symmetric boundaries, for a given conformation, $\gamma = (\gamma_1, \gamma_2, \cdots, \gamma_N)$, a landscape domain is delimited, further denoted as $< \gamma,\ \eta > \equiv ([\gamma_1 - \eta_1, \gamma_1 + \eta_1],\ [\gamma_2 - \eta_2, \gamma_2 + \eta_2],\ \cdots,\ [\gamma_N - \eta_N, \gamma_N + \eta_N])$. The introduced definition and representation synthetically maps, over the conformer concept, nevertheless encompassing a less conformational structure significance, *i.e.* no underlying *specific* base template is associated to the given domain. Therefore, the term of *cell* is preferred in the following, describing, by direct association, a bounded structural subspace, as opposed to conformer, in order to designate a $< \gamma,\ \eta >$ entity. For simplicity, the assumption of having $\eta_i = \delta,\ 1 \leq i \leq N$, is considered in the following, where δ represents an *a priori* fixed arbitrary positive value. Additionally, having as basis the formulated assumption, a direct notation $< \gamma > \equiv < \gamma,\ \eta >$, with $\eta_i = \delta,\ 1 \leq i \leq N$, is employed in the following, as to designate a cell. An intuitive graphical representation is given in Fig. 9, depicting the transition from a highly multimodal energy landscape to a smoother, conformer fitness space. From an implementation point of view, the representation is constructed as an extension of the previously defined model, permitting the reuse of the entire developed algorithmic architecture, with no or less modifications.

A direct evaluation would consist of considering a $< \gamma >$ cell as designating an ensemble of solutions. Consequently, the problem resides in defining an appropriate evaluation function which, for a specified δ value and for a given cell, $< \gamma >$, offers a coherent evaluation, quantifying the *stability* of an overlapping conformer. Nevertheless, no complete characterization of a particular cell is possible, unless accounting for the cumulated interaction and contribution of an infinite number of conformations, confined within the cell boundaries. Consequently, an extrapolation formalism has to be defined, the evaluation function being constrained to infer on a *finite*, *representative* subset of conformations. Furthermore, the evaluation function has to be *reproducible*. Otherwise stated, assuming that representative independently sampled subsets $\mathscr{S}_i \subseteq < \gamma >,\ i \in \mathbb{N}$ are given, *comparable* evaluation results have to be provided. The construction of a *representative* cell subset hence demands for a within cell sampling to be performed, algorithmic basis being provided by the already defined approach.

Holding for the aforementioned specifications and having as support intuitive underlying physical concepts, a Gibbs free energy evaluation is considered – refer to Section 3.1 and Horvath *et al.* [26] for additional references and details. The function relies on the individual evaluation of a set of sampled solutions, $\gamma^s \in \mathscr{S}$,

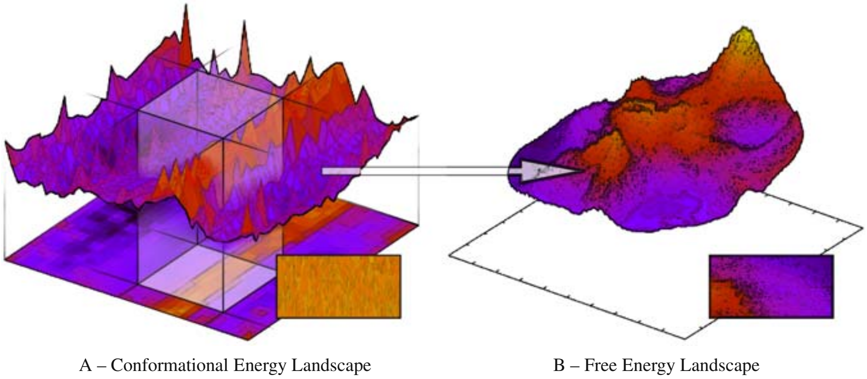

A – Conformational Energy Landscape
B – Free Energy Landscape

Fig. 9. Underlying conceptual basis of a free energy clustered sampling algorithm. A finite set of solutions, sampled within the boundaries of a specified cell (A), is employed for constructing a free energy evaluation – resulting in a singular free energy surface point (B). A more tractable landscape is obtained at the price of a higher computational load.

$\mathscr{S} \subseteq <\gamma>$. Extrapolating over the formalization details, an entropy equivalent measure is obtained, offering a characterization of the within $<\gamma>$ cell key minima depth and width. As determined by the nature of the evaluation function, a less sensitive to extreme perturbation energy values evaluation is attained, resulting in a smoothing effect. A graphical simplified corresponding exemplification, for the 1L2Y protein, is illustrated in Fig. 9.

As mentioned in the previous paragraphs, the construction of a *representative* set has to be addressed, as part of the fitness function definition. A first design decision consists in determining an optimal δ value. High values result in a reversion towards the initially addressed problem while, at the opposite end, reduced values imply the exploration space to be segmented into a large number of cells. The former case, while offering the advantage of simplifying the search space clustering, requires the support of a thorough intensive sampling, posing a reproducibility problem and, hence, inducing a high computational load. In analogous manner, the latter case, while assuring for *representative* sampled sets, results in an expensive exploration process, due to an explosion of the number of cells to be explored. With no or less information acquired, at the extreme case, the initial conformational energy landscape is potentially reproduced. Consequently, a sampling algorithm dependent balance has to be assured in order to allow for a pertinent segmentation of the search space and as to exploit the information which can be derived by assessing an ensemble of conformations. Therefore, a second correlated design decision, concerns the exploration algorithm to be employed – a random sampling would stand as a simple and fast candidate solution although offering no *reproducibility* guarantee, unless reduced size cells are considered. An exploration intensive approach, allowing for the search to be conducted over extended landscape domains, although enforcing the imposed demands, can potentially result in a redundant oversampling.

Assembling the introduced representation model and the free energy evaluation function, a meta-evolutionary algorithm has been constructed, the exploration being conducted over clusters in the conformational energy space. A hierarchical design is offered, comporting multiple parallelization levels. As highly complex aspects are addressed, no effective approach can be defined unless extensive distributed computational resources are employed. A first parallelization layer is inserted at the global meta-exploration level, the evaluation of each solution being synchronously delegated to local samplers. Further, each of the sampling processes deploys several island algorithms, for each island, a parallel evaluation of the conformations being performed at each generation, with additional synchronous multi-start refinement and migration processes. A schematic representation is given in Fig. 10. Note that, following the parallelization hierarchy, a highly scalable approach is attained, as determined by the decomposition of the parallel tasks. Given that, the implied design decisions mainly depend on the selected local sampling algorithm, the defined architecture is presented starting with the lower exploration layer as to end with the meta-exploration algorithm level.

As main criterion in proposing a local sampler solution, the requirement for an exploration intensive algorithm has been considered, as to allow for free energy evaluations on large cells within acceptable reproducibility limits. As demonstrated by the previous results, the algorithmic model proposed in Section 3.4, stands as a powerful candidate solution. Consequently, the same exact architecture has been used, with several modifications as detailed in the following. In depth details and analysis test cases, standing as basis for the herein obtained results, were also presented in [54, 55], addressing multiple operators, local search algorithms, adaptive and dynamic mechanisms, etc. Nonetheless, as we are here interested in exposing the hierarchical nature of the algorithms, opposing local and global sampling paradigms, no further details are here included.

Conducting several trial experimentations, it has been determined as coherent and sufficient to set a value of $\delta = 45$, corresponding to a $\pi/4$ angular value and allowing for wide extended cells to be defined. Further, a discrete representation has been adopted, where, for a $< \gamma >$ cell corresponding genotype, the enclosed $< \gamma_i >,\ 1 \leq i \leq N$, *loci* has been defined as having values from the $\{0, \cdots, 7\}$ set, with a corresponding angle value in the $[\delta(< \gamma_i > -1),\ \delta(< \gamma_i > +1)]$ interval. An inter-cells overlap has been allowed as to avoid boundary constraints, *e.g* torsional angle values requiring fine tuning near boundary limits. Note that the representation employed by the local sampling algorithm has not been modified, a mapping being defined as to assure the coherence of the representation.

Having as a pragmatic constraint the requirement of allowing for a fast sampling process to be conducted, a reduced population size of only 30 solutions has been assigned, for each island of the sampling algorithm, to be evolved over 10 generations. The exact same configuration of the operators and inter-algorithm migration topologies has been maintained, as presented in Section 3.4, with a down-scaling of the afferent parameter values. Local search refinement has been set to be triggered at every 5 generations, additionally, migrations being performed at every 2 generations, with an exchange of 10 individuals. Furthermore, a maximum of 10

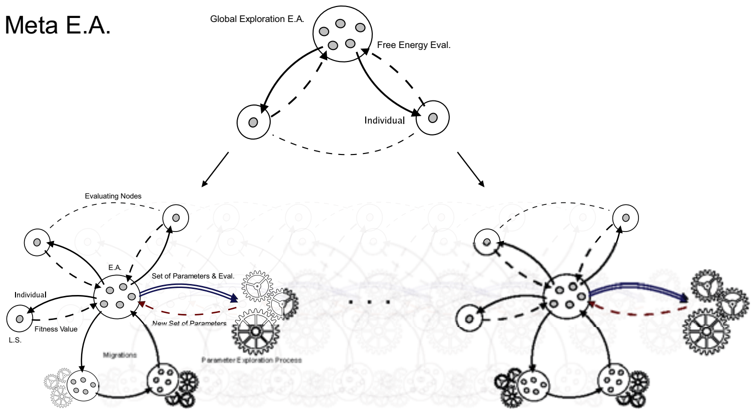
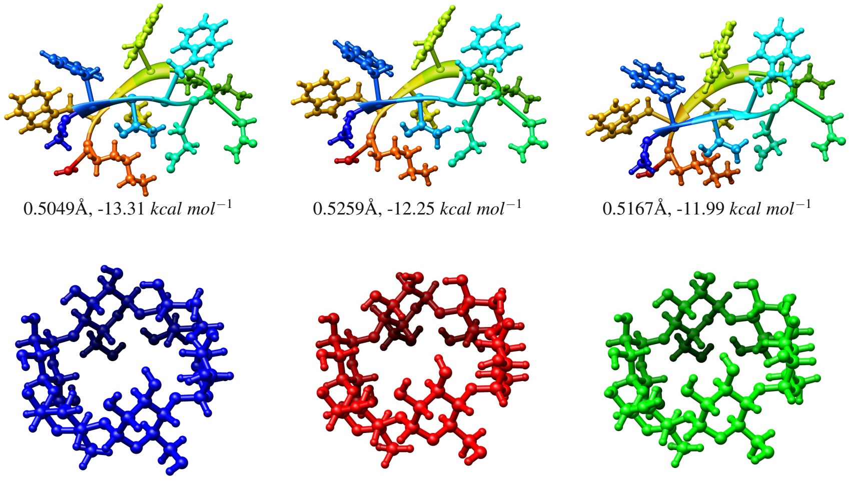


Fig. 10. A conceptual model depicting the architecture of the meta-exploration algorithm. Global exploration is carried at a conformer level, for each conformer associated cell, local sampling based free energy evaluations being computed.

conjugate gradient steps has been set, as opposed to the initial default configuration of 30 steps. As the exploration is carried out inside specified cell boundaries, all the determined solutions, as provided by each island, contribute to the construction of a representative sampling set. Therefore, at the end of the sampling process, a screening is performed, a set of the best found 30 distinct conformations being assembled. The gathered set further stands as basis for computing the free energy evaluation, characterizing the initial subjected cell.

Discrete combined operators have been employed, as to maintain a coherence of the representation, without introducing repairing mechanisms. Mutation has been defined as to be carried on a swap, random flip and a complete shuffle operators, with equal rates and with a 0.3 overall probability. In analogous manner, a uniform and a two-points crossover operators, with equal rates and with a 0.95 overall probability have been specified. A fitness sharing selection strategy is included, the distance, for two specified cells, $<\gamma^a>$, $<\gamma^b>$, being defined as the percentage of positional different *loci*. In this context, two solutions are considered to be part of the same cluster if found at a distance below 0.25, *i.e.* less than a quarter of the *loci* having different values. Additionally, the replacement is carried on a stochastic tournament strategy, with weak elitism enabled and with a 0.95 probability of discarding a worse solution over a better one.

At execution time, a maximum walltime of 50 hours has been imposed, the algorithm being executed in successive runs over a variable number of computational resources, with an average of ~400 cores. The algorithm has been set to evolve a population of 30 solutions for 100 generations, each solution defining a cell to be sampled. As resulting from the obtained outcomes, the proposed approach offered impressive results – refer to Fig. 11 for a graphical illustration. As an example, for the 1LE1 protein, the algorithm ranked first the cell centered around the native reference, within the cell, the first ranked solution, with a -13.32 $kcal\ mol^{-1}$



0.5049Å, -13.31 *kcal mol*$^{-1}$ 0.5259Å, -12.25 *kcal mol*$^{-1}$ 0.5167Å, -11.99 *kcal mol*$^{-1}$

1.3689Å, 160.49 *kcal mol*$^{-1}$ 1.0582Å, 160.49 *kcal mol*$^{-1}$ 0.7264Å, 160.69 *kcal mol*$^{-1}$

Fig. 11. Tryptophan-zipper (first row) and α-cyclodextrin (second row) – best found conformations, ranked in concordance with the associated energy.

conformational energy fitness, standing as a perfect match, with a 0.5049Å RMSD. Additionally, an average RMSD of 0.6431Å has been attained for the 30 first ranked conformations, with a minimum, maximum RMSD of 0.3611Å (-9.23 *kcal mol*$^{-1}$), respectively 2.0860Å (-7.14 *kcal mol*$^{-1}$). In similar manner, for the α-cyclodextrin molecule, for the top 30 ranked conformations, a 3.7595Å average has been attained, with a minimum, maximum value of 0.5313Å (162.01 *kcal mol*$^{-1}$), respectively 8.9869Å (675.54 *kcal mol*$^{-1}$) – remarkable to notice, only 4 out of the 15 first ranked conformations had an RMSD above 1.0Å. As exposed in Fig. 11, for the first three α-cyclodextrin conformations, an RMSD of 1.3689Å (160.49 *kcal mol*$^{-1}$), 1.0582Å (160.49 *kcal mol*$^{-1}$), respectively 0.7264Å (160.69 *kcal mol*$^{-1}$) has been obtained. Although no similar results have been attained, in the given time frame, for the *tryptophan-cage* protein, undergoing independent studies, carried out in the context of the Docking@Grid project, confirmed an over-fitting bias of the employed force field, resulting in non-consistent results when addressing α-helices vs. β-sheets patterns.

5 Conclusions and Future Work

Allowing for extreme hybrid constructions to be defined and enclosing intrinsic parallel support, evolutionary algorithms comport, nevertheless, a high structural complexity level. Different evolutionary parallel models were employed, initial experimentations standing as a proof for the intensive exploration capabilities of the approach. An extension of the initial approach was defined, addressing conformers instead of singular conformations. A free energy evaluation function was introduced in the model, endorsing the evaluation of clusters of conformations as an ensemble and quantifying the width and the depth of the representative conformer minima region. Impressive results were attained for the *tryptophan-zipper* protein and for the α-cyclodextrin conformational benchmark, with a below 1.0Å RMSD average for the first 30 ranked 1LE1 conformations. All experimentations were conducted on Grid'5000 [11], different MPI distributions [23, 43, 22] being employed at execution time. To conclude with, an effective high-performance parallel hybrid conformational sampling algorithm was constructed, answering the initially defined *ANR Dock Project – Conformational Sampling and Docking on Computational Grids* directions.

An unlimited number of consequent prospective directions may be considered, enforcing the obtained outcomes, the exploration of novel parallel paradigms, etc. A consequential study entailing exploration approach enhancements stands as an adjacent objective in order to encompass high-throughput conformational screening support. Finally, extensive background for arising technologies, *e.g.* General Purpose GPUs (Graphics Processing Units) [5], MPICH-G4/MPIg, etc., is considered as to expand the ParadisEO framework, including fault-tolerance, desktop computing and volatile environments support.

References

1. Alba, E., Talbi, E.G., Luque, G., Melab, N.: Metaheuristics and parallelism. In: Parallel Metaheuristics: A New Class of Algorithms. Wiley Series on Parallel and Distributed Computing, vol. 4, pp. 79–104. Wiley, Chichester (2005)
2. Alder, B., Wainwright, T.: Phase transition for a hard sphere system. Journal of Chemical Physics 27, 1208–1209 (1957)
3. Alder, B., Wainwright, T.: Studies in molecular dynamics. i. general method. Journal of Chemical Physics 31, 459–466 (1959)
4. Bernstein, F., Koetzle, T., Williams, G., Meyer Jr., E.F., Brice, M., Rodgers, J., Kennard, O., Shimanouchi, T., Tasumi, M.: The protein data bank: a computer-based archival file for macromolecular structures. Journal of Molecular Biology 112, 535–542 (1977)
5. Buck, I.: Gpu computing: Programming a massively parallel processor. In: Proceedings of the International Symposium on Code Generation and Optimization (CGO 2007), p. 17. IEEE Computer Society, Washington (2007)
6. Butenhof, D.: Programming with POSIX Threads. Professional Computing Series. Addison-Wesley Longman Publishing Co., Boston (1997)
7. Cahon, S., Melab, N., Talbi, E.G.: Paradiseo: A framework for the reusable design of parallel and distributed metaheuristics. Journal of Heuristics 10(3), 357–380 (2004)
8. Cahon, S., Melab, N., Talbi, E.G.: An enabling framework for parallel optimization on the computational grid. In: Proceedings of International Symposium on Cluster Computing and the Grid (CCGrid 2005), vol. 2, pp. 702–709. IEEE Computer Society, Washington (2005)
9. Calland, P.Y.: On the structural complexity of a protein. Protein Engineering 16(2), 79–86 (2003), `http://peds.oxfordjournals.org/cgi/content/abstract/16/2/79`

10. Cantu-Paz, E.: Efficient and Accurate Parallel Genetic Algorithms. Kluwer Academic Publishers, Norwell (2000)
11. Cappello, F., Caron, E., Dayde, M., Desprez, F., Jegou, Y., Primet, P., Jeannot, E., Lanteri, S., Leduc, J., Melab, N., Mornet, G., Namyst, R., Quetier, B., Richard, O.: Grid'5000: A Large Scale and Highly Reconfigurable Grid Experimental Testbed. In: Proceedings of IEEE/ACM International Workshop on Grid Computing (GRID 2005), pp. 99–106. IEEE Computer Society, Washington (2005), http://dx.doi.org/10.1109/GRID.2005.1542730
12. Cozzone, A.: Proteins: Fundamental chemical properties. Encyclopedia of Life Sciences, 1–10 (2002), http://doi.wiley.com/10.1038/npg.els.0001330
13. Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A., Yannakakis, M.: On the complexity of protein folding. Journal of computational biology 5(3), 423–465 (1998)
14. Dauber-Osguthorpe, P., Roberts, V., Osguthorpe, D., Wolff, J., Genest, M., Hagler, A.: Structure and energetics of ligand binding to proteins: Escherichia coli dihydrofolate reductase-trimethoprim, a drug-receptor system. Proteins: Structure, Function, and Genetics 4(1), 31–47 (1988), http://dx.doi.org/10.1002/prot.340040106
15. Dill, K.: Theory for the folding and stability of globular proteins. Biochemistry 24(6), 1501–1509 (1985), http://view.ncbi.nlm.nih.gov/pubmed/3986190
16. Dorsett, H., White, A.: Overview of molecular modelling and ab initio molecular orbital methods suitable for use with energetic materials. Tech. Rep. DSTO-GD-0253, Department of Defense, Weapons Systems Division, Aeronautical and Maritime Research Laboratory, Salisbury, South Australia (2000)
17. Fletcher, R., Powell, M.: A rapidly convergent descent method for minimization. Computer Journal 6, 163–168 (1963)
18. Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. Computer Journal 7, 149–154 (1964)
19. Foster, I., Kesselman, C.: The Grid: Blueprint for a new computing infrastructure. Morgan Kaufmann Publishers, Los Altos (2003)
20. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of High Performance Computing Applications 15(3), 200–222 (2001), http://dx.doi.org/10.1177/109434200101500302
21. Garwin, L., Lincoln, T.: A Century of Nature: Twenty-One Discoveries that Changed Science and the World. University of Chicago Press, Chicago (2003)
22. Graham, R., Shipman, G., Barrett, B., Castain, R., Bosilca, G., Lumsdaine, A.: Open mpi: A high-performance, heterogeneous mpi. In: Proceedings of CLUSTER. IEEE, Los Alamitos (2006), http://dblp.uni-trier.de/db/conf/cluster/cluster2006.html#GrahamSBCBL06
23. Gropp, W.: Mpich2: A new start for mpi implementations. In: Kranzlmüller, D., Kacsuk, P., Dongarra, J., Volkert, J. (eds.) PVM/MPI 2002. LNCS, vol. 2474, p. 7. Springer, Heidelberg (2002)
24. Gropp, W., Lederman, S., Lumsdaine, A., Lusk, E., Nitzberg, B., Saphir, W., Snir, M.: MPI: The Complete Reference, vol. 2. MIT Press, Cambridge (1998)
25. Hestenes, M., Stiefel, E.: Methods of conjugate gradients for solving linear systems. Journal of Research of the National Bureau of Standards 49(6), 409–436 (1952)
26. Horvath, D., Brillet, L., Roy, S., Conilleau, S., Tantar, A.A., Boisson, J.C., Melab, N., Talbi, E.G.: Local vs. global search strategies in evolutionary grid-based conformational sampling & docking. In: Proceedings of IEEE Congres on Evolutionary Computation, CEC 2009 (2009)

27. Ingber, L.: Adaptive simulated annealing (asa), global optimization c-code. Tech. rep., Caltech Alumni Association (1993)
28. Ingber, L.: Simulated annealing: Practice versus theory. Journal of Mathematical Computation Modelling 18(11), 29–57 (1993)
29. Ingber, L.: Adaptive simulated annealing (asa): Lessons learned. Control and Cybernetics 25, 33–54 (1996)
30. Ingber, L.: Adaptive simulated annealing (asa) and path-integral (pathint) algorithms: Generic tools for complex systems. Tech. rep., Chicago, IL (2001)
31. Ingber, L., Rosen, B.: Genetic algorithms and very fast simulated reannealing: A comparison. Mathematical Computer Modeling 16(11), 87–100 (1992)
32. Jones, T., Forrest, S.: Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Proceedings of International Conference on Genetic Algorithms, pp. 184–192. Morgan Kaufmann, San Francisco (1995)
33. Keijzer, M., Guervós, J., Romero, G., Schoenauer, M.: Evolving objects: A general purpose evolutionary computation library. In: Proceedings of European Conference on Artificial Evolution (EA 2002), pp. 231–244. Springer, London (2002)
34. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983), `citeseer.ist.psu.edu/kirkpatrick83optimization.html`
35. Krauter, K., Buyya, R., Maheswaran, M.: A taxonomy and survey of grid resource management systems for distributed computing. Software Practice and Experience 32(2), 135–164 (2002), `citeseer.ist.psu.edu/krauter01taxonomy.html`
36. Lander, E.S., Linton, L.M., Birren, B., et al.: Initial sequencing and analysis of the human genome. Nature 409(6822), 860–921 (2001)
37. Little, P.: Dna sequencing: the silent revolution. In: A Century of Nature: Twenty-One Discoveries that Changed Science and the World, ch. 16. University of Chicago Press, Chicago (2003)
38. McCammon, J.A., Gelin, B.R., Karplus, M.: Dynamics of folded proteins. Nature 267(5612), 585–590 (1977)
39. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equation of state calculations by fast computing machines. The Journal of Chemical Physics 21(6), 1087–1092 (1953), `http://link.aip.org/link/?JCP/21/1087/1`
40. Neumaier, A.: Molecular modeling of proteins and mathematical prediction of protein structure. SIAM Review 39(3), 407–460 (1997), `citeseer.ist.psu.edu/neumaier97molecular.html`
41. Ngo, J.T., Marks, J.: Computational complexity of a problem in molecular structure prediction. Protein Engineering 5(4), 313–321 (1992), `http://peds.oxfordjournals.org/cgi/content/abstract/5/4/313`
42. Pande, V., Baker, I., Chapman, J., Elmer, S., Khaliq, S., Larson, S., Rhee, Y., Shirts, M., Snow, C., Sorin, E., Zagrovic, B.: Atomistic protein folding simulations on the submillisecond timescale using worldwide distributed computing. Biopolymers 68(1), 91–109 (2003)
43. Pant, A., Jafri, H.: Communicating efficiently on cluster based grids with mpich-vmi. In: Proceedings of CLUSTER, pp. 23–33. IEEE Computer Society, Los Alamitos (2004), `http://dblp.uni-trier.de/db/conf/cluster/cluster2004.html#PantJ04`
44. Polak, E., Ribière, G.: Note sur la convergence des méthodes de directions conjuguées. Revue française d'informatique et de recherche opérationnelle 16, 35–43 (1969)
45. Ponder, J., Case, D.: Force fields for protein simulations. Advances in Protein Chemistry 66, 27–85 (2003)

46. Rabow, A., Scheraga, H.: Improved genetic algorithm for the protein folding problem by use of a Cartesian combination operator. Protein Science 5(9), 1800–1815 (1996), http://www.proteinscience.org/cgi/content/abstract/5/9/1800
47. Schulten, K., Phillips, J., Kale, L., Bhatele, A.: Biomolecular modeling in the era of petascale computing. In: Bader, D. (ed.) Petascale Computing: Algorithms and Applications, pp. 165–181. Chapman & Hall/CRC Press, Boca Raton (2008)
48. Shewchuk, J.: An introduction to the conjugate gradient method without the agonizing pain. Tech. rep., Carnegie Mellon University, Pittsburgh, PA, USA (1994), http://portal.acm.org/citation.cfm?id=865018
49. Shirts, M., Pande, V.: COMPUTING: Screen Savers of the World Unite! Science 290(5498), 1903–1904 (2000)
50. Stewart, C., Müller, M., Lingwall, M.: Progress towards petascale applications in biology: Status in 2006. In: Proceedings of Euro-Par Workshops, pp. 289–303 (2006)
51. Talbi, E.G.: A taxonomy of hybrid metaheuristics. Journal of Heuristics 8(5), 541–564 (2002)
52. Tantar, A.A., Melab, N., Demarey, C., Talbi, E.G.: Building a virtual globus grid in a reconfigurable environment - a case study: Grid5000. Tech. Rep. inria-00168130, INRIA, France (2007), http://hal.inria.fr/inria-00168130/en
53. Tantar, A.A., Melab, N., Talbi, E.G.: A grid-based genetic algorithm combined with an adaptive simulated annealing for protein structure prediction. Soft Computing 12(12), 1185–1198 (2008)
54. Tantar, A.-A., Melab, N., Talbi, E.-G.: An analysis of dynamic mutation operators for conformational sampling. In: Biologically-inspired Optimisation Methods: Parallel Algorithms, Systems and Applications. Studies in Computational Intelligence. Springer, Heidelberg (2009)
55. Tantar, E., Tantar, A.A., Melab, N., Talbi, E.G.: Analysis of local search algorithms for conformational sampling. In: Advances in Parallel Computing, Parallel Programming and Applications on Grids, P2P and Networked-based Systems. IOS Press, Amsterdam (2009)
56. Venter, J., Venter, J., Adams, M., et al.: The sequence of the human genome. Science 291(5507), 1304–1351 (2001), http://www.sciencemag.org/cgi/content/abstract/291/5507/1304
57. Van de Waterbeemd, H., Carter, R., Grassy, G., Kubinyi, H., Martin, Y., Tute, M., Willett, P.: Glossary of terms used in computational drug design. Pure and Applied Chemistry 69(5), 1137–1152 (1997)
58. White, A., Zerilli, F., Jones, H.: Ab initio calculation of intermolecular potential parameters for gaseous decomposition products of energetic materials. Tech. Rep. DSTO-TR-1016, Department of Defense, Energetic Materials Research and Technology Department, Naval Surface Warfare Center, DSTO-TR-1016, Melbourne Victoria 3001, Australia (2000)