

Constraint-Handling Through Multi-Objective Optimization: the Hydrophobic-Polar Model for Protein Structure Prediction

Mario Garza-Fabre*, Eduardo Rodriguez-Tello, Gregorio Toscano-Pulido

Information Technology Laboratory, CINVESTAV-Tamaulipas
Parque Científico y Tecnológico TECNOTAM
Km. 5.5 carretera Cd. Victoria-Soto La Marina
Cd. Victoria, Tamaulipas 87130, MÉXICO
+52 (834) 107 0220

Abstract

In the multi-objective approach to constraint-handling, a constrained problem is transformed into an unconstrained one by defining additional optimization criteria to account for the problem constraints. In this paper, this approach is explored in the context of the hydrophobic-polar model, a simplified yet challenging representation of the protein structure prediction problem. Although focused on such a particular case of study, this research work is intended to contribute to the general understanding of the multi-objective constraint-handling strategy. First, a detailed analysis was conducted to investigate the extent to which this strategy impacts on the characteristics of the fitness landscape. As a result, it was found that an important fraction of the infeasibility translates into neutrality. This neutrality defines potentially shorter paths to move through the landscape, which can also be exploited to escape from local optima. By studying different mechanisms, the second part of this work highlights the relevance of introducing a proper search bias when handling constraints by multi-objective optimization. Finally, the suitability of the multi-objective approach was further evaluated in terms of its ability to effectively guide the search process. This strategy significantly improved the performance of the considered search algorithms when compared with respect to commonly adopted techniques from the literature.

Keywords: Constraint-handling, evolutionary multi-objective optimization, fitness landscape analysis, search bias, protein structure prediction, hydrophobic-polar model

1. Introduction

Evolutionary computation methods and other metaheuristic algorithms have been successfully used to solve complex optimization problems which arise in a diversity of scientific and engineering applications. Often, however, optimization involves not only to reach the best value for a given objective function (or set of objective functions), but also to satisfy a certain set of predefined requirements called constraints. Therefore, additional mechanisms need to be implemented within metaheuristic algorithms in order to search effectively through this kind of constrained solution spaces.

The *hydrophobic-polar* (HP) model [1, 2] is an abstract formulation of the *protein structure prediction* (PSP) problem, where hydrophobicity is assumed to be the main stabilizing force in the protein folding process. Under this model, PSP is defined as the problem of finding a *self-avoiding embedding* of the protein chain on a given lattice, such that the interaction among hydrophobic amino acids is maximized. From the computational point of view, the HP model entails a challenging

problem in combinatorial optimization [3, 4]. One of the main sources of difficulty in this problem lies in the fact that, using the existing problem representations, a significant portion of the solution space encodes infeasible (non-self-avoiding) protein structures. Hence, it is important to devise effective mechanisms for handling the constraints that this problem presents. Two main research directions have been adopted to cope with this issue. On the one hand, the search can be confined to the space of only feasible, self-avoiding protein conformations. On the other hand, infeasible protein conformations can also be taken into consideration, which has been achieved in the literature by implementing a *penalty strategy*. From the literature, however, it is not possible to identify a clear consensus on which of the two directions, *i.e.*, to avoid or to consider infeasible conformations, could lead to the development of more efficient metaheuristics for solving this problem [5–9].

Premised upon the belief that infeasible conformations can provide valuable information for guiding the search process, this research work inquires into the use of multi-objective optimization as an alternative constraint-handling strategy for the HP model. Particularly, constraints in the HP model are treated as a supplementary optimization criterion, leading to an unconstrained multi-objective problem.¹ Using such an alternative

*Corresponding author

Email addresses: mgarza@tamamps.cinvestav.mx (Mario Garza-Fabre),
ertello@tamamps.cinvestav.mx (Eduardo Rodriguez-Tello),
gtoscano@tamamps.cinvestav.mx (Gregorio Toscano-Pulido)

¹The process of restating a single-objective problem as a multi-objective

formulation of the HP model, infeasible solutions can become incomparable with respect to feasible ones, having thus better opportunities for participating throughout the search process. In contrast to the penalty strategy, which represents one of the most widely used techniques in the constraint-handling literature, in essence the multi-objective (MO) method does not require the fine-tuning of the penalty parameters;² in the penalty strategy, finding the right balance between objective function and penalty values has been regarded to be a difficult optimization problem itself [10, 11]. The use of multi-objective optimization for handling constraints is not a novel idea; recent reviews on this topic can be found in [11, 12]. Nevertheless, it was not until recently that the preliminary results of this research reported for the first time, to the best of the authors' knowledge, the application of the MO constraint-handling strategy to the particular HP model of the PSP problem [13].

Building further on this research, the primary aim of this study is to contribute to the general understanding of the functioning of the MO constraint-handling technique. First, a detailed analysis is conducted in order to investigate the potential effects of the problem transformation from the perspective of the fitness landscape. More specifically, it is evaluated how the use of the MO problem formulation impacts on an important property of the fitness landscape: *neutrality*. It has been argued that the MO approach to constraint-handling could be rather ineffective if a search bias towards the feasible region is not introduced [14]. Therefore, the second part of this document concerns the study of different mechanisms which can be employed for providing the MO strategy with such a search bias. The last part of this research work extends the comparative analysis reported in [13], where the MO approach is evaluated with respect to commonly adopted techniques from the specialized literature. While the preliminary results presented in [13] assumed a fixed biasing scheme for the MO method and focused only on the performance of a population-based algorithm, the different biasing mechanisms analyzed in the second part of this study, as well as both single-solution-based and population-based algorithms, have been included in the present study. Likewise, only 15 test instances for the two-dimensional HP model (based on the square lattice) were used in [13]. In contrast, the present study covers also the three-dimensional case (based on the cubic lattice) and a total of 30 test cases have been considered.

The remainder of this document is organized as follows. Section 2 provides background concepts and sets the notation used in this study. Section 3 reviews related work on constraint-handling methods for the HP model as well as on the topic of single-objective to multi-objective transformations. The studied MO constraint-handling approach is described in Section 4. Section 5 presents the analysis with regard to the fitness landscape transformation. The search bias issue is addressed in Section 6. The comparative study which focuses on search performance is covered in Section 7. Finally, Section 8 discusses the main findings and presents the conclusions of this

study. Appendices at the end of this document contain supplementary information with regard to implementation details of the considered search algorithms, performance measures, test instances, the methodology followed for the statistical significance analyses, and the utilized experimental platform.

2. Background concepts and notation

2.1. Single-objective and multi-objective optimization

Without loss of generality, a *single-objective optimization problem* can be formally stated as follows:

$$\begin{aligned} &\text{Minimize} && f(\mathbf{x}), \\ &\text{subject to} && \mathbf{x} \in \mathcal{X}_{\mathcal{F}}, \end{aligned} \quad (1)$$

where \mathbf{x} is a *solution vector*; $\mathcal{X}_{\mathcal{F}}$ denotes the *feasible set*, i.e., the set of all *feasible solution vectors* in the search space \mathcal{X} , $\mathcal{X}_{\mathcal{F}} \subseteq \mathcal{X}$; and $f : \mathcal{X} \rightarrow \mathbb{R}$ is the objective function to be optimized. The aim is thus to find the feasible solution(s) yielding the optimum value for the objective function; that is, to find $\mathbf{x}^* \in \mathcal{X}_{\mathcal{F}}$ such that $f(\mathbf{x}^*) = \min\{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}_{\mathcal{F}}\}$.

Similarly, a *multi-objective optimization problem* can be formally defined as follows:

$$\begin{aligned} &\text{Minimize} && \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})]^T, \\ &\text{subject to} && \mathbf{x} \in \mathcal{X}_{\mathcal{F}}, \end{aligned} \quad (2)$$

where $\mathbf{f}(\mathbf{x})$ is the *objective vector* and $f_i : \mathcal{X} \rightarrow \mathbb{R}$ is the i -th objective function, $i \in \{1, 2, \dots, k\}$. Rather than searching for a single optimal solution, the task in multi-objective optimization is to identify a set of trade-offs among the conflicting objectives. More formally, the goal is to find a set of *Pareto-optimal solutions* \mathcal{P}^* , such that $\mathcal{P}^* = \{\mathbf{x}^* \in \mathcal{X}_{\mathcal{F}} \mid \nexists \mathbf{x} \in \mathcal{X}_{\mathcal{F}} : \mathbf{x} < \mathbf{x}^*\}$. The symbol “ $<$ ” denotes the *Pareto-dominance* relation [15]:

$$\begin{aligned} \mathbf{x} < \mathbf{x}' &\Leftrightarrow \forall i \in \{1, \dots, k\} : f_i(\mathbf{x}) \leq f_i(\mathbf{x}') \quad \wedge \\ &\exists j \in \{1, \dots, k\} : f_j(\mathbf{x}) < f_j(\mathbf{x}'). \end{aligned} \quad (3)$$

If $\mathbf{x} < \mathbf{x}'$, then \mathbf{x} is said to *dominate* \mathbf{x}' . Otherwise, \mathbf{x}' is said to be *nondominated* with respect to \mathbf{x} , denoted by $\mathbf{x} \not< \mathbf{x}'$. The image of \mathcal{P}^* in the objective space is the so-called *Pareto-optimal front*, usually also referred to as the *trade-off surface*.

2.2. Fitness landscapes and neutrality

The notion of a fitness landscape, first introduced by Wright [16], has been found to be useful in understanding the most essential characteristics of certain optimization problems, or problem classes. By analyzing the fitness landscape, it is possible to gain further insight into problem difficulty as a means of explaining, or even predicting, the performance of search algorithms. Fitness landscape analysis is expected to provide important clues for guiding the development of more competitive search mechanisms, which are able to deal with (or to take advantage of) the particular characteristics of the given optimization task. Some fundamental definitions on this topic, which are relevant according to the scope of this study, are presented below. For a more comprehensive literature review on fitness landscapes analysis the reader can be referred to [17–21].

one is usually referred to as *multi-objectivization*; refer to Section 3.2.

²However, the MO strategy may require additional parameters or the combination with other mechanisms for biasing purposes.

A *fitness landscape* can be generally defined in terms of a triplet $(\mathcal{X}, \mathcal{N}, \xi)$. The first element, \mathcal{X} , represents the set of all potential solutions to the problem, *i.e.*, the *search space*. The notion of connectedness among solutions in \mathcal{X} is introduced by the so-called *neighborhood structure*, $\mathcal{N} : \mathcal{X} \rightarrow 2^{\mathcal{X}}$, a function which maps each possible solution $\mathbf{x} \in \mathcal{X}$ to a set of solutions $\mathcal{N}(\mathbf{x}) \subseteq \mathcal{X}$. Hence, $\mathcal{N}(\mathbf{x})$ is referred to as the neighborhood of \mathbf{x} and each $\mathbf{x}' \in \mathcal{N}(\mathbf{x})$ is called a *neighbor* of \mathbf{x} . Finally, ξ denotes the evaluation scheme, consisting of (i) a measure (or set of measures) to serve as an indicator of the quality of the different solution candidates; and (ii) a mechanism to impose an ordering relation given the adopted quality measure(s). As the evaluation scheme, in single-objective optimization a *fitness function* (usually directly related to the objective function of the problem) is considered and a simple ordering sets the preference relation among solutions.³ In the multi-objective context, however, a number of (conflicting) criteria determine the quality of solutions, so that defining an ordering relation is not as straightforward as in the single-objective case. The partial order induced by the Pareto-dominance relation is assumed in this study.

The fitness landscape of a problem can be studied in terms of different properties, being the *neutrality* property of particular importance given the purposes of the present study. The standard definition of neutrality, in the single-objective case, refers to the degree to which a landscape contains connected areas of equal fitness [20]. Considering a broader notion to cover also the multi-objective case, neutrality can be understood as the result of the *incomparability* that the adopted evaluation scheme ξ induces. The term incomparability is used in this study to indicate the situation where no preferences can be imposed between a pair of solutions, so that these solutions are considered equivalent when evaluated under ξ . Two different solutions $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ are said to be *neutral* (incomparable), denoted by $\text{neutral}(\mathbf{x}_1, \mathbf{x}_2)$, if either they share the same fitness value (single-objective case), or they are nondominated, in the Pareto sense, with respect to each other (multi-objective case).

Having defined neutrality, a series of related basic concepts can be introduced as follows. The *neutral neighborhood* of a solution $\mathbf{x} \in \mathcal{X}$ is given by the subset of all its *neutral neighbors*: $\mathcal{N}_n(\mathbf{x}) = \{\mathbf{x}' \in \mathcal{N}(\mathbf{x}) \mid \text{neutral}(\mathbf{x}, \mathbf{x}')\}$. The total number of neutral neighbors of \mathbf{x} , *i.e.*, the cardinality of $\mathcal{N}_n(\mathbf{x})$, is known as the *neutrality degree* of \mathbf{x} , and the ratio of the neutrality degree to the size of the neighborhood is referred to as the *neutrality ratio*. A *neutral fitness landscape* is characterized by a large number of solutions presenting a high degree of neutrality. This leads to (potentially large) connected areas of incomparable solutions called *plateaus*, more formally referred to as *neutral networks*. Consider the *neutrality graph* $G = (\mathcal{X}, \mathcal{E}_n)$ where $\mathcal{E}_n = \{(\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{X}^2 \mid \mathbf{x}_2 \in \mathcal{N}_n(\mathbf{x}_1)\}$. Each connected component of the graph G corresponds to a different neutral network. In other words, a neutral network is a connected subgraph $G' = (\mathcal{X}', \mathcal{E}'_n)$ of G , $\mathcal{X}' \subseteq \mathcal{X}$ and $\mathcal{E}'_n \subseteq \mathcal{E}_n$, where (i) there exists a path connecting any pair of solutions $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}'$, and (ii) there exists no edge $(\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{E}_n \setminus \mathcal{E}'_n$ such that $\mathbf{x}_1 \in \mathcal{X}'$ and

$\mathbf{x}_2 \in \mathcal{X} \setminus \mathcal{X}'$. The neutral network of a solution \mathbf{x} will be denoted as $NN(\mathbf{x})$. Finally, another important concept is that of a *neutral walk*. A neutral walk from \mathbf{x}_1 to \mathbf{x}_k refers to a sequence of solutions $\langle \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k \rangle$ such that $\mathbf{x}_{i+1} \in \mathcal{N}_n(\mathbf{x}_i)$, $1 \leq i < k$. That is, a neutral walk represents a path on a neutral network.

2.3. The HP model for protein structure prediction

Proteins are fundamental elements of living organisms. These chain-like molecules are composed from a set of 20 different building blocks called amino acids. The specific sequence of amino acids of a protein determines how it folds into a unique compact conformation responsible for its biological functioning. Among all the possible conformations that a protein can adopt, it is believed that the optimal conformation, often referred to as the native state, corresponds to the one minimizing the overall free-energy [22]. Thus, the *protein structure prediction* (PSP) problem, can be stated as the problem of finding the functional, energy-minimizing conformation for a protein given only its amino acid sequence.

Amino acids can be classified on the basis of their affinity for water. *Hydrophilic* or *polar* amino acids (*P*) are usually found at the outer surface of proteins. By interacting with the aqueous environment, *P* amino acids contribute to the solubility of the molecule. In contrast, *hydrophobic* or *nonpolar* amino acids (*H*) tend to pack on the inside of proteins, where they interact with one another to form a water-insoluble core. This phenomenon, usually referred to as *hydrophobic collapse*, is a major driving force in protein folding, representing the reasoning and motivation behind the hydrophobic-polar (HP) model of the PSP problem studied in this paper [1, 2].

In the HP model, proteins are abstracted as chains of *H*- and *P*-type beads. Protein sequences, which are originally defined over a 20-letters alphabet, are now of the form $S = \langle a_1, a_2, \dots, a_\ell \rangle$, where $a_i \in \{H, P\}$ denotes the *i*-th amino acid and ℓ is the length of the sequence. A protein conformation is represented in this model as an embedding of the protein chain on a given lattice. Both the two-dimensional square and three-dimensional cubic lattices are considered in this study. For a protein conformation to be considered feasible (*i.e.*, valid), its corresponding embedding on the lattice is required to satisfy different properties, as detailed in Section 2.3.2. With the aim of emulating the so-called hydrophobic collapse, the goal in the HP model is to maximize the interaction among *H* amino acids in the lattice. Such interactions are to be referred to as *H-H topological contacts*. Two *H* amino acids a_i and a_j are said to form a topological contact if they are nonconsecutive in S but adjacent in the lattice. The objective is thus to find a feasible protein conformation where the number of *H-H* topological contacts, *HHtc*, is maximized. Adhering to the notation of the field, an energy function, to be minimized, is defined as the negative of *HHtc*.

Let \mathcal{X} be the set of all potential protein conformations, *i.e.*, the search space, and let $\mathcal{X}_{\mathcal{F}}$ denote the subset of all the feasible states ($\mathcal{X}_{\mathcal{F}} \subsetneq \mathcal{X}$). PSP under the HP model can be formally defined as the problem of finding $\mathbf{x}^* \in \mathcal{X}_{\mathcal{F}}$ such that $E(\mathbf{x}^*) = \min\{E(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}_{\mathcal{F}}\}$. The energy function $E : \mathcal{X} \rightarrow \mathbb{R}$ maps

³In this study, a fitness function is assumed to be always maximized (the goal is to search for the fittest solution candidate).

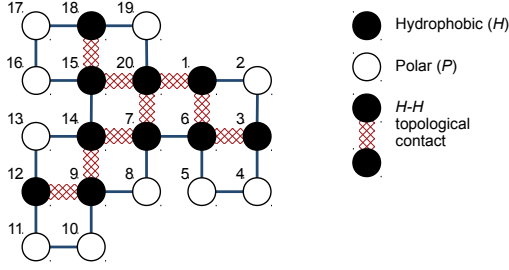


Figure 1: Optimal conformation for sequence 2d4 of length $\ell = 20$ on the two-dimensional square lattice. Black and white beads denote H and P amino acids, respectively. Amino acids have been numbered from 1 to ℓ according to their positions in the protein sequence S . The energy of this conformation is $E = -9$, since there are 9 H - H topological contacts, $HHtc = 9$.

each possible conformation $\mathbf{x} \in \mathcal{X}$ to an energy value:

$$E(\mathbf{x}) = \sum_{a_i, a_j} e(a_i, a_j), \quad (4)$$

where

$$e(a_i, a_j) = \begin{cases} -1, & \text{if } a_i \text{ and } a_j \text{ are both } H \text{ and} \\ & \text{they form a topological contact;} \\ 0, & \text{otherwise.} \end{cases}$$

As an example, the optimal structure for a protein sequence of length $\ell = 20$ on the two-dimensional square lattice is presented in Figure 1. This example corresponds to sequence 2d4, one of the adopted test cases for the HP model (Appendix A.2). The prediction of protein structures based on the HP model represents a hard combinatorial optimization problem which has been proved to be NP-complete [3, 4]. An extensive literature exists on the use of metaheuristic approaches to address this problem, including genetic algorithms [23, 24], memetic and hybrid algorithms [25–27], tabu search [28], ant colony optimization [29, 30], immune-based algorithms [8, 31], particle swarm optimization [32, 33], differential evolution [9, 34], estimation of distribution algorithms [35, 36], artificial plant optimization [37], and firefly-inspired algorithms [38].

2.3.1. Representation of solutions

In this study, an *internal coordinates representation* based on *relative moves* has been considered [39]. That is, a protein conformation is encoded as a sequence of moves specifying the lattice position for each amino acid with regard to the preceding one. On the three-dimensional cubic lattice, conformations are thus defined as sequences in $\{F, L, R, U, D\}^{\ell-2}$, to denote the forward, left, right, up and down moves from one amino acid to the next. Only moves $\{F, L, R\}$ are considered when using the two-dimensional square lattice. The relative moves encoding implements a local reference system which rotates at each encoding decision (other than F). An example is provided in Figure 2. Note that no backward moves are allowed, ensuring that the encoded conformations will always be *one-step self-avoiding*. Finally, note also that only $\ell - 2$ encoding decisions need to be taken by fixing the position of the first amino acid and assuming the first move to be forward (F).



Figure 2: Internal coordinates representation based on relative moves. Encoding scheme (left). Structure encoded as FLFRRLRR (right).

2.3.2. Problem constraints

In the HP model, a feasible protein conformation defines an embedding of the protein chain on the lattice such that two different properties are satisfied: *connectivity* and *self-avoidance*. On the one hand, connectivity requires consecutive amino acids in the protein sequence to be placed at adjacent positions of the lattice. On the other hand, self-avoidance implies that the conformation has to be free of collisions; *i.e.*, two different amino acids can not be assigned to the same lattice position. While connectivity is implicitly satisfied by using an internal coordinates representation, as described in Section 2.3.1, such a representation scheme can not ensure the self-avoidance of the encoded conformations; refer to the example provided in Figure 3. Therefore, an explicit mechanism is required to be implemented in order to address the self-avoidance constraint.

3. Related work

3.1. Constraint-handling in the HP model

In the literature, two basic directions have been taken to address the self-avoidance constraint which relates to the feasibility of protein conformations in the HP model of the PSP problem. On the one hand, the search can concentrate on the feasible space; that is, considering only solutions encoding self-avoiding protein conformations. This is usually accomplished either (i) by adapting the variation operators to iterate until new feasible conformations are generated, *i.e.*, infeasible conformations are always rejected [6, 31, 40–43]; (ii) by using specialized operators which are closed on the feasible space, *i.e.*, always transforming feasible conformations into other feasible conformations [7, 44, 45]; or (iii) by implementing repairing procedures in order to convert from infeasible to feasible conformations [7, 9, 36, 46]. These three constraint-handling strategies can be referred to as the *rejecting*, *preserving* and *repairing strategies*, respectively [47]. On the other hand, infeasible conformations can also be allowed to participate during the search process. This is commonly achieved by implementing a *penalty strategy*, where the energy value of a candidate conformation suffers a decrease according to the number of collisions (overlaps) in encoded protein structure [5, 8, 39, 48, 49].

It has been argued that the path from one compact feasible conformation to another, can be significantly shorter if the search is allowed to proceed through the space of infeasible conformations [5]. This has been, perhaps, the main motivation for applying penalty strategies when dealing with the HP model. An example of this scenario was given by Krasnogor *et al.* [5]. Also, the authors provided some guidelines on how to

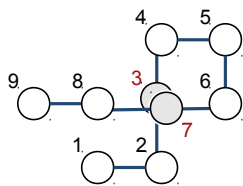


Figure 3: An infeasible protein conformation encoded (using the relative moves representation) as FLFRRRFF. A collision was produced when amino acids 3 and 7 were mapped to the same lattice position.

design an appropriate penalty function for this problem. Nevertheless, no experimental results were reported to support these recommendations. In [6], Duarte-Flores and Smith compared between the performance of two variants of a genetic algorithm (GA). The first GA implemented a rejecting strategy, iterating crossover and mutation until feasible offspring were generated. The second GA used a penalty function adhering to the guidelines provided by Krasnogor *et al.* [5]. As a result, a better performance was observed from the use of the first variant of the GA, where infeasible conformations were always discarded.

Using a GA, Cotta compared the use of a penalty function with respect to two alternative approaches [7]. In the first, referred to as the feasible-space approach, the crossover and mutation operators were adapted to produce only feasible offspring. In the second alternative, infeasible offspring were accepted as the output of variation operators, but they were subsequently processed using a repairing procedure. Both the two alternative approaches were based on a backtracking algorithm. As reported in [7], better results were obtained in most of the cases using the penalty method when compared to the feasible-space approach, but the best overall performance of the implemented GA was obtained by using the repairing procedure.

Almeida *et al.* [8] explored the influence of allowing infeasible conformations on the performance of an immune-based algorithm (IA). In a first variant of the IA, only feasible conformations were permitted. In a second variant of the IA, infeasible individuals were accepted during the initialization process and when new random individuals were required as a consequence of applying the *aging operator* (infeasible individuals were penalized). However, in both variants of the IA the *hypermutation* and *hypermacromutation* operators were adapted to produce only feasible conformations; infeasible mutations were always rejected. The inclusion of infeasible solutions slightly increased the performance of the IA in most of the cases, while significantly reducing the computational effort.

Santos and Diéguez [9] evaluated the advantages of incorporating a repairing strategy into their differential evolution (DE) algorithm. In an initial DE implementation, all infeasible conformations were simply penalized and assigned a fitness value of 0. In a second DE variant, two different repairing operators were implemented; the first working on the amino acid coordinates (phenotype space), and the second acting on the conformation encoding (genotype space). These repairing operators, however, were not based on a backtracking strategy as those explored by Cotta [7]; whenever the position of the colliding amino acids could not be repaired, the infeasible conformations

were allowed to remain in the population. The reported results indicate that the use of the repairing operators improved the search performance of the proposed DE algorithm.

Summarizing, there is not strong evidence in the literature (from the authors' point of view) regarding whether it can be better to allow or to prevent infeasible protein conformations from being considered during the search process. Rather, from the above discussed works, it is possible to note that very different and, to some extent, contradictory results have been reported in this respect. One of the aims of the present study is to contribute in providing further insight into this matter.

3.2. Single-objective to multi-objective transformations

Restating a single-objective problem in terms of two or more objective functions, a process known as *multi-objectivization* [50], represents a current and promising research direction which has served as the basis for the development of more competitive search algorithms. Multi-objectivization can be done either through the *decomposition* of the original objective function of the problem [50, 51], or by means of the addition of new *supplementary objectives* [52, 53]. In either case, this transformation can largely facilitate the process of solving the original problem. A recent review on this topic was reported in [12].

In the former approach, multi-objectivization by decomposition, the original objective is split into several components, each treated as an objective function under the new alternative problem formulation. It has been shown that the only possible effect of decomposition is the introduction of plateaus in the fitness landscape [51]. Originally comparable solutions may become incomparable (nondominated in terms of the Pareto-dominance relation) with regard to the new multi-objectivized formulation. Such an effect can be potentially exploited as a means of escaping from local optima [50, 51]. Some of the works reported in this direction include the following: [50, 51, 54]. Also, multi-objective approaches which have been proposed for the PSP problem fall into this category, either those focusing on detailed (all-atom) energy models [55–62], or those based on the HP model as explored in the authors' previous work [63–65].

In the latter approach to multi-objectivization, additional information is incorporated in the form of one or more supplementary objectives (also often referred to as *artificial* or *helper objectives*). As analyzed in [53], two different effects can be achieved through this transformation. On the one hand, incomparability among solutions can be introduced, such as it occurs when multi-objectivizing by decomposition. This is due to the conflict between the original and the new defined optimization criteria. On the other hand, this transformation may also break existing plateaus, allowing originally incomparable solutions to become comparable as a consequence of the new considered information. In the literature, the addition of objective functions represents the most extensively studied approach to multi-objectivization. In [12], a distinction is made between multi-objectivization proposals where supplementary objectives are problem-dependent and are computed based solely on information from the solution under consideration [52, 66–69], and those where they act as diversity measures [70–75].

In [12], separate treatment is given to those proposals where additional objective functions are implemented as a strategy to deal with constrained problems. In such methods, the constrained single-objective problem is reformulated as an unconstrained multi-objective problem [76–79]. A number of successful applications of the multi-objective approach to constraint-handling have been reported [11, 12]. This is the type of transformation this paper studies in the context of the HP model of the PSP problem (preliminary results were reported in [13]). As discussed more in detail in Section 4, this constraint-handling strategy exploits the effects of multi-objectivization in such a way that infeasible solutions can become incomparable with respect to feasible ones, which allows them to be potentially considered during the search process. An important contribution of this paper is the analysis of how this effect changes the fitness landscape and can influence the behavior of search algorithms. Though other different constraint-handling approaches for the HP model have been adopted in order to accept infeasible solutions during the search process, *e.g.*, penalty strategies, an advantage of the multi-objective approach is that it usually involves fewer parameters that need to be set [11, 12, 14]. Nevertheless, it has been shown that the lack of a search bias represents a critical issue that can compromise the effectiveness of this technique [14]. Therefore, part of this work is devoted to inquire further into this important subject.

4. Handling constraints in the HP model by multi-objective optimization

It is the authors’ belief that considering infeasible protein conformations during optimization can boost the performance of metaheuristics for solving PSP under the HP model (arguments on this respect have also been given in the literature [5]). Therefore, it is important to devise new constraint-handling mechanisms, which allow these algorithms to exploit the vast amount of infeasibility that the HP model involves, as a means of steering the search process in a more effective manner.

The use of multi-objective optimization is here explored as an alternative constraint-handling strategy for the HP model of the PSP problem. The HP model is restated in multi-objective form by incorporating an additional objective function which accounts for the problem constraints. In this way, this originally constrained single-objective optimization problem is transformed into an unconstrained multi-objective one. More formally, a two-objective formulation of the problem, $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$, is defined as follows ($\mathbf{x} \in \mathcal{X}$):

$$f_1(\mathbf{x}) = E(\mathbf{x}), \quad (5)$$

$$f_2(\mathbf{x}) = \text{Collisions}(\mathbf{x}), \quad (6)$$

where $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are to be minimized; $E(\mathbf{x})$ is the conventional energy function of the HP model, as defined in Section 2.3; and $\text{Collisions}(\mathbf{x})$ denotes the total number of colliding amino acid pairs (a_i, a_j) such that both a_i and a_j were mapped to the same lattice coordinates in the encoded protein structure.

Using the above described multi-objective formulation of the HP model, all feasible solutions $\mathbf{x} \in \mathcal{X}_{\mathcal{F}}$ will feature a value

of $f_2(\mathbf{x}) = 0$. Thus, the original characteristics of the feasible areas of the fitness landscape are preserved. That is, the Pareto-dominance relation induces the conventional rank ordering among feasible solutions based on the original optimization objective (f_1). Moreover, since feasible solutions present the best possible value in f_2 , an infeasible solution (with $f_2 > 0$) will never be preferred over a feasible solution under the multi-objective formulation. In general, the alternative multi-objective formulation will lead to the explicit consideration of trade-offs between the two defined criteria, f_1 and f_2 . An infeasible conformation \mathbf{x}_1 may become incomparable, *i.e.*, nondominated in the Pareto sense, with respect to a feasible conformation \mathbf{x}_2 . This depends upon how \mathbf{x}_1 and \mathbf{x}_2 compare to each other with regard to the primary objective function f_1 . Therefore, the multi-objective approach for handling constraints allows infeasible conformations to compete against feasible ones, being potentially accepted and exploited during the search process.

5. Fitness landscape transformation

Whereas infeasible solutions are usually regarded and treated as inferior, or even as inadmissible solutions during the search process, such a distinction between feasible and infeasible solutions is not captured when handling constraints by multi-objective optimization. As discussed in Section 4, the multi-objective strategy allows infeasible solutions to become incomparable, under certain conditions, with respect to feasible ones. Such an effect of the problem transformation leads to an increase in the neutrality of the fitness landscape. That is, given a feasible solution $\mathbf{x} \in \mathcal{X}_{\mathcal{F}}$, some of the surrounding infeasible solutions may become incomparable with regard to \mathbf{x} , thus becoming members of its neutral neighborhood, $N_n(\mathbf{x})$.

In this section, an analysis is conducted with the aim of investigating the extent to which the multi-objective constraint-handling strategy impacts on the neutrality of the HP model’s fitness landscapes. A fitness landscape is defined by a triplet $(\mathcal{X}, \mathcal{N}, \xi)$, as described in detail in Section 2.2.⁴ Two variants of the evaluation scheme ξ have been considered: the conventional single-objective (SO) formulation of the problem, and the alternative multi-objective (MO) formulation that handles constraints. In this way, by analyzing and comparing the landscapes induced by the two different evaluation schemes, it will be possible to assess and to gain further understanding of the effects that the studied problem transformation involves.

As stated in Section 2.3, the quality of a candidate solution in the HP model is evaluated in terms of an energy function, E , defined as the negative of the total number of H - H topological contacts that the encoded protein structure presents, HH_{tc} . Nevertheless, the use of positive rather than negative values, as well as the adoption of the term fitness (to be maximized) rather than that of energy (to be minimized), is considered more appropriate for the analysis here reported. Therefore, in the re-

⁴Both \mathcal{X} and \mathcal{N} were fixed during the analysis here presented. \mathcal{X} is defined by the relative moves encoding described in Section 2.3. $\mathcal{N}(\mathbf{x})$ is given by all possible single-variable perturbations of \mathbf{x} . Thus, $|\mathcal{N}(\mathbf{x})| = 2(\ell - 2)$ and $|\mathcal{N}(\mathbf{x})| = 4(\ell - 2)$ in the two- and three-dimensional lattices, respectively.

mainder of this section the fitness of a solution \mathbf{x} , $Fitness(\mathbf{x})$, will assume the value of $HHtc(\mathbf{x})$, i.e.,

$$Fitness(\mathbf{x}) = HHtc(\mathbf{x}) = -E(\mathbf{x}). \quad (7)$$

It is worthy to mention at this point that the term fitness is used in this study to refer to the quality of (feasible) solutions under the conventional, SO evaluation scheme of the HP model.⁵ In addition, it is important to briefly introduce the concept of a *fitness class*; a solution $\mathbf{x} \in X_{\mathcal{F}}$ will be said to belong to the fitness class c if it presents a fitness value of $Fitness(\mathbf{x}) = c$.

The analysis presented in this section requires an initial sample of solutions for each of the considered test instances. These samples were collected following the methodology described in Section 5.1. The fitness landscapes are investigated in this study by evaluating different properties of neutral networks (NNs); since it is known that neutral fitness landscapes, as those in the HP model, can be mainly described by their NNs [80]. Section 5.2 details the implemented procedure for sampling the NNs. Finally, the results of the performed fitness landscape analysis are discussed in Section 5.3.

5.1. Sampling of initial solutions

The implemented sampling strategy was conceived by taking into account the following considerations: (i) a total of M different feasible solutions for the given problem instance are to be generated; (ii) the M generated solutions are to be, if possible, evenly distributed over the different available fitness classes (all fitness classes should be well represented in the collected sample); and finally, (iii) the diversity among solutions belonging to the same fitness class should be maximized.

Algorithm 1 outlines the adopted sampling strategy. The procedure starts by initializing the sample set \mathcal{S} and by identifying the set of all possible fitness classes for the given problem instance, \mathcal{FC} (lines 1 and 2 in Algorithm 1). Iteratively, a search algorithm is executed and all solutions that this algorithm reaches during the search process are kept in \mathcal{U} (line 4). Then, the subset \mathcal{U}_c of solutions in \mathcal{U} belonging to each possible fitness class $c \in \mathcal{FC}$ is identified (line 6). Finally, the solution $\hat{\mathbf{x}} \in \mathcal{U}_c$ that best contributes to increasing the diversity in \mathcal{S} , if any, is included in the sample (lines 7 to 9). This process continues until completing the required sample.

Any metaheuristic could be implemented as the embedded search method. An Iterated Local Search (ILS) algorithm [81], based on the SO problem formulation, was used in this study; refer to Appendix B.4 for details. Due to its distinctive exploration behavior, the ILS method can potentially reach a different local optimum at each iteration. Each time the ILS was invoked during the sampling procedure, this algorithm was allowed to run for a total of 5×10^5 solution evaluations.

The diversity contribution estimates have been partially based on the diversification mechanism proposed by Chira [42]. Instead of measuring diversity in genotype (encoding) space,

⁵Although an alternative multi-objective formulation of the HP model is implemented as a constraint handling strategy, the goal remains always to solve the original single-objective problem.

Algorithm 1 Sampling of the initial solution sets.

Require: M
Ensure: \mathcal{S}
1: $\mathcal{S} \leftarrow \emptyset$
2: $\mathcal{FC} \leftarrow \{Fitness(\mathbf{x}) \mid \mathbf{x} \in X_{\mathcal{F}}\}$
3: **while** $|\mathcal{S}| < M$ **do**
4: $\mathcal{U} \leftarrow search_algorithm()$
5: **for all** $c \in \mathcal{FC}$ **do**
6: $\mathcal{U}_c \leftarrow \{\mathbf{x} \in \mathcal{U} \mid Fitness(\mathbf{x}) = c\}$
7: $\hat{\mathbf{x}} \leftarrow \arg \max_{\mathbf{x} \in \mathcal{U}_c} diversity(\mathbf{x}, \mathcal{S})$
8: **if** $diversity(\hat{\mathbf{x}}, \mathcal{S}) > 0$ **then**
9: $\mathcal{S} \leftarrow \mathcal{S} \cup \{\hat{\mathbf{x}}\}$
10: **end if**
11: **end for**
12: **end while**

in [42] diversity was computed from the *contact fingerprint* of candidate solutions. The contact fingerprint for a solution is given by the binary vector \mathbf{cf} , where each component $cf_i \in \{0, 1\}$ indicates whether a particular pair of amino acids in the encoded structure defines a topological contact or not. Vector \mathbf{cf} considers as many components as the total number of amino acid pairs which can potentially form a topological contact.⁶ The use of the contact fingerprint rather than the encoding of solutions certainly fosters the development of more effective diversity promotion mechanisms. This can be explained by the fact that very different encodings may represent the same protein structure (after rotation or reflection). Note, however, that significantly different structures may also present the same \mathbf{cf} vector if they share the same set of topological contacts. This has motivated the use of a more fine-grained version of this approach, referred to in this study as the *distance fingerprint*.

The distance fingerprint for a given solution is defined by the vector \mathbf{df} , each of whose components df_i measures the distance between the lattice coordinates of a particular pair of amino acids. The Manhattan distance was employed for this sake. A total of $\binom{\ell}{2} - 2\ell + 3$ components describe the distance fingerprint vector \mathbf{df} ; i.e., only amino acid pairs (a_i, a_j) such that $|j - i| \geq 3$ require to be considered. Finally, the diversity contribution for a new candidate \mathbf{x} with respect to the already collected sample \mathcal{S} , $diversity(\mathbf{x}, \mathcal{S})$, has been computed as the minimum Hamming distance (H_d) between the distance fingerprint vector of \mathbf{x} and that of any $\mathbf{x}' \in \mathcal{S}$ with the same fitness value as \mathbf{x} . Formally,

$$diversity(\mathbf{x}, \mathcal{S}) = \min \{ H_d(\mathbf{df}(\mathbf{x}), \mathbf{df}(\mathbf{x}')) \mid \mathbf{x}' \in \mathcal{S} \wedge Fitness(\mathbf{x}) = Fitness(\mathbf{x}') \}. \quad (8)$$

5.2. Sampling of neutral networks

Neutral networks (NNs) in a neutral fitness landscape can be considerably large, so that their exhaustive exploration becomes computationally prohibitive even for relatively small problem instances. In the literature, NNs are usually sampled through

⁶In order for an amino acid pair (a_i, a_j) to form a topological contact, i and j need to be of opposite parity and $|j - i| \geq 3$.

Algorithm 2 $pNN()$ - partial computation of neutral networks.**Require:** \mathbf{x} , $depthLevel$, $maxDepth$ **Ensure:** $NN(\mathbf{x})$

```

1:  $NN(\mathbf{x}) \leftarrow (\mathcal{V}, \mathcal{E}) : \mathcal{V} = \{\mathbf{x}\}, \mathcal{E} = \emptyset$ 
2: if  $depthLevel < maxDepth$  then
3:   for all  $\mathbf{x}' \in \mathcal{N}_n(\mathbf{x})$  do
4:      $NN(\mathbf{x}') \leftarrow pNN(\mathbf{x}', depthLevel + 1, maxDepth)$ 
5:      $NN(\mathbf{x}) \leftarrow NN(\mathbf{x}) \cup NN(\mathbf{x}')$ 
6:      $\mathcal{E} \leftarrow \mathcal{E} \cup \{(\mathbf{x}, \mathbf{x}')\}$ 
7:   end for
8: end if

```

neutral walks [18, 80, 82]. In this study, however, an alternative approach has been taken, as described below.

Given a sample set \mathcal{S} of M different solution candidates, the neutral network $NN(\mathbf{x})$ for each $\mathbf{x} \in \mathcal{S}$ has been partially computed based on the $pNN()$ procedure presented in Algorithm 2. As shown in this algorithm, $NN(\mathbf{x})$ is constructed recursively in a depth-first manner by allowing this procedure to reach a maximum defined depth level ($maxDepth$). The initially given solution \mathbf{x} is assumed to be at depth level 0 ($depthLevel = 0$). At each call to the $pNN()$ method, $NN(\mathbf{x})$ is first initialized to the graph containing no edges and including the provided solution \mathbf{x} as the only node (line 1 in Algorithm 2). If the maximum allowed depth level has not been reached (line 2), the sub-network $NN(\mathbf{x}')$ for every neutral neighbor \mathbf{x}' of \mathbf{x} is obtained from a subsequent execution of $pNN()$ (i.e., by giving \mathbf{x}' as the new starting point and by increasing the value of $depthLevel$, line 4). The resulting sub-network $NN(\mathbf{x}')$ is then merged with the parent network $NN(\mathbf{x})$ by means of a graph union operation, here denoted as \cup (line 5).⁷ Finally, edge $(\mathbf{x}, \mathbf{x}')$ is included in $NN(\mathbf{x})$ in order to establish the linkage between $NN(\mathbf{x})$ and the $NN(\mathbf{x}')$ sub-network. Note that this strategy requires a mechanism to prevent cycling (revisiting solutions and computing their NN repeatedly and indefinitely).⁸ Partially computing the NN for a given solution \mathbf{x} is equivalent to traversing all possible neutral walks departing from \mathbf{x} , by restricting the length of the walks to the maximum defined depth level ($maxDepth$).

5.3. Fitness landscape analysis

This section presents the results of the conducted fitness landscape analysis. Due to the high computational costs involved, and given also the space requirements for reporting results, this analysis has focused on two (relatively) small problem instances: the 2d4 and 3d1 test sequences for the two- and three-dimensional lattices, respectively (see Appendix A.2).⁹ It is expected, however, that similar results can be obtained by

Table 1: Sample sets generated for instances 2d4 and 3d1 (instance 2d4 involves 10 different fitness classes, and 3d1 involves 12).

Seq.	Fitness class												Total
	0	1	2	3	4	5	6	7	8	9	10	11	
2d4	119	119	119	119	119	118	118	119	48	2	-	-	1000
3d1	84	83	83	84	83	84	83	83	83	84	83	83	1000

replicating this analysis to other instances. As indicated before, the performed analysis relies on initial solution samples. These samples were generated through the methodology detailed in Section 5.1. The size of the samples was fixed to $M = 1000$ for both the 2d4 and 3d1 instances. Details of the obtained sample sets are provided in Table 1. By following the adopted sampling methodology, it is (ideally) expected to generate sample sets such that about $M/|\mathcal{FC}|$ different solutions represent each possible fitness class $c \in \mathcal{FC}$. As shown in Table 1, this was the case of the sample set constructed for instance 3d1, where a total of 83 or 84 different solutions were produced for each of the $|\mathcal{FC}| = 12$ available fitness classes. Note, however, that because of the funnel-like energy landscape which characterizes the HP model [83], not all fitness classes for some of the instances can be equally sampled. Only a reduced number of solutions with a high fitness value (fitness classes 8 and 9) were obtained when sampling the search space of instance 2d4. Therefore, a greater number of representatives for the remaining fitness classes were accepted in order to complete the M required solutions.

The neutral network (NN) of all the generated solutions has been explored by using both, the SO and MO evaluation schemes, as the bases for neutrality verification. In this way, changing the problem formulation from SO to MO was reflected as an alteration in the properties of the sampled NNs. The NNs were partially computed as described in Section 5.2, and the maximum allowed depth level in this procedure was set to $maxDepth = 5$ (for both the 2d4 and 3d1 instances) in order to alleviate the computational burden.¹⁰ In the remainder of this section, the NN for a given solution \mathbf{x} will be either referred to as $NN_{SO}(\mathbf{x})$ or $NN_{MO}(\mathbf{x})$, depending on whether it has been computed based on the SO or MO evaluation schemes. It is important to note that under the SO evaluation scheme all infeasible solutions are assumed to be inferior to any feasible one, so that $NN_{SO}(\mathbf{x})$ comprises only feasible states. In contrast, $NN_{MO}(\mathbf{x})$ may also involve infeasible nodes; this is because using the MO formulation an infeasible solution may become part of the neutral neighborhood, and therefore of the NN, for a feasible solution, as discussed in the preamble of Section 5.

5.3.1. Average neutrality and infeasibility ratios

In this section, the *average neutrality ratio* (ANR) is explored as a first step in analyzing the degree to which neutrality is affected by the studied problem transformation. The ANR is given by the arithmetic mean of the neutrality ratios, as defined previously in Section 2.2, for all the solutions in a NN

⁷Given $G_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $G_2 = (\mathcal{V}_2, \mathcal{E}_2)$, the graph union operation $G_1 \cup G_2$ produces $G_3 = (\mathcal{V}_3, \mathcal{E}_3)$ such that $\mathcal{V}_3 = \mathcal{V}_1 \cup \mathcal{V}_2$ and $\mathcal{E}_3 = \mathcal{E}_1 \cup \mathcal{E}_2$.

⁸In the implemented mechanism, the encoding of a given solution is treated either as a base-3 or as a base-5 number (two- and three-dimensional cases, respectively), which is then converted to a base-10 number and inserted into a self-balancing binary search tree. This enables efficient searches and reduces memory consumption compared to storing the complete solution encoding.

⁹Even for these relatively small test cases the search space (using the relative moves encoding) is vast: 3^{18} for 2d4, and 5^{18} for 3d1.

¹⁰In spite of the use of a low value of $maxDepth = 5$, the resulting NNs were considerably large, as it will be analyzed in Section 5.3.2.

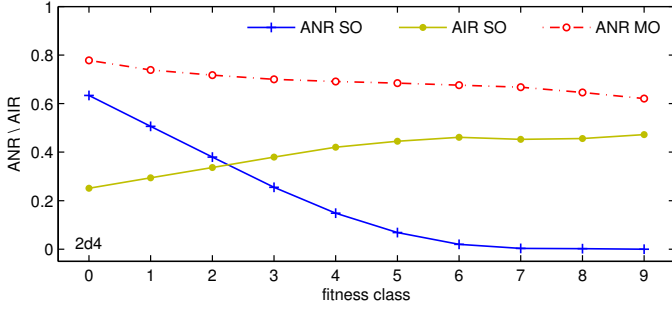


Figure 4: Average neutrality (ANR) and infeasibility (AIR) ratio, 2d4.

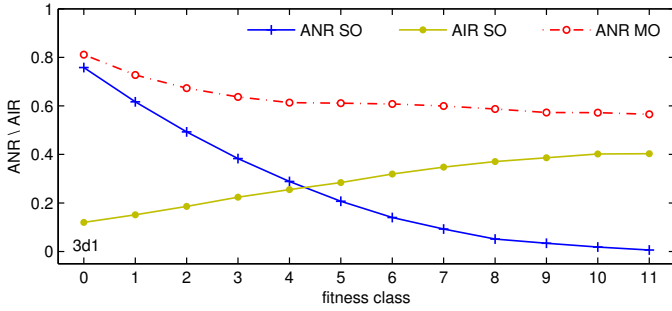


Figure 5: Average neutrality (ANR) and infeasibility (AIR) ratio, 3d1.

[19, 84]. This measure assumes values in the range $[0, 1]$, where $\text{ANR} = 1$ indicates the highest neutrality. The ANR measure was computed for all the sampled NN_{SO} and NN_{MO} networks, so that it will be possible to contrast the neutrality that the two different evaluation schemes (SO and MO) produce. In addition, the *average infeasibility ratio* (AIR) is investigated for the SO problem formulation. AIR is defined analogously to ANR, but calculated from the number of infeasible (rather than neutral) neighbors of solutions in the NN.¹¹ The results for the 2d4 and 3d1 instances are presented in Figures 4 and 5, where the mean ANR and AIR values appear organized (in the x-axis) according to the fitness class of the solution provided as the starting point for the NN sampling.

From the ANR and AIR values obtained through the use of the SO formulation (“ANR SO” and “AIR SO” curves in the plots), it is possible to highlight some general tendencies with regard to the neutrality and infeasibility of the HP model’s fitness landscapes. On the one hand, the poorer the quality of a solution, the greater tends to be the number of neutral mutations that the solution can produce. This is suggested by the high ANR values scored for the lowest fitness classes, which rapidly decreased with the increase in fitness. On the other hand, infeasibility becomes more abundant as superior fitness classes are considered. Solutions at the best fitness classes are usually surrounded by infeasible neighbors; as the obtained AIR values indicate, between 40% and 50% of the neighborhood for solutions at the best fitness classes is composed of infeasible states.

¹¹It is worthwhile to remember that the MO evaluation scheme does not distinguish between feasible and infeasible solutions. Hence, the AIR measure applies only to the SO formulation of the problem.

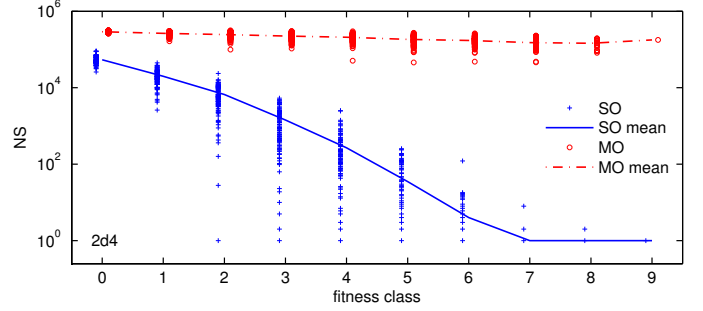


Figure 6: Size of the sampled neutral networks (NS). 2d4 test instance.

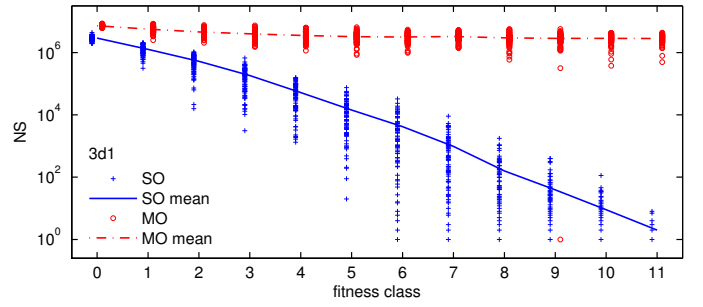


Figure 7: Size of the sampled neutral networks (NS). 3d1 test instance.

The above observations can be explained by the fact that fitness, given in terms of the number of H - H topological contacts, is directly related to the compactness of the encoded protein conformations. The higher the fitness value, the more compact the protein tends to be. Hence, it is reasonable to conjecture that most perturbations to the encoding of a compact conformation could lead either to an infeasible solution, or to a less folded structure which worsens the fitness. Finally, the use of the MO formulation of the problem reports an important increase in the ANR measure, with respect to the SO formulation, for all fitness classes of the two considered test instances. By introducing incomparability between feasible and infeasible solutions, a substantial fraction of the infeasibility has been translated into landscape neutrality due to the problem transformation.

5.3.2. Size of the neutral networks

This section investigates how the change in the problem formulation has impacted on the size (number of solutions) of the NNs. Figures 6 and 7 report the size of each computed NN_{SO} and NN_{MO} network, as well as the arithmetic mean for the different fitness classes. Notice that results in these figures are presented in a logarithmic (base-10) scale. These figures expose the high neutrality that characterizes the HP model’s fitness landscapes. Even though the sampling of NNs was restricted in this study by defining a maximum allowed depth level ($\text{maxDepth} = 5$), using the SO formulation the NNs at fitness class 0 are composed of about 10^5 and 10^6 solutions for the 2d4 and 3d1 instances, respectively. The high neutrality which can be found at low fitness classes leads to the formation of large NNs. On the contrary, solutions at the best fitness classes tend to be more isolated and enclosed by infeasible

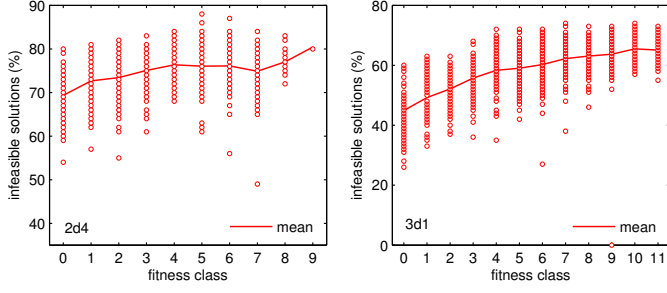


Figure 8: Percentage of infeasible solutions in the NNs sampled using the MO formulation. Instance 2d4 (left) and instance 3d1 (right).

states, as it was also discussed in Section 5.3.1. The increase in the neutrality ratios originated from the use of the MO formulation, has led to a significant rise in the size of the sampled NNs. As it can be seen from the plots, NNs computed based on the MO formulation can be several orders of magnitude larger than those computed based on the SO formulation. This behavior becomes more evident as higher fitness classes are considered. It is important to realize that, given a solution \mathbf{x} , $NN_{MO}(\mathbf{x})$ will always be a supergraph containing all nodes and edges of $NN_{SO}(\mathbf{x})$, but including also those nodes and edges which result from the neutrality introduced by the problem transformation. Thus, $NN_{MO}(\mathbf{x})$ will have at least the same size as $NN_{SO}(\mathbf{x})$. It is worth mentioning that only slight variations in the size of NN_{MO} networks can be perceived across the different fitness classes. Finally, it becomes relevant for this study the question of to what extent the sampled NN_{MO} networks are composed of infeasible solutions. Figure 8 addresses this question. Despite that a feasible solution was given in all the cases as the starting point for the NNs exploration, the bulk of NN_{MO} networks consists of infeasible states. According to Figure 8, between 70% and 80% of the nodes, in average, were found to be infeasible when focusing on the two-dimensional instance (respectively, between 50% and 70% for the three-dimensional case).

5.3.3. Connectivity between neutral networks

The introduction of neutrality into the fitness landscape leads to the formation of *neutral connections* between NNs. A neutral connection between NN_1 and NN_2 implies that, as a result of the problem transformation, (at least) a solution \mathbf{x}_1 from NN_1 became part of the neutral neighborhood of another solution \mathbf{x}_2 which belongs to NN_2 . Through such a neutral connection, NN_1 and NN_2 are merged together into a larger NN involving all nodes and edges of the original networks (plus the new edge(s) giving rise to the neutral connection).

In the particular context of the landscape transformation induced by the studied MO constraint-handling strategy, a neutral connection between two NNs can occur, if and only if, one of the two networks is feasible and the other infeasible (incomparability can only be introduced between a feasible and an infeasible state, see Section 4). Note, however, that two feasible NNs can be merged through a succession of neutral connections. More precisely, the linkage between two feasible networks NN_1 and NN_k can be given in the form of a sequence $\langle NN_1, NN_2, \dots, NN_{k-1}, NN_k \rangle$, such that each NN_i is neutrally

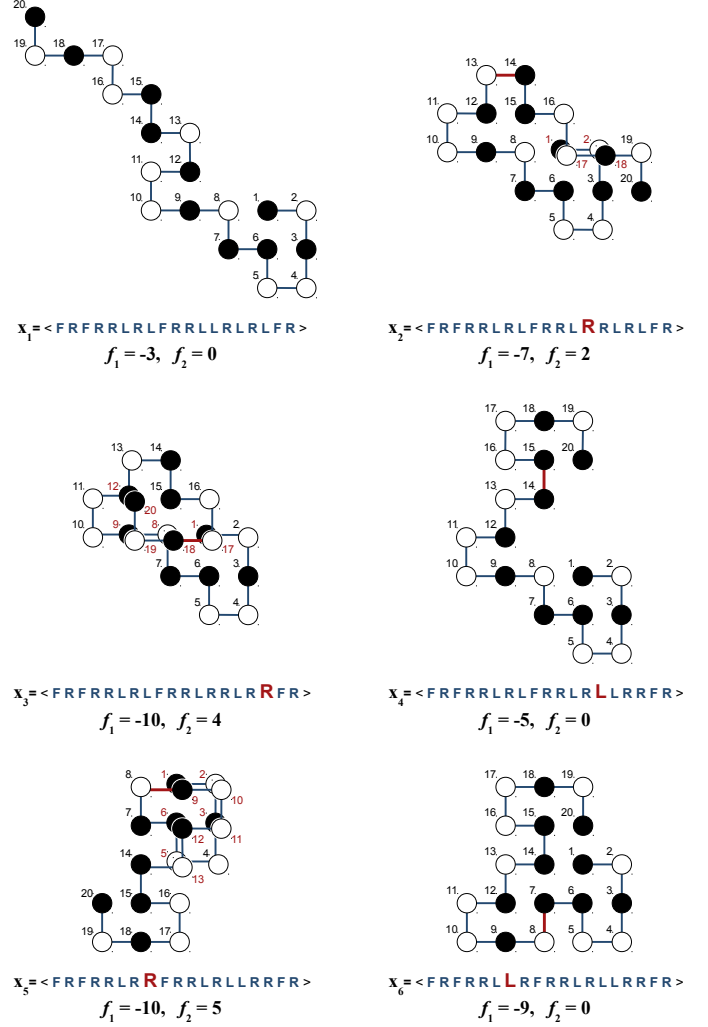


Figure 9: Neutral walk, based on the MO formulation, connecting six different solutions for instance 2d4. The encoding (based on relative moves) and objective function values are provided for each of the six solutions.

connected to NN_{i+1} , $1 \leq i < k$, $k \geq 3$, and at least NN_2 and NN_{k-1} are infeasible. In general, a minimum number of $m - 1$ infeasible NNs need to be traversed in order to connect m feasible NNs. To support these ideas, an example is provided in Figure 9. This figure illustrates a series of neutral moves between neighboring solutions (*i.e.*, a neutral walk), based on the MO formulation, from a feasible solution \mathbf{x}_1 with $Fitness(\mathbf{x}_1) = 3$, to another feasible solution \mathbf{x}_6 with $Fitness(\mathbf{x}_6) = 9$ (the global optimum for instance 2d4).¹² In this example, the feasible $NN(\mathbf{x}_1)$ and $NN(\mathbf{x}_6)$ networks have been connected by establishing intermediate neutral connections to (and between) four other different NNs; namely, the infeasible networks $NN(\mathbf{x}_2)$, $NN(\mathbf{x}_3)$ and $NN(\mathbf{x}_5)$, and the feasible network $NN(\mathbf{x}_4)$. The six solutions (\mathbf{x}_1 to \mathbf{x}_6), and all solutions in their respective NNs, become members of the same NN under the MO evaluation scheme. Therefore, the observed increase in the size of the sampled NNs is not exclusively due to the addition of a signif-

¹²It is important to remember that both the f_1 and f_2 objective functions are to be minimized, and that $f_1(\mathbf{x}) = E(\mathbf{x}) = -Fitness(\mathbf{x})$.

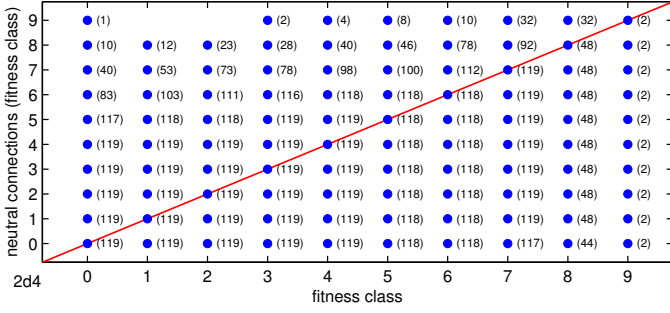


Figure 10: Neutral connections formed between fitness classes, 2d4.

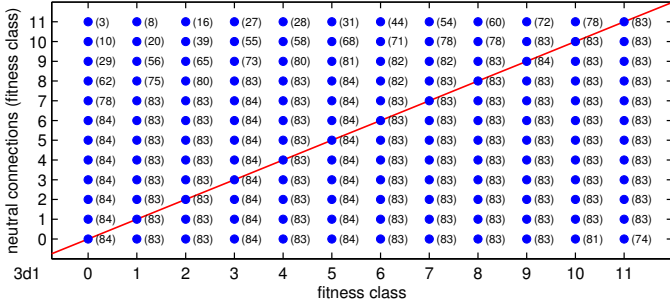


Figure 11: Neutral connections formed between fitness classes, 3d1.

icant number of infeasible nodes, as analyzed in Section 5.3.2, but is also a result of the combination with other feasible NNs. It should thus be noted that, as in the example, NNs resulting from the use of the MO formulation may involve solutions at different fitness classes (in the original problem formulation) and varying degrees of infeasibility.

To elaborate further on this matter, this section analyzes how the use of the MO formulation during the performed sampling produced neutral connections between NNs from distinct fitness classes. Figures 10 and 11 present the obtained results for instances 2d4 and 3d1, respectively. For each fitness class c , these figures indicate whether and how many of the NNs computed for solutions at this fitness class formed neutral connections to NNs at each other possible fitness class c' . Through these figures it is then possible to gain an insight into the diversity of fitness classes that a NN, computed based on the MO formulation, may involve. It is important to clarify that only connections to feasible NNs have been accounted for in this analysis. Diagonals in Figures 10 and 11 are used only as a reference (*i.e.*, all NNs connect to themselves at their corresponding fitness classes) to illustrate the single-objective case, so that all other connections not appearing along the diagonal are due to the landscape transformation. It can be seen from the figures that neutral connections were created between almost each possible pair of fitness classes. As the only exceptions, no connections to fitness class 9 were identified when sampling the NNs for fitness classes 1 and 2 of instance 2d4, as shown in Figure 10. More connections appear indicated below (rather than above) the diagonals in Figures 10 and 11. Indeed, the vast majority of the sampled NNs for the different fitness classes formed neutral connections to NNs at all other lower fitness

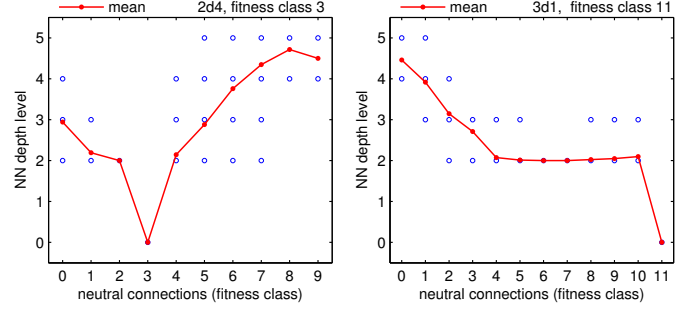


Figure 12: Neutral connections generated in relation to the depth levels reached during the NNs computation. Fitness class 3 of instance 2d4 (left) and fitness class 11 of instance 3d1 (right). Connections to the respective fitness classes (3 and 11) occurred at depth level 0 (starting point of the NN exploration).

classes, as it can be appreciated from the plots. This highlights the fact that inferior fitness classes are easier to reach than the superior ones (because of the funnel-like search landscape of the studied problem [83]). Despite the use of a considerably low value for parameter *maxDepth* during the NN sampling procedure, namely *maxDepth* = 5, such a reduced number of allowed neutral steps was still enough to establish connections between even the most distant fitness classes. That is, Figure 10 reveals that 1 out of the 119 explored NNs at fitness class 0, the worst fitness class, reached the optimum solution for instance 2d4 at fitness class 9. Note also that the 2 computed NNs from fitness class 9 connected to fitness class 0. A similar behavior can be observed with regard to instance 3d1. Figure 11 shows that 3 out of the 84 NNs from fitness class 0 merged with NNs at the best fitness class, *i.e.*, 11, and that a total of 74 connections between these fitness classes occurred in the opposite direction. All such connections were achieved after a maximum number of *maxDepth* successive neutral moves from the solution given as the starting point for the NNs computation.

Although neutral connections were established between NNs at distant fitness classes, as discussed above, it is possible to see from Figures 10 and 11 that the number of neutral connections tends to decrease as the distance between fitness classes increases (higher numbers of neutral connections are shown close to the diagonal). Therefore, the distance in fitness relates to the likelihood that two NNs can connect. This is particularly true when the number of allowed intermediate neutral connections is bounded (as done in this study with the use of parameter *maxDepth*). This point can be better illustrated in Figure 12. Taking as examples the fitness class 3 for instance 2d4 and fitness class 11 for instance 3d1, this figure indicates how the neutral connections to the different fitness classes arose as each allowed depth level was reached during the NNs computation.¹³ As shown in Figure 12, connections to feasible NNs in a range of fitness classes occurred promptly at depth level 2, *i.e.*, by traversing a single infeasible state (as mentioned before, two feasible NNs cannot be directly connected). It is possible to note, however, that a larger number of intermediate connections were usually required to reach the most distant fitness classes.

¹³Similar results to those presented in Figure 12 were obtained for the remaining fitness classes of the considered test instances.

The formation of neutral connections between two feasible networks NN_1 and NN_2 can be understood as the definition of (previously nonexistent) neutral paths bridging the corresponding regions of the feasible space. By allowing movement across infeasible areas, any solution \mathbf{x}_1 from NN_1 could potentially be reached through a neutral walk departing at an arbitrary solution \mathbf{x}_2 belonging to NN_2 . On the one hand, this can be particularly relevant when dealing with problems which present multiple disconnected feasible regions. On the other hand, even in connected feasible spaces, the length of the shortest path between two feasible solutions can be significantly greater if this path considers only feasible intermediate folding states [5]. In the example provided in Figure 9, the feasible solutions \mathbf{x}_1 and \mathbf{x}_6 differ exactly in $d = 5$ encoding positions. By exhaustive enumeration, it was found that all the $d! = 120$ possible shortest paths (of length d) connecting \mathbf{x}_1 and \mathbf{x}_6 involve infeasible solutions.¹⁴ Therefore, the shortest feasible path between this pair of solutions is necessarily longer (of length greater than d). It is worth mentioning that 11 out of the 120 shortest paths between \mathbf{x}_1 and \mathbf{x}_6 represent neutral paths under the studied MO problem formulation (one of them illustrated in Figure 9). In a related analysis reported in [6], Duarte-Flores and Smith computed all possible shortest paths from a set of near-optimal solutions to the global optimum of a particular HP model’s instance on the triangular lattice. As a result, only about 12% (on average) of the explored paths were found to be strictly feasible. The fact that most of the shortest paths between feasible regions traverse infeasible areas emphasizes the advantages of using the MO constraint-handling strategy. Furthermore, a path within the boundaries of the feasible space may require the explicit movement towards inferior fitness classes, especially when connecting different *basins of attraction*.¹⁵ The alternative fitness landscape induced by the MO strategy may potentially define neutral paths between distinct basins of attraction, which can be exploited as a means of escaping from local optima.

6. Introducing a search bias

By defining trade-offs between the quality and feasibility of solution candidates, the multi-objective (MO) approach to handle constraints allows for the exploitation of useful information from infeasible areas of the fitness landscape. Despite the potential advantages of the MO strategy in terms of the landscape transformation, as analyzed in Section 5.3, its lack of a proper search bias may also lead to detrimental effects on the ability of search algorithms for locating promising regions of the feasible space. That is, if a bias towards the feasible region is not introduced, a significant fraction of the computational effort can be invested in evaluating infeasible solutions. Depending on the particular problem characteristics, an unbiased search based on the MO method could even fail to reach a feasible solution [14].

¹⁴More specifically, 24 out of these 120 shortest paths involve 2 infeasible solutions, other 48 paths include 3 infeasible solutions, and all the 4 intermediate points are infeasible for the remaining 48 paths.

¹⁵The basin of attraction of a local optimum \mathbf{x} , involves the areas of the landscape which lead (or tend to lead) directly to \mathbf{x} .

In this section, the importance of coupling the studied MO constraint-handling strategy to an effective biasing mechanism is investigated. Three different biasing methods for the MO strategy are to be evaluated in terms of how their implementation impacts on the performance of search algorithms. A basic single-solution-based evolutionary algorithm (EA), called the (1+1) EA, and a basic genetic algorithm (GA), a population-based technique, have been considered.¹⁶ Details of these algorithms and the adopted parameter settings are provided in Appendix B.1 and Appendix B.3. The used HP model’s test instances, the performance measures and the experimental platform are described in Appendix A. The results obtained during the analysis of the three different biasing approaches are separately presented in Sections 6.1, 6.2 and 6.3.

6.1. Archiving

In evolutionary multi-objective optimization, maintaining a repository with the current approximation of the Pareto-optimal set, and thus of the Pareto front, is usually assumed to be a crucial issue [85, 86]. Hereafter, this kind of nondominated solutions repository is to be called *archive*, and the way in which this archive is constructed, updated and utilized during the search process will be referred to as the *archiving strategy*. This section analyzes the extent to which an archiving strategy can influence the behavior of the implemented (1+1) EA when using the MO constraint-handling technique.

Rather than functioning as a source of genetic material, *i.e.*, as a population, in the archiving (1+1) EA the archive is used only with the aim of introducing a bias in the selection process. In order to be accepted, a new candidate individual must represent a competitive trade-off between the two defined optimization objectives. This is determined by comparing it with respect to the whole Pareto front approximation stored in the archive. In this way, although (strictly speaking) an explicit bias towards the feasible region is not being applied, archiving restricts the movement of the algorithm, allowing it to concentrate on promising regions, either feasible or infeasible, of the fitness landscape. The archiving variant of the (1+1) EA is described in detail in Appendix B.2.

Figures 13 and 14 contrast the performance of the basic and archiving variants of the (1+1) EA, using the MO strategy, for all two- and three-dimensional test instances. Results are reported in terms of the relative root mean square error, RMSE, computed over a total of 31 independent executions of each experiment (the RMSE measure is defined in Appendix A.1). In all the cases, the two algorithms were run for a maximum number of 10^6 solution evaluations. As it can be seen from the plots, the use of the archiving strategy within the (1+1) EA has led to a significant improvement in the RMSE for all the 30 adopted test cases (lower RMSE is preferred). It is also possible to observe that the benefits of archiving tend to become more evident as the problem size (length of the protein sequence) increases.

¹⁶Note that some of the studied biasing methods are only suitable, and thus are only analyzed here, either for the (1+1) EA or for the GA.

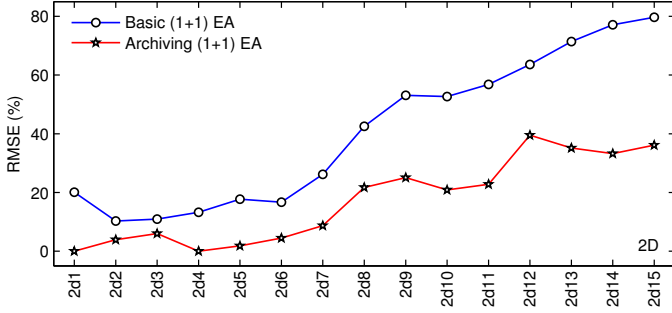


Figure 13: RMSE obtained by the basic and archiving variants of the (1+1) EA when using the MO strategy. Two-dimensional instances.

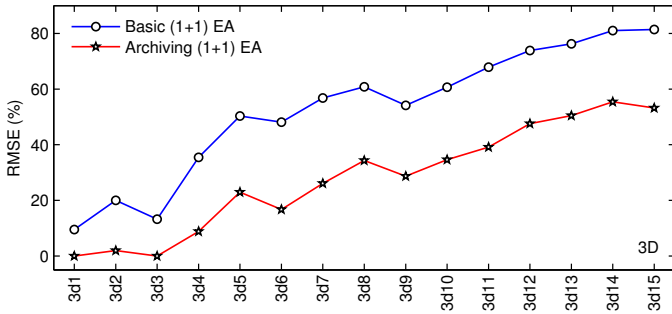


Figure 14: RMSE obtained by the basic and archiving variants of the (1+1) EA when using the MO strategy. Three-dimensional instances.

From the above results, archiving was found to be essential for guiding the search effectively when the MO constraint-handling strategy is implemented. In the words of Handl *et al.* [51]: “*this way of using an archive yields a negative efficiency preserving strategy, i.e., it prevents degradation of solutions. We say there is degradation if the current solution is replaced at some later iteration by one that it dominates. Such degradation prevents convergence and can lead to endless cycling between solutions that are not mutually incomparable*”. Without archiving, therefore, the (1+1) EA based on the MO strategy may drift through the search landscape, moving away from or moving towards the feasible region in a bias-free manner.

Finally, it should be noted that archiving strategies can also be used in the context of population-based methods (where archives are usually referred to as secondary populations). The considered GA, however, relies on an elitist selection scheme which inherently preserves in the population the current Pareto front approximation (or at least part of it due to the fixed population size). Thus, the biasing effects obtained through archiving are implicitly incorporated in this algorithm.

6.2. Feasibility rules

One of the simplest, yet effective and widely used constraint-handling methods, consists in defining a set of rules on which the discrimination among individuals is to be based. This approach is commonly referred to as the use of *feasibility rules* in the specialized literature [11, 87–90]. The popularity of this method stems not only from its parameter-free nature, but also from its ability to be combined with other constraint-handling

mechanisms, as reviewed in [11]. One of the most representative works on this topic was reported by Deb [91], where it was proposed a GA implementing a binary tournament selection operator which relies on the following three criteria:

1. If comparing between two feasible solutions, the one with the best objective function value is to be preferred.
2. If comparing between two infeasible solutions, the one with the lowest infeasibility degree is to be preferred.
3. If comparing between a feasible and an infeasible solution, the feasible one is to be preferred.

The first criterion can be generalized to the case where two solutions presenting the same degree of constraint violation are considered. This more general case involves the comparison between feasible individuals, as in the original rule, but covers also the case where two infeasible individuals with the same infeasibility degree are being compared. Such a later scenario has not been accounted for in the originally proposed set of rules [92]. This extended version of the first criterion is implicitly satisfied when handling constraints by multi-objective optimization. The multi-objective (MO) approach, however, lacks the bias towards the feasible region that the second and third discrimination criteria represent.

This section explores how the use of simple feasibility rules based on the MO method can help in guiding the search process effectively in the implemented GA.¹⁷ More specifically, in the studied approach the preference relation between two solutions $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ will depend on the following criteria:

1. If $\mathbf{x}_1 < \mathbf{x}_2$, then solution \mathbf{x}_1 is to be preferred.
2. If $\mathbf{x}_2 < \mathbf{x}_1$, then solution \mathbf{x}_2 is to be preferred.
3. Otherwise, i.e., $\mathbf{x}_1 \not< \mathbf{x}_2$ and $\mathbf{x}_2 \not< \mathbf{x}_1$, the solution with the lowest degree of constraint violation (lowest f_2 value) is to be preferred.

That is, individuals will be first compared based on the Pareto-dominance relation. Whenever no preferences can be imposed by using the Pareto-dominance relation, the degree of infeasibility of the solutions will be adopted as a secondary discrimination criterion. The implementation of these rules required the adaptation of both the *selection-for-variation* and *selection-for-survival* processes of the GA. On the one hand, in *selection-for-variation* the binary tournament selection operator was simply equipped with the new defined set of rules. On the other hand, the *selection-for-survival* process is performed by means of *nondominated sorting* (NDS), as in the *nondominated sorting genetic algorithm*, NSGA-II [93]. As treated more extensively in Appendix B.3, NDS works by defining layers of nondominated individuals, which are then iteratively included (from the best down to the worst layer) until completing the new GA population. If this iterative selection

¹⁷Implementing this approach within the (1+1) EA results in an overpenalization scenario; once a feasible solution is reached, infeasible solutions will never be considered. Thus, the conducted analysis focuses only on the GA.

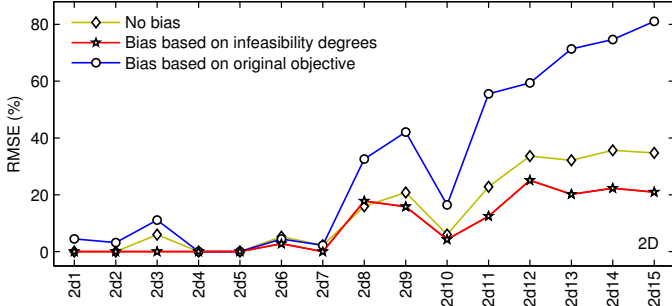


Figure 15: Introducing a search bias in the GA by using feasibility rules. RMSE obtained for all the two-dimensional test instances.

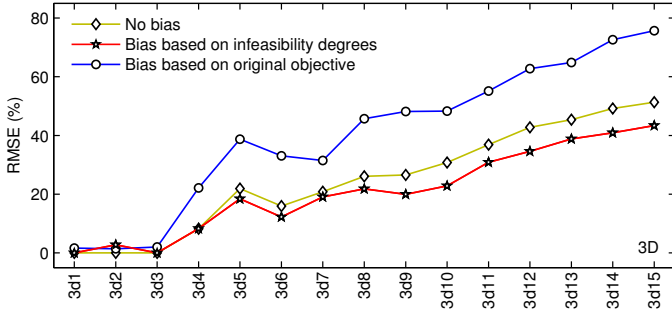


Figure 16: Introducing a search bias in the GA by using feasibility rules. RMSE obtained for all the three-dimensional test instances.

procedure faces a nondominated layer containing more individuals than the number of free slots in the population, the secondary criterion based on infeasibility degrees is applied in order to choose the remaining survivors.¹⁸ Since only the last considered nondominated layer is discriminated based on such an infeasibility-based criterion, a significant portion of the infeasible individuals could potentially be selected.¹⁹ This reduces the selection pressure and, thus, can contribute in overcoming *premature convergence*, a problem usually related to the use of the feasibility rules approach for handling constraints [11].

The performance of the GA using the MO constraint-handling strategy was evaluated with and without incorporating the above described infeasibility-based secondary criterion. In addition, a third variant of the GA was considered where such a secondary discrimination criterion is based on the original objective function, and not on the infeasibility degrees. In this way, it will be possible to analyze not only the importance of having a search bias, but also the effects that can be achieved if this bias favors either one or the other of the two optimization objectives defined by the MO strategy. Figures 15 and 16 present the obtained results, in terms of the RMSE measure, for all two- and three-dimensional test cases.²⁰ In all the experiments, a maximum number of 10^6 evaluations was used

as the stopping condition and 31 repetitions were performed. The introduction of a search bias towards the feasible region allowed the GA to score the best RMSE values in most of the cases. Although no important differences can be appreciated for the smallest problem instances, the advantages of introducing this bias are more clear when focusing on the hardest ones (rightmost part of the figures). It is interesting to observe from the plots that, rather than benefiting optimization, biasing the search according to the original objective has impacted negatively on performance. Realize that in a set of nondominated solutions, computed based on the MO strategy, there can be at most one feasible solution; all other solutions within the set are infeasible and, by definition, strictly better than the feasible member with regard to the original objective. Therefore, the discrimination of nondominated individuals based solely on the original objective will favor those individuals which, despite showing a prominent behavior for this criterion, represent the poorest trade-offs in terms of infeasibility. Consequently, the search process can be guided away from the feasible space.

Finally, it is important to evaluate the studied MO-based set of feasibility rules with respect to the original set of feasibility rules reported by Deb [91] (described at the beginning of this section). Figures 17 and 18 compare the online (throughout the search) performance scored by the GA when using the two different approaches. Results are given in terms of the overall relative root mean square error, O-RMSE (see Appendix A.1) and are presented in steps of 50,000 solution evaluations. According to Figures 17 and 18, the original Deb's feasibility rules allowed the GA to achieve lower O-RMSE values during the first stages of the search process. This is due to the high selection pressure associated with this method. Despite enabling a faster convergence, a high selection pressure tends to lead search algorithms to converge prematurely to suboptimal solutions. As commented before, this has been one of the main criticisms with regard to the use of this constraint-handling strategy [11]. Implementing the MO-based feasibility rules within the framework of the NDS procedure (*selection-for-survival* stage) allows infeasible solutions to effectively compete against feasible ones for a place in the new GA's population. This prevents the excessive selection bias and, as observed from the plots, enables a better convergence at the end of the optimization process.

6.3. Proportional bias

In multi-objective optimization, introducing a bias can be understood as the articulation of preferences to capture the relative importance of the different optimization criteria. Consider a two-objective problem, denoted by the objective vector $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})]^T$. If f_2 was determined to be a more important objective function than f_1 , it can be hypothesized that the addition of *noise* to f_1 , a noise which is proportional and directly dependent on f_2 , would produce a biasing effect in order to favor f_2 during the search process. That is, the incorporation of noise into f_1 relaxes the selection pressure with regard to this objective (the real contribution of f_1 to guiding the search is reduced). Moreover, by relating the injected noise with function f_2 , the selection pressure with respect to f_2 is strengthened, yielding a bias. From this reasoning, the above multi-objective

¹⁸NSGA-II uses the so-called *crowding distance* as the secondary criterion.

¹⁹The number of selected feasible individuals will always match the number of considered nondominated layers. This is because there can be at most one feasible solution per nondominated set computed based on the MO strategy.

²⁰For each of the instances, the results presented in Figures 15 and 16 correspond to the lowest RMSE values obtained by evaluating a set of different parameter configurations of the GA, refer to Section 7.2.1 for details.

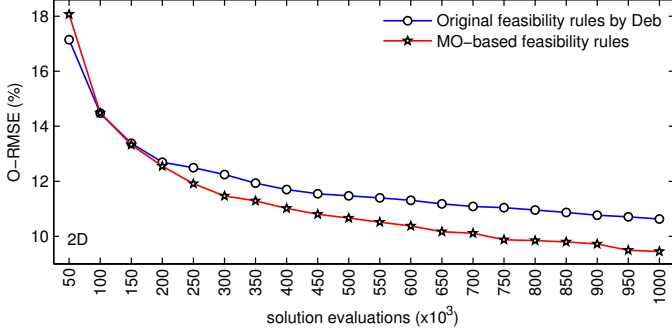


Figure 17: Comparing the feasibility rules based on the multi-objective (MO) strategy with regard to those originally proposed by Deb [91]. Online performance (O-RMSE) of the GA when solving the two-dimensional instances.

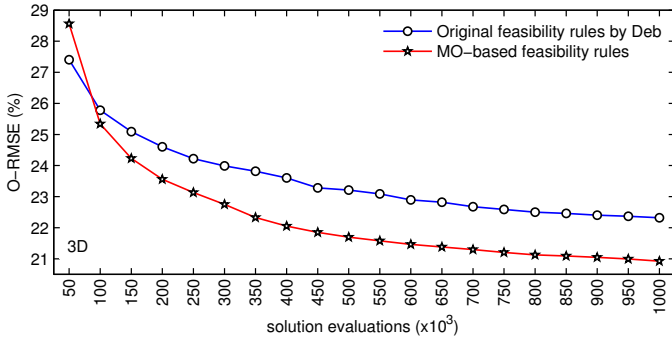


Figure 18: Comparing the feasibility rules based on the multi-objective (MO) strategy with regard to those originally proposed by Deb [91]. Online performance (O-RMSE) of the GA when solving the three-dimensional instances.

problem, where f_2 is assumed to be the highest priority objective, can alternatively be stated in terms of the objective vector $\mathbf{f}'(\mathbf{x}) = [f'_1(\mathbf{x}), f'_2(\mathbf{x})]^T$, such that

$$f'_1(\mathbf{x}) = f_1(\mathbf{x}) + \omega \left(\frac{f_2(\mathbf{x})}{f_2^{\max}} \right) (f_1^{\max} - f_1^{\min}), \quad (9)$$

$$f'_2(\mathbf{x}) = f_2(\mathbf{x}), \quad (10)$$

where f_1^{\max} and f_1^{\min} are respectively the maximum and minimum known values for function f_1 (since the beginning of the search process), and f_2^{\max} is the maximum known value for function f_2 .²¹ In (9), factor $(f_1^{\max} - f_1^{\min})$ represents the maximum known difference between f_1 values, so that by using this value it would be possible to alter the preference relation between any pair of solutions with regard to f_1 . Factor $\left(\frac{f_2(\mathbf{x})}{f_2^{\max}} \right)$ allows the incorporated noise to be proportional to the scored f_2 performance; *i.e.*, this factor tends to 1 as worse f_2 values are considered (minimization assumed). Therefore, the better the solution in objective f_2 , the lower the perturbation to its f_1 objective value. Finally, the *bias strength* ω is a user defined parameter introduced with the aim of evaluating the impact of

further controlling the magnitude of the applied noise. Using this strategy, two different solutions which are incomparable (mutually nondominated) with respect to their original \mathbf{f} objective vectors, could be discriminated (in favor of the best f_2 performance) if compared with respect to their alternative objective vectors \mathbf{f}' . This strategy can thus be implemented within a search algorithm in order to set a search bias.

This section tests the ability of the above described strategy to provide the multi-objective (MO) constraint-handling approach with an effective bias. The objective function f_2 in the MO problem formulation, which accounts for the degree of constraint violation, is to be defined as the most important criterion in order to bias the search towards the feasible region. It should be emphasized that, under the MO formulation, the use of the proportional biasing mechanism affects only infeasible individuals; *i.e.*, $f_2(\mathbf{x}) = 0$ for all feasible individuals $\mathbf{x} \in \mathcal{X}_{\mathcal{F}}$, so that no noise is added to their f_1 values. The analysis here presented focuses on the two implemented algorithms: the (1+1) EA and the GA. In these algorithms, the alternative objective vectors \mathbf{f}' of all individuals are to be computed at each iteration to serve as the basis for driving selection. A set of values in the range $[0, 2]$ have been explored for the bias strength parameter ω , where $\omega = 0$ indicates that no bias is to be applied. In all the cases, the algorithms were run for a total of 10^6 solution evaluations, 31 independent executions were performed, and results are evaluated in terms of the O-RMSE measure.

The results for the (1+1) EA are shown in Figures 19 and 20. In addition to the basic (1+1) EA, the evaluation of the proportional biasing mechanism covers also the archiving variant of the (1+1) EA, as described and analyzed in Section 6.1. Hence, the archiving (1+1) EA studied in this section integrates two different biasing methods (*i.e.*, archiving and the proportional bias). Figures 19 and 20 confirm the need for an effective biasing strategy when the handling of constraints is approached by multi-objective optimization. Without a bias ($\omega = 0$), the basic (1+1) EA scored considerably high O-RMSE values for both the two- and the three-dimensional test instances. Note, however, that the performance of this algorithm was gradually improved with the increasing value of ω . The best performance for the basic (1+1) EA was reached at $\omega = 1.6$ for the two-dimensional instances, and $\omega = 1.7$ for three-dimensional case, where the O-RMSE measure was decreased by more than 23% in both cases with respect to the corresponding results at $\omega = 0$. Due to the implicit bias that the archiving (1+1) EA involves, the rewards of implementing the proportional biasing strategy were not as remarkable as those for the basic version of this algorithm. Nevertheless, most of the explored ω values allowed the archiving (1+1) EA to achieve slight but still appreciable decreases in the O-RMSE measure. While the basic (1+1) EA performed the best for high ω values ($\omega > 1.5$), a less strength of the proportional bias was required when using the self-biasing archiving (1+1) EA (whose performance deteriorated for the highest ω values). The archiving (1+1) EA showed its best performance when using a value of $\omega = 1.3$ (two-dimensional case), and $\omega = 0.9$ (three-dimensional case).

²¹ Alternatively, f_1^{\max} , f_1^{\min} and f_2^{\max} could be computed from the current population or Pareto front approximation. These values could even be fixed if this problem-dependent information is known a priori.

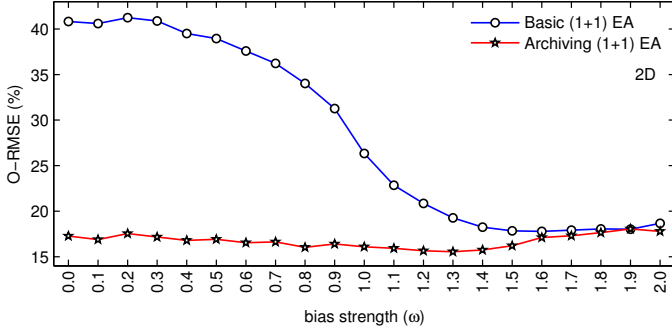


Figure 19: Introducing a proportional bias in the MO strategy. O-RMSE obtained by the basic and archiving (1+1) EA for the two-dimensional instances.

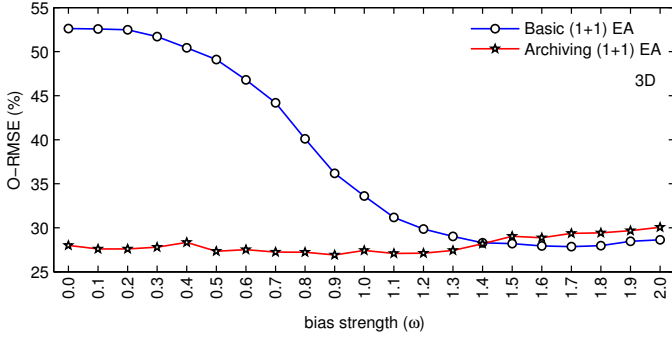


Figure 20: Introducing a proportional bias in the MO strategy. O-RMSE obtained by the basic and archiving (1+1) EA for the three-dimensional instances.

Finally, Figures 21 and 22 present the obtained results with regard to the implemented GA.²² As it can be seen from the plots, the incorporation of the proportional biasing mechanism (by using $\omega > 0$) has led to a noticeable enhancement in the performance of the GA. The behavior of the GA exhibits a clear tendency to improve with the increase in the bias strength parameter ω . It is interesting to note, however, that once the best O-RMSE values were reached at $\omega = 1.6$ and $\omega = 1.5$ (for the two- and three-dimensional cases, respectively), this tendency changes and the GA's performance begins to decline for higher ω values. From this, and given that the above analyzed (1+1) EA suffered a performance decrease when using the highest ω values as well, it is possible to say that the excessive bias could also be detrimental to the search efficiency. In this particular context, the increase in ω tends to produce overpenalization. Therefore, defining the proper amount of search bias could be a non-trivial, problem-dependent and algorithm-dependent task.

7. Impact on search performance

This section investigates the suitability of the multi-objective optimization (MO) strategy for handling constraints in the HP model. To this end, the MO strategy is evaluated and compared with respect to two different constraint-handling approaches

²²For each ω value, the O-RMSE in Figures 21 and 22 corresponds to the best performance obtained by evaluating a set of different parameter configurations for the GA; details provided in Section 7.2.1.

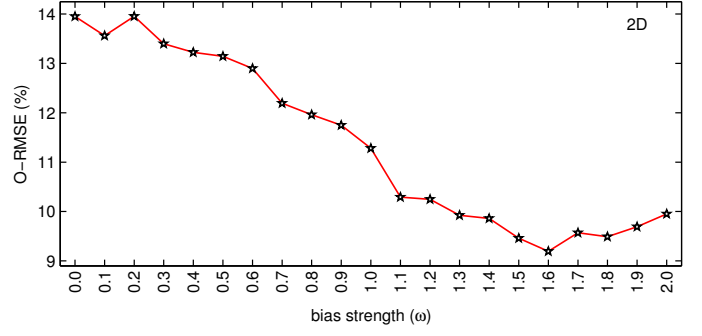


Figure 21: Introducing a proportional bias in the MO constraint-handling strategy. O-RMSE obtained by the GA for the two-dimensional test instances.

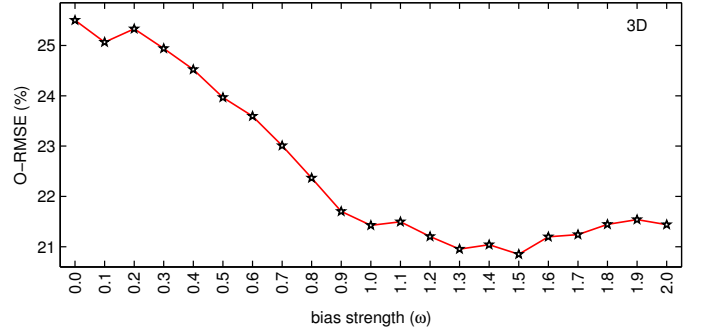


Figure 22: Introducing a proportional bias in the MO constraint-handling strategy. O-RMSE obtained by the GA for the three-dimensional test instances.

usually adopted in the specialized literature, namely, the rejection of infeasible protein conformations and the application of penalties. These approaches are to be referred to as the reject (RJ) and penalty function (PF) strategies and are described in detail in Appendix C. As discussed in Section 6, introducing a proper search bias is crucial for the success of the MO strategy. This issue is further addressed in this section by evaluating the different biasing mechanisms studied in Section 6 with respect to each other. The comparative analysis presented in this section focuses on the impact that the various studied constraint-handling methods have on the performance of search algorithms. Two different evolutionary algorithms (EAs) have been considered, a basic single-solution-based EA and a population-based EA. The corresponding analyses are covered in Sections 7.1 and 7.2.

7.1. Analysis for a single-solution-based algorithm

A basic single-solution-based evolutionary algorithm (EA), the so-called (1+1) EA, has been implemented in order to assess the impact of using the studied constraint-handling methods. Five different approaches are considered, the reject (RJ) and penalty function (PF) strategies taken as the baseline, and three variants of the multi-objective (MO) technique originated from the use of the biasing mechanisms analyzed in Section 6: (i) MO+AR, where archiving is used to bias the search process; (ii) MO+PB, where a proportional bias is introduced; and (iii) MO+AR+PB, which combines both the archiving and the proportional biasing mechanisms. In the (1+1) EA, the discrimi-

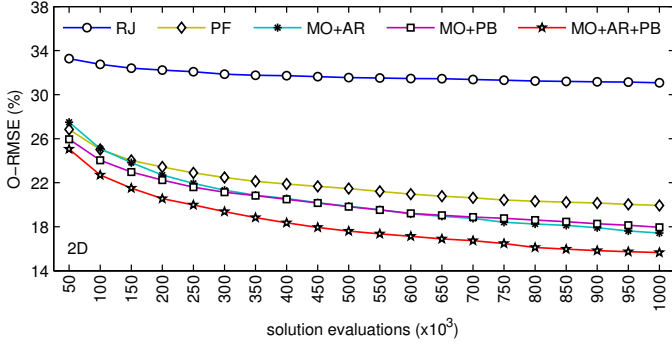


Figure 23: O-RMSE scored by the (1+1) EA as the search process progressed. Two-dimensional test instances.

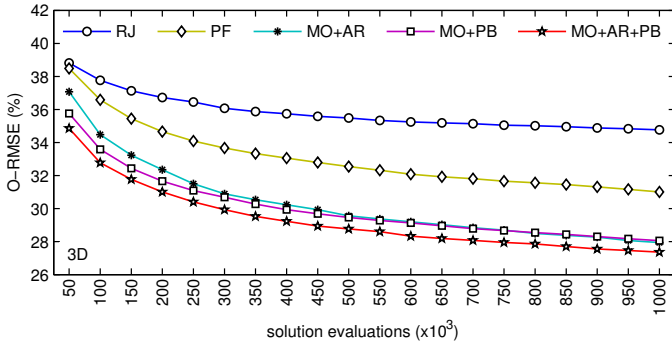


Figure 24: O-RMSE scored by the (1+1) EA as the search process progressed. Three-dimensional test instances.

nation among candidate individuals is adapted according to the constraint-handling method to be used. In this way, the search behavior, and thus the performance of this algorithm, will be determined by each of the different studied techniques.

The proportional biasing mechanism, which leads to the MO+PB and MO+AR+PB strategies, requires the adjustment of the bias strength parameter, ω . Similarly, the PF method requires the fine-tuning of the penalty weight, ρ . The analysis here conducted considers the best performing settings for these parameters, as they were respectively derived in Section 6.3 and Appendix C.2. All implementation details and settings for the (1+1) EA are provided in Appendix B.1. Also, the archiving variant of the (1+1) EA, on which the MO+AR and MO+AR+PB approaches rely, is described in Appendix B.2. Details on the test instances, the performance measures, the methodology for the statistical significance analysis, and the utilized experimental platform, are included in Appendix A.

Figures 23 and 24 show the online performance achieved by the (1+1) EA when using the studied constraint-handling approaches. Performance is expressed in terms of the overall relative root mean square error, O-RMSE, computed from 100 independent executions of each experiment.²³ Results are reported in steps of 50,000 solution evaluations until completing the

²³Notice that, while previous analyses considering different parameter settings for the compared approaches were based on 31 repetitions of the experiments, detailed analyses using the best performing settings are based on 100 repetitions in order to compare more representative performance samples.

maximum number of 10^6 evaluations defined as the stopping condition. From these figures, it is possible to see that the lowest O-RMSE values, in both the two- and the three-dimensional test cases, were reached by using MO+AR+PB. Therefore, the use of archiving together with the introduction of a proportional bias constitutes a more effective biasing strategy when compared to the separate use of these mechanisms. MO+PB presented a more accelerated convergence than MO+AR at the first stages of the search. This can be explained by the fact that, given that MO+AR does not explicitly bias the search towards the feasible region (as discussed in Section 6.1), this method invests more effort in exploring infeasible states. It is worth noting, however, that such an investment has paid off; the slope in the corresponding curve is more pronounced, indicating that MO+AR exhibits a greater tendency to improve. This allowed MO+AR to score the second best O-RMSE values at the end of the search process. Finally, PF provided a more competitive behavior for the (1+1) EA when compared to the use of RJ, which obtained the poorest overall performance.

To further compare the studied constraint-handling approaches, Tables 2 and 3 detail the results for all two- and three-dimensional test instances at the end of the search process (after 10^6 solution evaluations). The results for each of the instances are given in terms of the best obtained energy value (E_b), the number of performed executions where this solution quality was reached (ν), and the arithmetic mean (\bar{E}). In addition, the O-RMSE measure is provided at the bottom of the tables. The lowest average energy for each of the instances, as well as the best O-RMSE values, appears **shaded**. As it can be seen from the tables, the use of the three multi-objective strategies improved the average performance of the algorithm, in the vast majority of the cases, with respect to the RJ and PF methods. An interesting behavior can be observed with regard to the MO+AR and MO+PB approaches. While MO+AR tends to perform better than MO+PB for the shortest test sequences, MO+PB scored more competitive results for the largest ones. This suggests that, by not explicitly introducing a search bias, MO+AR yields a broader exploration. Nevertheless, an explicit and more effective bias seems to be required if the hardness of the problem instances increases. Note, however, that MO+AR was found when analyzing Figures 23 and 24 to present a greater tendency to improve as the search process progresses. The MO+AR strategy could thus be expected to meet or even exceed the results of MO+PB for the largest test cases if the algorithm is allowed to run for a higher number of solution evaluations. The best overall performance was exposed by the MO+AR+PB method. By combining the advantages of the two different biasing mechanisms, MO+AR+PB decreased the O-RMSE measure by 15.43% and 4.28% with respect to RJ and PF in the two-dimensional instances, respectively, and by 7.4% and 3.64% in the three-dimensional case.

Finally, Table 4 complements the information provided in Tables 2 and 3 in order to highlight whether the performance differences between the studied approaches were statistically significant or not. Each row in this table compares two strategies, say A and B, which is denoted as “A / B”. If a significant performance difference exists between A and B for a par-

Table 2: Results obtained by the (1+1) EA when using the studied constraint-handling strategies. Two-dimensional test cases.

Seq.	ℓ	E^*	RJ		PF		MO+AR		MO+PB		MO+AR+PB	
			E_b (ν)	\bar{E}	E_b (ν)	\bar{E}	E_b (ν)	\bar{E}	E_b (ν)	\bar{E}	E_b (ν)	\bar{E}
2d1	18	-4	-4 (6)	-2.62	-4 (54)	-3.54	-4 (100)	-4.00	-4 (86)	-3.86	-4 (100)	-4.00
2d2	18	-8	-7 (65)	-6.62	-8 (54)	-7.54	-8 (84)	-7.84	-8 (83)	-7.83	-8 (86)	-7.86
2d3	18	-9	-8 (50)	-7.46	-9 (43)	-8.42	-9 (70)	-8.70	-9 (18)	-8.18	-9 (86)	-8.86
2d4	20	-9	-8 (16)	-6.67	-9 (64)	-8.60	-9 (98)	-8.97	-9 (33)	-8.29	-9 (95)	-8.95
2d5	20	-10	-8 (32)	-7.24	-10 (39)	-8.99	-10 (86)	-9.86	-10 (44)	-9.10	-10 (58)	-9.55
2d6	24	-9	-9 (1)	-7.03	-9 (46)	-8.45	-9 (83)	-8.83	-9 (33)	-8.30	-9 (69)	-8.69
2d7	25	-8	-8 (1)	-5.68	-8 (15)	-7.01	-8 (43)	-7.41	-8 (24)	-7.19	-8 (50)	-7.49
2d8	36	-14	-12 (7)	-9.82	-13 (10)	-11.28	-14 (1)	-11.36	-13 (8)	-11.34	-14 (2)	-11.58
2d9	48	-23	-19 (2)	-14.88	-20 (2)	-16.79	-23 (1)	-17.69	-20 (2)	-17.41	-21 (2)	-17.83
2d10	50	-21	-19 (1)	-14.78	-21 (1)	-16.49	-20 (6)	-17.04	-21 (1)	-17.46	-21 (2)	-17.77
2d11	60	-36	-32 (1)	-26.12	-32 (3)	-28.20	-34 (1)	-27.81	-32 (2)	-28.75	-33 (2)	-28.81
2d12	64	-42	-31 (1)	-24.00	-31 (7)	-26.04	-33 (1)	-26.25	-32 (3)	-28.13	-33 (2)	-26.68
2d13	85	-53	-41 (1)	-34.50	-44 (1)	-37.75	-45 (1)	-35.54	-45 (1)	-39.03	-46 (1)	-38.78
2d14	100	-48	-38 (1)	-29.69	-41 (2)	-34.55	-41 (1)	-32.90	-40 (1)	-34.74	-42 (1)	-34.78
2d15	100	-50	-40 (1)	-31.30	-40 (1)	-34.70	-40 (1)	-32.60	-41 (2)	-35.73	-42 (3)	-36.23
O-RMSE			31.08%		19.93%		17.43%		17.95%		15.65%	

Table 3: Results obtained by the (1+1) EA when using the studied constraint-handling strategies. Three-dimensional test cases.

Seq.	ℓ	E^*	RJ		PF		MO+AR		MO+PB		MO+AR+PB	
			E_b (ν)	\bar{E}	E_b (ν)	\bar{E}	E_b (ν)	\bar{E}	E_b (ν)	\bar{E}	E_b (ν)	\bar{E}
3d1	20	-11	-11 (58)	-10.43	-11 (93)	-10.92	-11 (100)	-11.00	-11 (95)	-10.95	-11 (99)	-10.99
3d2	24	-13	-13 (13)	-11.07	-13 (54)	-12.37	-13 (94)	-12.94	-13 (43)	-12.16	-13 (77)	-12.74
3d3	25	-9	-9 (58)	-8.42	-9 (97)	-8.97	-9 (100)	-9.00	-9 (98)	-8.98	-9 (100)	-9.00
3d4	36	-18	-18 (14)	-15.24	-18 (25)	-16.08	-18 (46)	-16.97	-18 (23)	-16.37	-18 (57)	-17.27
3d5	46	-35	-28 (2)	-24.06	-30 (3)	-25.33	-32 (1)	-27.30	-30 (1)	-25.80	-31 (2)	-26.89
3d6	48	-31	-28 (2)	-22.70	-29 (1)	-24.00	-30 (1)	-26.04	-30 (1)	-24.71	-29 (7)	-25.49
3d7	50	-34	-27 (1)	-21.15	-27 (3)	-22.19	-30 (1)	-24.68	-28 (2)	-23.17	-28 (2)	-23.87
3d8	58	-44	-33 (3)	-26.74	-34 (3)	-28.75	-37 (1)	-29.76	-34 (7)	-29.97	-37 (1)	-30.50
3d9	60	-55	-46 (2)	-38.30	-47 (3)	-40.67	-48 (2)	-40.00	-48 (1)	-41.13	-49 (2)	-42.01
3d10	64	-59	-45 (2)	-34.88	-47 (1)	-36.44	-48 (1)	-38.86	-50 (1)	-39.01	-50 (1)	-38.78
3d11	67	-56	-38 (3)	-30.62	-41 (1)	-32.22	-40 (2)	-33.30	-41 (1)	-33.60	-42 (1)	-33.97
3d12	88	-72	-46 (1)	-36.44	-49 (1)	-37.15	-47 (5)	-38.96	-48 (1)	-40.15	-51 (1)	-40.14
3d13	103	-58	-39 (1)	-29.25	-38 (2)	-29.67	-39 (1)	-29.70	-39 (1)	-33.10	-40 (1)	-31.01
3d14	124	-75	-45 (2)	-33.32	-46 (2)	-34.11	-46 (1)	-34.29	-52 (1)	-39.28	-50 (1)	-36.26
3d15	136	-83	-51 (1)	-37.66	-51 (1)	-37.94	-50 (1)	-38.11	-51 (1)	-43.41	-51 (2)	-40.42
O-RMSE			34.76%		31.00%		27.94%		28.05%		27.36%	

Table 4: Statistical analysis for comparing the performance of the (1+1) EA when using the analyzed constraint-handling approaches.

	Two-dimensional instances															Three-dimensional instances															Overall	
	2d1	2d2	2d3	2d4	2d5	2d6	2d7	2d8	2d9	2d10	2d11	2d12	2d13	2d14	2d15	3d1	3d2	3d3	3d4	3d5	3d6	3d7	3d8	3d9	3d10	3d11	3d12	3d13	3d14	3d15		
PF / RJ	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+					26+ 0-	
MO+AR / RJ	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+					27+ 0-
MO+PB / RJ	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+		30+ 0-
MO+AR+PB / RJ	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+		30+ 0-
MO+AR / PF	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+		19+ 3-
MO+PB / PF	+	+	-	-		-	+			+	+		+	+		+				+	+	+	+		+	+	+	+	+	+		17+ 3-
MO+AR+PB / PF	+	+	+	+	+	+	+	+	+	+	+			+		+			+	+	+	+	+	+	+	+	+	+	+	+		26+ 0-
MO+PB / MO+AR	-		-	-	-	-	-					+	+	+	+	+	-	-	-	-	-	-		+				+	+	+		9+ 12-
MO+AR+PB / MO+AR				+		-	-				+	+		+	+	+			+		-	-		+				+	+	+		11+ 5-
MO+AR+PB / MO+PB	+		+	+	+	+	+		+				-				+		+	+	+	+						-	-	-		12+ 4-

ticular instance, the corresponding cell is either marked $+$ or marked $-$ depending on whether such a difference was in favor of, or against A. Unmarked cells indicate that there was not a statistically important difference between A and B. The right-most column of the table summarizes the results of this analysis. As shown in Table 4, PF and MO+AR significantly outperformed RJ in 26 and 27 of the instances. Both MO+PB and MO+AR+PB achieved a statistically significant performance increase with regard to RJ for all the 30 adopted test sequences. The MO+AR, MO+PB and MO+AR+PB strategies scored significantly better results than PF in 19, 17 and 26 instances, respectively. Nevertheless, MO+AR and MO+PB were each significantly surpassed by PF in 3 of the two-dimensional test cases. By comparing among the multi-objective strategies, it is first possible to confirm that MO+PB was statistically superior to MO+AR in 9 of the largest (hardest) test cases, while significantly inferior to MO+AR for 12 of the smallest (easiest) ones. Finally, the table indicates that MO+AR+PB significantly improved the performance of the algorithm for 11 and 12 of the instances with respect to MO+AR and MO+PB, but there were still important differences favoring MO+AR and MO+PB respectively in 5 and 4 of the cases.

7.2. Analysis for a population-based algorithm

As the population-based method, the genetic algorithm (GA) described in detail in Appendix B.3 has been considered. The implementation of the different constraint-handling strategies influences the selection process, which is a major determinant of the GA's behavior. Hence, by evaluating the performance of the GA, it will be possible to inquire into the advantages of using the studied constraint-handling approaches. Four different strategies are to be analyzed: the reject (RJ) and penalty function (PF) methods adopted as reference, and the multi-objective approaches introducing a search bias by means of feasibility rules (MO+FR) and proportional biasing (MO+PB).

The four studied constraint-handling strategies are first evaluated in Section 7.2.1 under different parameter settings for the GA. The purpose of such an initial evaluation is to identify the most appropriate GA conditions for each of the approaches, to be adopted during the more detailed comparative analysis presented later in Section 7.2.2. The reader is referred to Appendix A for details on the considered test cases, performance measures, the methodology followed during the statistical significance analysis, and the experimental platform.

7.2.1. Settings for the genetic algorithm

In this section, the RJ, PF, MO+FR and MO+PB strategies are evaluated under different conditions of the implemented GA. Three recombination and mutation probabilities were considered: $p_c \in \{0.8, 0.9, 1.0\}$, $p_m \in \{\frac{1}{\ell-2}, \frac{2}{\ell-2}, \frac{3}{\ell-2}\}$. Thus, a total of 9 configurations of the GA are investigated. The population size was fixed to $N = 100$ in all the cases. The GA was allowed to run for a maximum number of 10^6 solution evaluations and a total of 31 repetitions for each experiment were performed. Figures 25 and 26 plot the O-RMSE measure scored by the four studied constraint-handling approaches when using

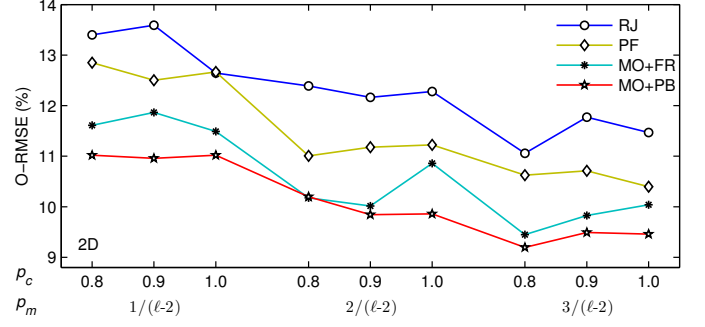


Figure 25: Evaluating different GA settings. Two-dimensional case.

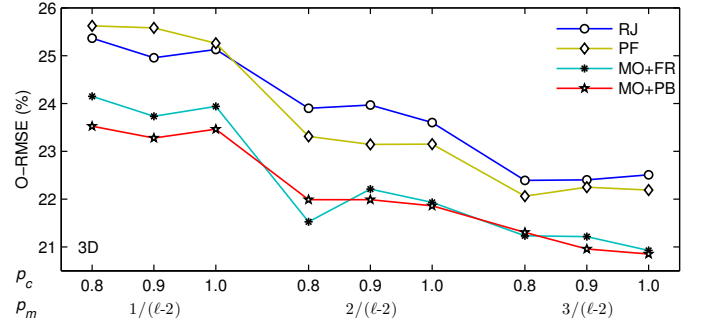


Figure 26: Evaluating different GA settings. Three-dimensional case.

the different GA settings. The PF and MO+PB strategies require the tuning of the penalty weight (ρ) and the bias strength (ω) parameters, respectively. For each evaluated configuration of the GA, the results of PF and MO+PB reported in Figures 25 and 26 correspond to the best O-RMSE obtained by considering a diverse set of values for the respective parameters (see Section 6.3 and Appendix C.2 for details). It is evident from Figures 25 and 26 that both MO+FR and MO+PB achieved lower O-RMSE values in all cases when compared with respect to RJ and PF. The MO+PB strategy tends to perform better than MO+FR for most GA settings, particularly when focusing on the two-dimensional instances. Finally, the plots indicate that the use of PF yields better results in comparison to the use of RJ in most cases. In general, no clear tendency in the GA's performance can be distinguished with respect to the variation in the recombination probability. It is possible to observe, however, that regardless of the constraint-handling strategy used the GA responded positively to the increased mutation rate.

For further analyses presented in Section 7.2.2, the settings for the GA which allowed each of the compared approaches to reach the lowest O-RMSE value have been selected. The selected recombination probabilities are as follows: (i) two-dimensional instances, $p_c = 0.8$ for RJ, MO+FR and MO+PB, and $p_c = 1.0$ for PF; (ii) three-dimensional instances, $p_c = 0.8$ for RJ and PF, and $p_c = 1.0$ for MO+FR and MO+PB. The mutation probability was set to $p_m = \frac{3}{\ell-2}$ in all the cases.

7.2.2. Comparative analysis

A detailed comparative analysis among the RJ, PF, MO+FR and MO+PB strategies is presented in this section. In the re-

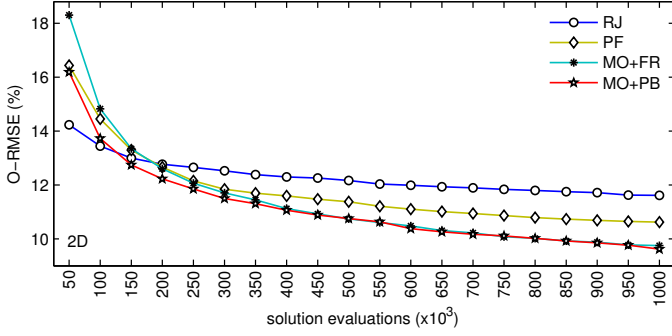


Figure 27: O-RMSE scored by the GA as the search process progressed. Two-dimensional test instances.

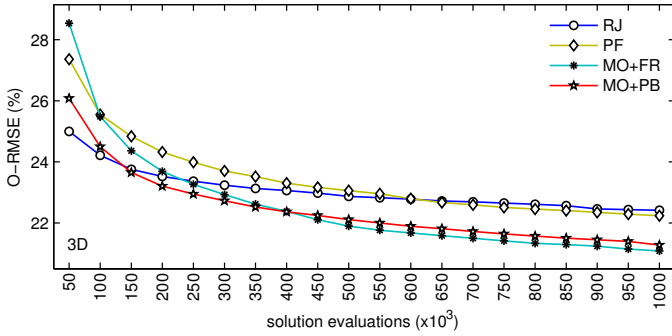


Figure 28: O-RMSE scored by the GA as the search process progressed. Three-dimensional test instances.

ported experiments, the best performing parameter settings for PF and MO+PB are considered (refer to Section 6.3 and Appendix C.2 for details). Likewise, the best performing GA conditions for each of the approaches are used (Section 7.2.1).

Figures 27 and 28 show the online convergence (measured in terms of the O-RMSE) presented by the GA when using the different constraint-handling approaches. The progress in the search is reported in slots of 50,000 solution evaluations until reaching a maximum allowed number of 10^6 evaluations. These figures are quite revealing in several respects. First, it is possible to note from the plots that the best results at the end of the search process were obtained by using the multi-objective strategies (MO+FR and MO+PB), in both the two- and the three-dimensional test cases. The RJ method, which exhibited the worst performance at the end, scored the best O-RMSE values at the beginning of the search. Thus, the use of RJ enabled a faster convergence towards moderate-quality individuals. Given that PF, MO+FR and MO+PB invest an additional amount of effort in evaluating infeasible protein conformations, these strategies require more time to locate promising regions of the solutions space. By allowing the algorithm to move through infeasible states, however, these methods are more likely to reach better results at the end of the optimization process; as it can be perceived from the slope in the corresponding convergence curves. Finally, although MO+FR and MO+PB competed with the best O-RMSE values at the end, it is important to observe that MO+FR showed a significantly inferior performance at the first stages of the search (indeed

the poorest performance among the four compared techniques). This is because the bias introduced in MO+FR is not as restrictive as that involved in MO+PB and PF, so that MO+FR dedicates more resources to the exploration of infeasible regions.

Tables 5 and 6 detail the above presented results of the GA after 10^6 solution evaluations. The information in these tables is organized in the same manner as in Tables 2 and 3 described in Section 7.1. As shown in Tables 5 and 6, both the MO+FR and MO+PB strategies reached a better average energy for most of the instances, thereby lowering the O-RMSE, in comparison to RJ and PF. While MO+PB scored the best O-RMSE value for the two-dimensional instances, MO+FR obtained the lowest value for this measure in the three-dimensional case. Even though no important conclusions can be drawn regarding the superiority of the multi-objective methods with respect to each other, it is possible to see from the tables that MO+FR achieved a better \bar{E} value in most cases. Finally, despite the poor overall performance of RJ, this strategy outperformed the other three approaches at solving one two-dimensional instance (2d12) and a two of the three-dimensional test cases (3d2 and 3d7).

Table 7 outlines the results of the statistical significance analysis. The interpretation of this table is the same as for Table 4 described at the end of Section 7.1. As it can be observed from Table 7, no significant performance differences between the four compared approaches were found when dealing with the smallest test instances; the four studied constraint-handling methods scored similarly competitive results. PF was significantly superior to RJ in 5 of the instances, but significantly inferior at solving the 2d12 instance. Both MO+FR and MO+PB significantly increased the performance of the GA in 15 of the test cases with respect to RJ. In addition, these multi-objective strategies statistically outperformed the PF approach in 13 and 10 instances. Nonetheless, MO+FR presented a significantly lower performance in 3 and 2 of the test cases in comparison to RJ and PF, respectively. Finally, there was a statistically significant difference between the multi-objective strategies for 8 of the instances; in 3 of these cases such a significant difference favors MO+PB, while it favors MO+FR in the 5 remaining cases.

8. Conclusions

The multi-objective (MO) approach to constraint-handling has been investigated in the context of the HP model for protein structure prediction (PSP). The HP model was reformulated as an unconstrained multi-objective problem by treating constraints as an additional objective function. Rather than discriminating feasible from infeasible solutions, the MO strategy defines trade-offs between quality (original objective) and feasibility. This gives infeasible solutions the opportunity to be considered and exploited during optimization.

In the first part of this study, a thorough fitness landscape analysis was conducted in order to evaluate the effects that the (single-objective to multi-objective) problem transformation involves. As a result, it was found that a significant portion of the infeasibility translates into landscape neutrality. Under the MO problem formulation, it is possible for an infeasible solution to become part of the neutral neighborhood of a feasible solution.

Table 5: Results obtained by the GA when using the studied constraint-handling strategies. Two-dimensional test cases.

Seq.	ℓ	E^*	RJ		PF		MO+FR		MO+PB	
			$E_b(\nu)$	\bar{E}	$E_b(\nu)$	\bar{E}	$E_b(\nu)$	\bar{E}	$E_b(\nu)$	\bar{E}
2d1	18	-4	-4 (98)	-3.98	-4 (100)	-4.00	-4 (100)	-4.00	-4 (100)	-4.00
2d2	18	-8	-8 (100)	-8.00	-8 (100)	-8.00	-8 (100)	-8.00	-8 (100)	-8.00
2d3	18	-9	-9 (100)	-9.00	-9 (100)	-9.00	-9 (99)	-8.99	-9 (100)	-9.00
2d4	20	-9	-9 (99)	-8.99	-9 (100)	-9.00	-9 (100)	-9.00	-9 (100)	-9.00
2d5	20	-10	-10 (97)	-9.94	-10 (100)	-10.00	-10 (100)	-10.00	-10 (100)	-10.00
2d6	24	-9	-9 (93)	-8.93	-9 (86)	-8.86	-9 (94)	-8.94	-9 (96)	-8.96
2d7	25	-8	-8 (57)	-7.57	-8 (82)	-7.82	-8 (95)	-7.95	-8 (90)	-7.90
2d8	36	-14	-14 (2)	-11.76	-14 (2)	-12.11	-14 (4)	-11.95	-14 (3)	-12.27
2d9	48	-23	-22 (2)	-18.90	-22 (2)	-18.96	-22 (7)	-19.67	-22 (6)	-19.58
2d10	50	-21	-21 (15)	-19.19	-21 (15)	-19.17	-21 (33)	-20.14	-21 (31)	-19.77
2d11	60	-36	-33 (3)	-30.10	-34 (2)	-30.78	-35 (1)	-31.37	-34 (2)	-30.90
2d12	64	-42	-39 (1)	-33.34	-38 (1)	-32.47	-37 (1)	-31.49	-38 (2)	-32.82
2d13	85	-53	-48 (2)	-42.96	-47 (3)	-43.04	-49 (2)	-43.20	-48 (1)	-43.59
2d14	100	-48	-40 (2)	-35.70	-41 (2)	-36.46	-43 (1)	-37.12	-43 (1)	-37.02
2d15	100	-50	-43 (2)	-37.61	-44 (1)	-38.09	-44 (2)	-39.47	-43 (1)	-38.91
O-RMSE			11.61%		10.62%		9.75%		9.62%	

Table 6: Results obtained by the GA when using the studied constraint-handling strategies. Three-dimensional test cases.

Seq.	ℓ	E^*	RJ		PF		MO+FR		MO+PB	
			$E_b(\nu)$	\bar{E}	$E_b(\nu)$	\bar{E}	$E_b(\nu)$	\bar{E}	$E_b(\nu)$	\bar{E}
3d1	20	-11	-11 (100)	-11.00	-11 (100)	-11.00	-11 (100)	-11.00	-11 (100)	-11.00
3d2	24	-13	-13 (100)	-13.00	-13 (98)	-12.98	-13 (98)	-12.96	-13 (97)	-12.96
3d3	25	-9	-9 (98)	-8.98	-9 (99)	-8.99	-9 (100)	-9.00	-9 (100)	-9.00
3d4	36	-18	-18 (32)	-16.54	-18 (31)	-16.57	-18 (35)	-16.84	-18 (40)	-16.79
3d5	46	-35	-32 (1)	-27.78	-34 (1)	-28.45	-32 (4)	-28.92	-32 (1)	-28.34
3d6	48	-31	-30 (1)	-26.94	-31 (1)	-27.18	-31 (1)	-27.39	-30 (4)	-27.45
3d7	50	-34	-31 (2)	-27.91	-31 (1)	-27.75	-32 (1)	-27.40	-31 (1)	-27.78
3d8	58	-44	-38 (2)	-33.30	-38 (1)	-33.07	-41 (1)	-34.52	-38 (2)	-33.72
3d9	60	-55	-49 (2)	-43.61	-49 (1)	-44.02	-50 (1)	-44.45	-49 (1)	-44.60
3d10	64	-59	-53 (1)	-47.02	-53 (1)	-47.15	-51 (2)	-45.63	-52 (3)	-47.54
3d11	67	-56	-42 (2)	-37.71	-44 (1)	-37.78	-44 (3)	-38.68	-44 (1)	-38.28
3d12	88	-72	-52 (4)	-44.75	-50 (5)	-44.97	-54 (1)	-47.30	-53 (1)	-46.36
3d13	103	-58	-40 (1)	-34.36	-43 (1)	-33.76	-40 (5)	-35.35	-41 (1)	-35.16
3d14	124	-75	-53 (1)	-41.34	-50 (2)	-41.38	-51 (1)	-43.56	-53 (1)	-43.08
3d15	136	-83	-53 (1)	-44.44	-54 (2)	-44.69	-55 (1)	-46.94	-55 (2)	-46.28
O-RMSE			22.42%		22.24%		21.08%		21.27%	

Table 7: Statistical analysis for comparing the performance of the GA when using the different constraint-handling approaches analyzed.

	Two-dimensional instances															Three-dimensional instances															Overall
	2d1	2d2	2d3	2d4	2d5	2d6	2d7	2d8	2d9	2d10	2d11	2d12	2d13	2d14	2d15	3d1	3d2	3d3	3d4	3d5	3d6	3d7	3d8	3d9	3d10	3d11	3d12	3d13	3d14	3d15	
PF / RJ							+	+			+	-		+						+											5+1-
MO+FR / RJ							+		+	+	+	-		+	+		+	+	-	+	+	-	+	+	+	+	+	+	+	+	15+3-
MO+PB / RJ							+	+	+	+	+		+	+	+		+	+		+	+		+		+	+	+	+	+	+	15+0-
MO+FR / PF							+		+	+	+	-		+	+		+			+		+	-	+	+	+	+	+	+	+	13+2-
MO+PB / PF						+			+	+					+					+	+		+	+		+	+	+	+	+	10+0-
MO+PB / MO+FR								+		-	-	+					-			-		-	+		-						3+5-

This has prompted an important increase in the neutrality degree of solutions and, consequently, in the size of the neutral networks (NNs). Such a landscape transformation has led to the establishment of neutral connections between feasible and infeasible NNs. Through a series of neutral connections, however, it is possible to bridge different regions of the feasible space, potentially belonging to diverse fitness classes (this can be especially useful when dealing with disjoint feasible spaces). By being allowed to traverse (originally inaccessible) infeasible areas, a search algorithm can thus exploit these neutral connections in the form of new neutral paths to navigate the landscape. The new defined neutral paths can not only be shorter than the existing feasible paths, but can also play a central role in helping the algorithm to escape from local optima.

Despite the aforementioned advantages that the alternative multi-objective landscape entails, an excessive increase in the neutrality may also prevent a search algorithm from moving in the correct direction. The conducted landscape analysis not only reported a considerable growth in the size of the NNs due to the use of the MO problem formulation. It was also found that these NNs are mainly composed of infeasible solutions. Without a proper search bias, therefore, the computational resources can be exhausted by exploring uninteresting areas of the solution space (as it was also pointed out by Runarsson and Yao [14]). From the fitness landscape perspective, providing the MO strategy with a search bias can be understood as the removal of part of the neutrality that this strategy originally introduces. The goal becomes, thus, to benefit from having access to the infeasible areas of the landscape, at the same time that most of the effort is invested in exploiting promising search directions. The second part of this work studied the effectiveness of different mechanisms for biasing the search towards the feasible region, which can be coupled to the MO constraint-handling strategy. Three different biasing mechanisms were evaluated; namely, the use of an archiving strategy, the incorporation of a secondary discrimination criterion (use of feasibility rules), and the application of a proportional bias dependent on the degree of constraint violation. On the one hand, the results of such an evaluation confirmed the need for performing a well-biased search when using the MO strategy. The behavior of the considered search algorithms was significantly improved with the implementation of all the three studied biasing approaches. On the other hand, it was also possible to observe that a very strong bias could lead to override the positive effects of the landscape transformation. Thus, the task of identifying the most appropriate amount of bias for a particular problem and search algorithm, could be not as straightforward as might be thought.

In the last part of this study, the MO constraint-handling strategy was further explored by carrying out a comparative analysis where two different approaches from the literature were considered: namely, a rejecting strategy (RJ) where the search is confined to the space of only feasible conformations, and a penalty function (PF) where infeasible solutions are penalized according to the number of conflicts they present. The different strategies were evaluated in terms of the performance of basic evolutionary algorithms. As a result, the use of the MO strategy significantly improved the performance of the imple-

mented algorithms when compared with respect to both the RJ and PF methods. This highlights the suitability of the studied MO approach. It was also found that PF scored better results in most cases with regard to the RJ strategy. The fact that both MO and PF performed better than RJ, and that RJ requires a considerable amount of additional computational resources, gives further support to the belief that considering infeasible protein conformations may contribute to the design of more competitive algorithms for solving the HP model of the PSP problem; this has been a subject of concern in the specialized literature.

To the best of the authors' knowledge, the preliminary results of this research, as reported in [13], represent the first efforts on the use of multi-objective optimization methods to face the constraint-handling requirement which arises when dealing with the HP model of the PSP problem. Basic evolutionary algorithms have been used in this study for evaluating the suitability of this approach. From the obtained results, it is expected that the MO strategy can be incorporated as a means of improving the performance of established state-of-the-art algorithms for solving this problem. This issue needs to be investigated in order to derive more general conclusions. Furthermore, the present study explored for the first time, as far as the authors are aware, the potential effects of the MO constraint-handling strategy through a fitness landscape analysis. Although such an analysis focused on a particular case of study, the HP model of the PSP problem, similar effects to those observed with regard to the landscape transformation can be expected from the use of the MO strategy in other problem domains. Therefore, the performed analysis contributes to the general understanding of the MO approach for handling constraints. It seems important, however, to replicate this analysis to different problems in order to further support the acquired understanding.

Appendix A. Test cases, performance assessment and experimental conditions

Appendix A.1. Performance measures

The experimental results reported in this paper are evaluated in terms of the conventional energy function of the HP model, as it was introduced in Section 2.3. This is the objective function of the studied problem. As such, the energy value of the candidate protein conformations has been the primary quality indicator adopted in the specialized literature.

Two additional performance measures were considered, both computed over multiple independent executions of the implemented search algorithms. First, the *relative root mean square error* (RMSE) for a given test instance t is defined as follows:

$$\text{RMSE}(t) = 100\% \sqrt{\frac{1}{R} \sum_{r=1}^R \left(\frac{E_r(t) - E^*(t)}{E^*(t)} \right)^2}, \quad (\text{A.1})$$

where $E_r(t)$ denotes the energy of the best solution found during a single execution r , R is the total number of executions carried out, and $E^*(t)$ is the optimal (or best known) energy value for instance t . Thus, RMSE indicates the performance scored for a particular instance t as the average deviation between the

quality of the achieved solutions and the quality of the target solution.²⁴ While the range of possible energy values varies from instance to instance, RMSE is defined in a common 0% to 100% scale. This makes possible to evaluate (and plot) together the results obtained for the different considered test instances. $RMSE(t) = 0\%$ is the preferred value for this measure.

Finally, the *overall relative root mean square error* (O-RMSE) measure extends RMSE in order to assess the overall performance of the studied approaches, considering all the test instances. Having defined RMSE, O-RMSE can be formally stated as follows:

$$O-RMSE = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} RMSE(t), \quad (A.2)$$

where \mathcal{T} is the set of all instances. In this way, $O-RMSE = 0\%$ suggests the ideal situation where the optimal solution for each instance was reached during all the performed executions. By sacrificing details about the scored performance on each particular test instance, this global measure makes possible to investigate thoroughly the influence of varying important parameters of the studied techniques and search algorithms, as well as to analyze the search dynamics during optimization.

Appendix A.2. Test instances

A total of 30 well-known benchmark sequences for the HP model have been considered for the experimentation of this research project. Out of them, 15 are for the two-dimensional square lattice and the other 15 are for three-dimensional cubic lattice. Tables A.8 and A.9 present the full HP sequences, their length (ℓ) and the optimal or best known energy value (E^*) reported in the literature [25, 26, 31, 45, 94, 95].

Appendix A.3. Statistical significance analysis

The statistical significance analysis was conducted as follows. First, *D'Agostino-Pearson's omnibus K^2* test was used to evaluate the normality of data distributions. For normally distributed data, either *ANOVA* or the *Welch's t* parametric tests were used depending on whether the variances across the samples were homogeneous (*homoskedasticity*) or not. This was investigated using the *Bartlett's* test. For non-normal data, the nonparametric *Kruskal-Wallis* test was adopted. Finally, a significance level of $\alpha = 0.05$ has been considered.

Appendix A.4. Experimental platform

The algorithms implemented in this study were coded in ANSI C and compiled with *gcc* using the optimization flag *-O3*. All experiments performed were run sequentially on the Neptuno cluster at the Information Technology Laboratory, CINVESTAV-Tamaulipas. This cluster is equipped with 10 InfiniBand interconnected nodes, each of which features 8 cores running at 2.66 GHz, has a total of 16 GB of RAM, and uses the CentOS distribution of the Linux operating system.

²⁴It is worth noting that, as a measure of central tendency, RMSE conceals the performance exhibited on each individual execution. This implies a possible conflicting behavior between some individual execution results and the concerned measure which is computed over the entire sample of executions. Nevertheless, the use of this measure can lead to draw more general conclusions with respect to the behavior of the studied techniques.

Table A.8: Test instances for the two-dimensional square lattice.

	Sequence	ℓ	E^*
2d1	H ₂ P ₂ H ₂ P ₃ HP ₃ HP	18	-4
2d2	HHPHPH ₃ P ₃ H ₄ P ₂ H ₂	18	-8
2d3	PHP ₂ HPH ₃ PH ₂ PH ₅	18	-9
2d4	HHPHP ₂ H ₂ PHP ₂ HPH ₂ P ₂ HPH	20	-9
2d5	H ₃ P ₂ HPHPHP ₂ HPHPHP ₂ H	20	-10
2d6	H ₂ P ₂ HP ₂ HP ₂ HP ₂ HP ₂ HP ₂ HP ₂ H ₂	24	-9
2d7	P ₂ HP ₂ H ₂ P ₄ H ₂ P ₄ H ₂ P ₄ H ₂	25	-8
2d8	P ₃ H ₂ P ₂ H ₂ P ₃ H ₇ P ₂ H ₂ P ₄ H ₂ P ₂ HP ₂	36	-14
2d9	P ₂ HP ₂ H ₂ P ₂ H ₂ P ₃ H ₁₀ P ₆ H ₂ P ₂ H ₂ P ₂ HP ₂ H ₅	48	-23
2d10	H ₂ (PH) ₄ H ₃ P(HP ₃) ₃ (P ₃ H) ₃ PH ₄ (PH) ₄ H	50	-21
2d11	P ₂ H ₃ PH ₃ P ₃ H ₁₀ PHP ₃ H ₁₂ P ₄ H ₆ PH ₂ PHP	60	-36
2d12	H ₁₂ PHPH(P ₂ H ₂ P ₂ H ₂ P ₂ H) ₃ PHPH ₁₂	64	-42
2d13	H ₄ P ₄ H ₁₂ P ₆ (H ₁₂ P ₃) ₃ HP ₂ H ₂ P ₂ H ₂ HPH	85	-53
2d14	P ₆ HPH ₂ P ₅ H ₃ PH ₃ PH ₂ P ₄ H ₂ P ₂ H ₂ PH ₃ PH ₁₀ PH ₂ PH ₇ P ₁₁ H ₇ P ₂ HPH ₃ P ₆ HPH ₂	100	-48
2d15	P ₃ H ₂ P ₂ H ₄ P ₂ H ₃ PH ₂ PH ₂ PH ₄ P ₈ H ₆ P ₂ H ₆ P ₉ HPH ₂ PH ₁₁ P ₂ H ₃ PH ₂ PHP ₂ HPH ₃ P ₆ H ₃	100	-50

Table A.9: Test instances for the three-dimensional cubic lattice.

	Sequence	ℓ	E^*
3d1	HHPHP ₂ H ₂ PHP ₂ HPH ₂ P ₂ HPH	20	-11
3d2	H ₂ P ₂ HP ₂ HP ₂ HP ₂ HP ₂ HP ₂ HP ₂ H ₂	24	-13
3d3	P ₂ HP ₂ H ₂ P ₄ H ₂ P ₄ H ₂ P ₄ H ₂	25	-9
3d4	P ₃ H ₂ P ₂ H ₂ P ₃ H ₇ P ₂ H ₂ P ₄ H ₂ P ₂ HP ₂	36	-18
3d5	P ₂ H ₃ PH ₃ P ₃ HPH ₂ PH ₂ P ₂ HPH ₄ PHP ₂ H ₅ PHPH ₂ P ₂ H ₂ P	46	-35
3d6	P ₂ HP ₂ H ₂ P ₂ H ₂ P ₃ H ₁₀ P ₆ H ₂ P ₂ H ₂ P ₂ HP ₂ H ₅	48	-31
3d7	H ₂ (PH) ₄ H ₃ P(HP ₃) ₃ (P ₃ H) ₃ PH ₄ (PH) ₄ H	50	-34
3d8	PH(PH ₃) ₂ P(PH ₂ PH) ₂ H(HP ₃) ₃ (H ₂ P ₂ H) ₂ PHP ₄ (H(P ₂ H) ₂) ₂	58	-44
3d9	P ₂ H ₃ PH ₃ P ₃ H ₁₀ PHP ₃ H ₁₂ P ₄ H ₆ PH ₂ PHP	60	-55
3d10	H ₁₂ PHPH(P ₂ H ₂ P ₂ H ₂ P ₂ H) ₃ PHPH ₁₂	64	-59
3d11	P(HPH ₂ PH ₂ PHP ₂ H ₃ P ₃) ₃ (HPH) ₃ P ₂ H ₃ P	67	-56
3d12	P(HPH) ₃ P ₂ H ₂ (P ₂ H) ₆ H(P ₂ H ₃) ₄ P ₂ (HPH) ₃ P ₂ HP(PHP ₂ H ₂ HP) ₂	88	-72
3d13	P ₂ H ₂ P ₅ H ₂ P ₂ H ₂ PHP ₂ HP ₂ HP ₃ H ₂ PH ₂ P ₆ HP ₂ HP HP ₂ HP ₃ H ₃ P ₄ H ₂ PH ₂ P ₅ H ₂ P ₄ HPH ₃ H ₅ P ₂ HP ₂	103	-58
3d14	P ₃ H ₃ PHP ₄ HP ₅ H ₂ P ₄ H ₂ P ₂ H ₂ (P ₄ H) ₂ P ₂ HP ₂ H ₂ P ₃ H ₂ PHPH ₃ P ₄ H ₃ P ₆ H ₂ P ₂ HP ₂ HPHP ₂ HP ₂ HP ₂ H ₃ P ₄ HPH ₃ H ₅ P ₄ H ₂ (PH) ₄	124	-75
3d15	HP ₃ HP ₄ HPH ₂ PH ₂ P ₄ HPH ₃ P ₄ HPHHPH ₄ P ₁₁ HP ₂ HP ₃ HPH ₂ P ₃ H ₂ HP ₂ HPHHPH ₃ HP ₃ H ₆ P ₃ H ₂ P ₂ H ₃ P ₃ H ₂ PH ₅ P ₉ HP ₄ HPH ₄	136	-83

Appendix B. Algorithms

Appendix B.1. Basic (1+1) evolutionary algorithm

The so-called (1+1) evolutionary algorithm (EA) is described in Algorithm 3. First, an initial parent individual \mathbf{x} is generated at random. At each generation, an offspring \mathbf{x}' is created by randomly and independently mutating \mathbf{x} at each encoding position with a given probability p_m . The new individual \mathbf{x}' is rejected only if it is strictly worse than the parent individual \mathbf{x} , otherwise \mathbf{x}' is accepted as the starting point for the next generation.

The acceptance criterion in the above described (1+1) EA, line 4 in Algorithm 3, depends upon the constrain-handling strategy to be applied. On the one hand, it can be based on the one-dimensional objective (energy) value of the candidate conformations, either including penalties or not (penalty

Algorithm 3 Basic (1+1) evolutionary algorithm.

```
1: choose  $\mathbf{x} \in X$  uniformly at random
2: repeat
3:    $\mathbf{x}' \leftarrow \text{mutate}(\mathbf{x})$ 
4:   if  $\mathbf{x}'$  is not worse than  $\mathbf{x}$  then
5:      $\mathbf{x} \leftarrow \mathbf{x}'$ 
6:   end if
7: until  $< \text{stop condition} >$ 
```

function, PF and reject, RJ strategies, respectively). On the other hand, acceptance will be based on the Pareto-dominance relation when applying the multi-objective, MO constraint-handling strategy, which is the focus of this research.

In the implemented (1+1) EA, individuals encode protein conformations using an internal coordinates representation based on relative moves, as detailed in Section 2.3.1. In all the cases, the mutation probability was fixed to $p_m = \frac{1}{\ell-2}$, where $\ell - 2$ denotes the length of the encoding. Finally, a number of 10^6 evaluations was adopted as the stopping condition.

Appendix B.2. Archiving (1+1) evolutionary algorithm

An archiving variant of the (1+1) evolutionary algorithm (EA) described in Appendix B.1 is sketched in Algorithm 4. In the archiving (1+1) EA, an external archive stores the nondominated solutions (in the Pareto sense) found along the evolutionary process. The implemented archiving strategy influences the search behavior of the algorithm in such a way that the offspring \mathbf{x}' is only accepted if it is not dominated by any individual in the archive. If accepted, \mathbf{x}' is included in the archive and all individuals dominated by \mathbf{x}' , and those mapping to the same objective vector $\mathbf{f}(\mathbf{x}')$, are removed. In the context of this study, the use of this archiving variant of the (1+1) EA makes only sense when implementing the multi-objective, MO strategy for constraint-handling.

Algorithm 4 Archiving (1+1) evolutionary algorithm.

```
1: choose  $\mathbf{x} \in X$  uniformly at random
2:  $\mathcal{A} \leftarrow \{\mathbf{x}\}$ 
3: repeat
4:    $\mathbf{x}' \leftarrow \text{mutate}(\mathbf{x})$ 
5:   if  $\nexists \hat{\mathbf{x}} \in \mathcal{A} : \hat{\mathbf{x}} < \mathbf{x}'$  then
6:      $\mathcal{A} \leftarrow \{\hat{\mathbf{x}} \in \mathcal{A} : \mathbf{x}' \not\prec \hat{\mathbf{x}} \wedge \mathbf{f}(\hat{\mathbf{x}}) \neq \mathbf{f}(\mathbf{x}')\} \cup \{\mathbf{x}'\}$ 
7:      $\mathbf{x} \leftarrow \mathbf{x}'$ 
8:   end if
9: until  $< \text{stop condition} >$ 
```

Appendix B.3. Genetic algorithm

The basic structure of the implemented genetic algorithm is presented in Algorithm 5. First, an initial parent population \mathcal{P} of size N is randomly generated. At each generation, the fittest individuals in \mathcal{P} are selected for mating (*selection-for-variation*). Then, a children population \mathcal{P}' is created by applying the genetic operators to the selected parents $\hat{\mathcal{P}}$. Finally, the

parent and children populations are combined and the best individuals are selected to survive in order to form the new parent population (*selection-for-survival*).

Algorithm 5 Genetic algorithm.

```
1: choose  $\mathcal{P} \subset X : |\mathcal{P}| = N$  uniformly at random
2: while  $< \text{stop condition} >$  do
3:    $\hat{\mathcal{P}} \leftarrow \text{selection-for-variation}(\mathcal{P})$ 
4:    $\mathcal{P}' \leftarrow \text{variation}(\hat{\mathcal{P}})$ 
5:    $\mathcal{P} \leftarrow \text{selection-for-survival}(\mathcal{P} \cup \mathcal{P}')$ 
6: end while
```

A crucial issue in GAs is selection; that is, how the discrimination among the individuals is carried out (at both the *selection-for-variation* and the *selection-for-survival* steps). In the implemented GA, this depends on the constraint-handling technique to be used. When using the reject (RJ) and penalty function (PF) approaches, a single-objective comparison among the individuals is performed. In contrast, when applying the multi-objective (MO) strategy for handling constraints the Pareto-dominance relation imposes a partial order among candidate individuals. The *nondominated sorting* procedure is used at the *selection-for-survival* stage, as it is implemented within the *nondominated sorting genetic algorithm II*, NSGA-II [93]. Roughly, the functioning of the nondominated sorting procedure is as follows. The nondominated individuals are initially identified and isolated into the first nondominated layer, \mathcal{L}_1 . From the remainder of the population, the new nondominated solutions are identified and assigned to the second nondominated layer, \mathcal{L}_2 . The process repeats until each individual in the population is classified. At the *selection-for-survival* stage, individuals are selected layer by layer, starting from \mathcal{L}_1 , until completing the required number of individuals. Whenever the number of individuals in the current layer exceeds the available capacity of the population, the conventional NSGA-II uses the so-called *crowding distance* as a secondary discrimination criterion. This allows to promote population diversity. In this study, however, crowding distance has not been incorporated to avoid attributing the performance that the GA achieves through the use of the MO strategy to such a diversification mechanism.

An internal coordinates representation based on relative moves has been adopted, see Section 2.3 for details. Binary tournament selection was employed as mating strategy. The implemented genetic operators are as follows. One-point crossover is applied according to a given probability p_c . In mutation, each encoding position is randomly and independently perturbed with probability p_m . It is worthy to mention that preliminary testing has been conducted in order to explore the effects of preventing duplicate individuals (clones) from the population. As a result, the performance of the different analyzed constraint-handling methods was significantly improved in all the cases when duplicate individuals were removed from the population; this mechanism was enabled for all the reported experiments. Finally, a maximum number of 10^6 solution evaluations was used as the termination criterion.

Appendix B.4. Iterated local search algorithm

Algorithm 6 outlines the general structure of a basic iterated local search (ILS) algorithm. The algorithm starts with a randomly generated conformation, denoted as \mathbf{x} . Then, a local search strategy is applied to \mathbf{x} until a local optimum \mathbf{x}^* is found. At each iteration, a perturbation \mathbf{x}' of the current local optimum \mathbf{x}^* is obtained and used as the starting point of another round of local search. After each local search, the new local optimum solution found \mathbf{x}'^* may be accepted as the new incumbent solution \mathbf{x}^* , based on a given acceptance criterion. This iterative procedure is repeated until a given stop condition is met.

Algorithm 6 Iterated local search (ILS).

```

1: choose  $\mathbf{x} \in \mathcal{X}$  uniformly at random
2:  $\mathbf{x}^* \leftarrow \text{LocalSearch}(\mathbf{x})$ 
3: repeat
4:    $\mathbf{x}' \leftarrow \text{Perturbation}(\mathbf{x}^*)$ 
5:    $\mathbf{x}'^* \leftarrow \text{LocalSearch}(\mathbf{x}')$ 
6:    $\mathbf{x}^* \leftarrow \text{AcceptanceCriterion}(\mathbf{x}^*, \mathbf{x}'^*)$ 
7: until  $< \text{stop condition} >$ 

```

Three main components, which determine the behavior of an ILS algorithm, have to be defined: (i) the embedded local search heuristic; (ii) the perturbation strength; and (iii) the acceptance criterion. A *best improvement local search* algorithm was used as the embedded heuristic, and basic settings for the perturbation strength and acceptance criterion were adopted according to the results reported in [96].

Appendix C. Baseline constraint-handling methods

Appendix C.1. Reject strategy

A basic reject strategy (RJ) was considered where only feasible protein conformations are accepted during the search process. A basic single-solution-based evolutionary algorithm (EA), the (1+1) EA, and a genetic algorithm (GA) are used in this study (refer to Appendix B for details). In order to implement the RJ strategy, the variation operators were adapted as follows. In the (1+1) EA, once mutation is to be applied to a particular encoding position (determined based on a given probability), all possible perturbations to this position are evaluated in random order until a feasible conformation is obtained. If no change in this position leads to a feasible conformation, the original value is restored. The GA uses a one-point crossover operator; in this operator all possible crossover points are explored in random order until feasible children are produced; otherwise, either one or both of the parents are copied unchanged. The mutation operator of the GA was adapted in the same manner as described above for the (1+1) EA. Note that such a persistent application of the variation operators involves an additional computational effort. Furthermore, the RJ strategy requires the algorithms to be provided with initial feasible individuals. The backtracking procedure proposed in [7] was used for generating such initial feasible individuals.

The above described RJ strategy is equivalent to the one analyzed by Duarte-Flores and Smith within a GA [6]. Similar

strategies have also been adopted in the context of different search metaheuristics. For example, the *hypermutation* and *hypermacromutation* mechanisms, implemented in some immune system-based algorithms for the HP model reported in the literature, operate in a similar feasibility-preserving fashion. These operators iteratively apply a series of mutations to the input solution and infeasible solutions encountered during this process are always discarded [8, 31, 43].

Appendix C.2. Penalty function

A constraint-handling strategy based on the use of a penalty function (PF) has been considered in this study. In the PF strategy, the energy (objective) value of a candidate solution is penalized according to the number of collisions that the encoded protein conformation presents. More formally, PSP under the HP model is restated as the problem of minimizing an alternative objective function $f(\mathbf{x})$ defined as follows ($\mathbf{x} \in \mathcal{X}$):

$$f(\mathbf{x}) = E(\mathbf{x}) + \rho \times \zeta \times \text{Collisions}(\mathbf{x}), \quad (\text{C.1})$$

where $E(\mathbf{x})$ denotes the conventional energy function of the HP model introduced in Section 2.3. $\text{Collisions}(\mathbf{x})$ refers to the total number of amino acid pairs (a_i, a_j) in \mathbf{x} such that a_i and a_j collide at the same lattice position. Finally, the value of ζ is to be large enough that, assuming a *penalty weight* of $\rho = 1$, it holds that $f(\mathbf{x}_i) \leq 0, \forall \mathbf{x}_i \in \mathcal{X}_{\mathcal{F}}$ while $f(\mathbf{x}_j) > 0, \forall \mathbf{x}_j \in \mathcal{X} \setminus \mathcal{X}_{\mathcal{F}}$. By defining the penalty weight ρ within the range $[0, 1]$, it will then be possible to move from an *underpenalization* scenario ($\rho = 0$), where comparisons are only based on the original objective function of the problem, to an *overpenalization* scenario ($\rho = 1$), where the penalty term dominates discrimination [97]. In this study, ζ was set to $\zeta = 2\ell_H + 2$ for the two-dimensional square lattice and $\zeta = 4\ell_H + 2$ for the three-dimensional cubic lattice. These values represent upper bounds on the number of H - H topological contacts that can be formed in the corresponding lattices and have also been considered in [5]. It should be noted that the value of ζ depends on the total number of hydrophobic amino acids in the protein sequence, ℓ_H .

With the aim of investigating the importance of the penalty weight ρ , and also to enable a more reliable comparative analysis in Section 7, different settings for this parameter have been explored in this study. Figures C.29 and C.30 show the performance scored by the (1+1) evolutionary algorithm (EA) and the genetic algorithm (GA) when using the PF method with a series of different ρ values in the range $[0, 1]$. Performance is expressed in terms of the O-RMSE (see Appendix A.1), computed over a total of 31 independent repetitions for each experiment. In general, the worst behavior of both the (1+1) EA and the GA was exhibited when no penalties were applied ($\rho = 0$).²⁵ Figure C.29 indicates that $\rho = 0.15$ allowed the (1+1) EA to achieve its best performance at solving the two-dimensional instances, and all considered ρ values in the range $[0.15, 1]$ produced the best results for the three-dimensional case. Regarding the GA, it is possible to observe from Figure C.30 that the

²⁵In Figures C.29 and C.30, the results obtained when using the lowest considered ρ values (leftmost data) have not been displayed in order to highlight details in the most relevant part of the plots.

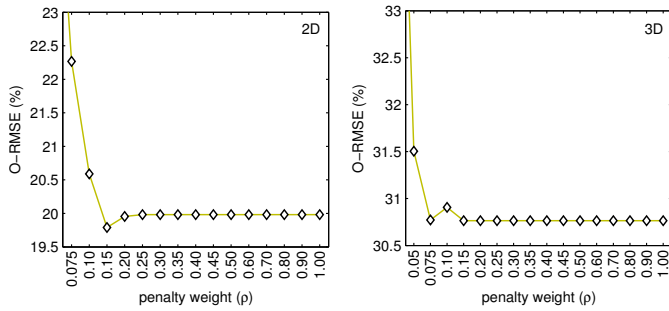


Figure C.29: Impact of varying the penalty weight (ρ) of the PF method on the (1+1) EA's performance. Two- (left) and three-dimensional (right) instances.

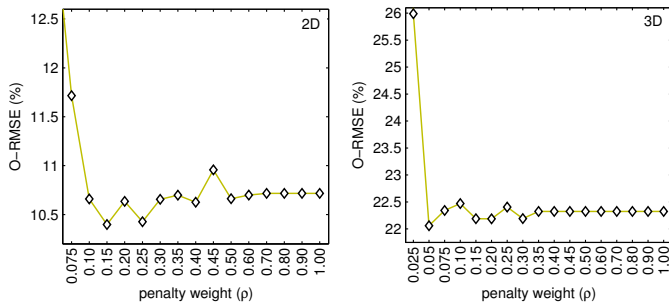


Figure C.30: Impact of varying the penalty weight (ρ) of the PF method on the performance of the GA. Two- (left) and three-dimensional (right) instances.

lowest O-RMSE values were reached by using $\rho = 0.15$ and $\rho = 0.05$ for the two-dimensional and three-dimensional test instances, respectively. The best performing settings for the PF method, as described above, have been considered during the comparative analysis conducted in Section 7.

Acknowledgment

The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Information Technology Laboratory, CINVESTAV-Tamaulipas. Also, the authors acknowledge support from CONACyT through projects 105060 and 99276.

References

- [1] K. Dill, Theory for the Folding and Stability of Globular Proteins, *Biochemistry* 24 (6) (1985) 1501–9.
- [2] K. Lau, K. Dill, A Lattice Statistical Mechanics Model of the Conformational and Sequence Spaces of Proteins, *Macromolecules* 22 (10) (1989) 3986–3997.
- [3] B. Berger, T. Leighton, Protein Folding in the Hydrophobic-Hydrophilic (HP) Model is NP-complete, in: *International Conference on Research in Computational Molecular Biology*, ACM, New York, NY, USA, 1998, pp. 30–39.
- [4] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, M. Yannakakis, On the Complexity of Protein Folding, in: *ACM Symposium on Theory of Computing*, ACM, Dallas, TX, USA, 1998, pp. 597–603.
- [5] N. Krasnogor, W. Hart, J. Smith, D. Pelta, Protein Structure Prediction With Evolutionary Algorithms, in: *Genetic and Evolutionary Computation Conference*, Morgan Kaufman, Orlando, FL, USA, 1999, pp. 1596–1601.
- [6] S. Duarte-Flores, J. Smith, Study of Fitness Landscapes for the HP model of Protein Structure Prediction, in: *IEEE Congress on Evolutionary Computation*, Vol. 4, Canberra, Australia, 2003, pp. 2338–2345.
- [7] C. Cotta, Protein Structure Prediction Using Evolutionary Algorithms Hybridized with Backtracking, in: *Artificial Neural Nets Problem Solving Methods*, Vol. 2687 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2003, pp. 321–328.
- [8] C. de Almeida, R. Gonçalves, M. Delgado, A Hybrid Immune-Based System for the Protein Folding Problem, in: *Evolutionary Computation in Combinatorial Optimization*, Vol. 4446 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Valencia, Spain, 2007, pp. 13–24.
- [9] J. Santos, M. Diéguez, Differential Evolution for Protein Structure Prediction Using the HP Model, in: *Foundations on Natural and Artificial Computation*, Vol. 6686 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2011, pp. 323–333.
- [10] T. Runarsson, X. Yao, Stochastic Ranking for Constrained Evolutionary Optimization, *IEEE Transactions on Evolutionary Computation* 4 (3) (2000) 284–294.
- [11] E. Mezura-Montes, C. A. Coello Coello, Constraint-handling in Nature-inspired Numerical Optimization: Past, Present and Future, *Swarm and Evolutionary Computation* 1 (4) (2011) 173–194.
- [12] C. Segura, C. Coello Coello, G. Miranda, C. León, Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimization, *4OR* 11 (3) (2013) 201–228.
- [13] M. Garza-Fabre, G. Toscano-Pulido and E. Rodriguez-Tello, Handling Constraints in the HP Model for Protein Structure Prediction by Multiobjective Optimization, in: *IEEE Congress on Evolutionary Computation*, Cancún, México, 2013, pp. 2728–2735.
- [14] T. Runarsson, X. Yao, Search Biases in Constrained Evolutionary Optimization, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 35 (2) (2005) 233–243.
- [15] V. Pareto, *Cours d'Economie Politique*, Droz, Genève, 1896.
- [16] S. Wright, The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution, *Proceedings of the 6th International Congress of Genetics* 1 (1932) 356–366.
- [17] P. Stadler, Fitness Landscapes, in: *Biological Evolution and Statistical Physics*, Vol. 585 of *Lecture Notes in Physics*, Springer Berlin / Heidelberg, 2002, pp. 183–204.
- [18] S. Verel, P. Collard, M. Tomassini, L. Vanneschi, Fitness Landscape of the Cellular Automata Majority Problem: View from the Olympus, *Theoretical Computer Science* 378 (1) (2007) 54–77.
- [19] L. Vanneschi, Y. Pirola, G. Mauri, M. Tomassini, P. Collard, S. Verel, A Study of the Neutrality of Boolean Function Landscapes in Genetic Programming, *Theoretical Computer Science* 425 (0) (2012) 34–57.
- [20] E. Pitzer, M. Affenzeller, A Comprehensive Survey on Fitness Landscape Analysis, in: *Recent Advances in Intelligent Engineering Systems*, Vol. 378 of *Studies in Computational Intelligence*, Springer Berlin Heidelberg, 2012, pp. 161–191.
- [21] K. Malan, A. Engelbrecht, A Survey of Techniques for Characterising Fitness Landscapes and Some Possible Ways Forward, *Information Sciences* 241 (2013) 148–163.
- [22] C. Anfinsen, Principles that Govern the Folding of Protein Chains, *Science* 181 (4096) (1973) 223–230.
- [23] H. Lopes, Evolutionary Algorithms for the Protein Folding Problem: A Review and Current Trends, in: *Computational Intelligence in Biomedicine and Bioinformatics*, Vol. 151 of *Studies in Computational Intelligence*, Springer Berlin / Heidelberg, 2008, pp. 297–315.
- [24] F. Custódio, H. Barbosa, L. Dardenne, A Multiple Minima Genetic Algorithm for Protein Structure Prediction, *Applied Soft Computing* 15 (Feb.) (2014) 88–99.
- [25] N. Krasnogor, B. Blackburne, E. Burke, J. Hirst, Multimeme algorithms for protein structure prediction, in: *Parallel Problem Solving from Nature*, Vol. 2439 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Granada, Spain, 2002, pp. 769–778.
- [26] M. Islam, M. Chetty, Clustered Memetic Algorithm With Local Heuristics for Ab Initio Protein Structure Prediction, *IEEE Transactions on Evolutionary Computation* 17 (4) (2013) 558–576.
- [27] M. Rashid, M. Newton, M. Hoque, S. Shatabda, D. Pham, A. Sattar, Spiral Search: A Hydrophobic-Core Directed Local Search for Simplified PSP on 3D FCC Lattice, *BMC Bioinformatics* 14 (Suppl 2) (2013) S16.
- [28] P. Pardalos, X. Liu, G. Xue, Protein Conformation of a Lattice Model

- Using Tabu Search, *Journal of Global Optimization* 11 (1) (1997) 55–68.
- [29] A. Shmygelska, H. Hoos, An Ant Colony Optimization Algorithm for the 2D and 3D Hydrophobic Polar Protein Folding Problem, *BMC Bioinformatics* 6 (1) (2005) 30.
 - [30] M. Nardelli, L. Tedesco, A. Bechini, Cross-lattice Behavior of General ACO Folding for Proteins in the HP Model, in: *ACM Symposium on Applied Computing, SAC '13*, ACM, Coimbra, Portugal, 2013, pp. 1320–1327.
 - [31] V. Cutello, G. Nicosia, M. Pavone, J. Timmis, An Immune Algorithm for Protein Structure Prediction on Lattice Models, *IEEE Transactions on Evolutionary Computation* 11 (1) (2007) 101–117.
 - [32] N. Mansour, F. Kanj, H. Khachfe, Particle Swarm Optimization Approach for Protein Structure Prediction in the 3D HP Model, *Interdisciplinary Sciences: Computational Life Sciences* 4 (3) (2012) 190–200.
 - [33] C. Zhou, C. Hou, Q. Zhang, X. Wei, Enhanced Hybrid Search Algorithm for Protein Structure Prediction Using the 3D-HP Lattice Model, *Journal of Molecular Modeling* 19 (9) (2013) 3883–3891.
 - [34] H. Lopes, R. Bitello, A Differential Evolution Approach for Protein Folding Using a Lattice Model, *Journal of Computer Science and Technology* 22 (6) (2007) 904–908.
 - [35] R. Santana, P. Larranaga, J. Lozano, Protein Folding in Simplified Models With Estimation of Distribution Algorithms, *IEEE Transactions on Evolutionary Computation* 12 (4) (2008) 418–438.
 - [36] B. Chen, L. Li, J. Hu, A Novel EDAs Based Method for HP Model Protein Folding, in: *IEEE Congress on Evolutionary Computation*, Trondheim, Norway, 2009, pp. 309–315.
 - [37] X. Cai, X. Wu, L. Wang, Q. Kang, Q. Wu, Hydrophobic-Polar Model Structure Prediction with Binary-Coded Artificial Plant Optimization Algorithm, *Journal of Computational and Theoretical Nanoscience* 10 (6) (2013) 1550–1554.
 - [38] B. Maher, A. Albrecht, M. Loomes, X. Yang, K. Steinhöfel, A Firefly-Inspired Method for Protein Structure Prediction in Lattice Models, *Biomolecules* 4 (1) (2014) 56–75.
 - [39] A. Patton, W. Punch III, E. Goodman, A Standard GA Approach to Native Protein Conformation Prediction, in: *International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 574–581.
 - [40] R. Unger, J. Moult, Genetic algorithm for 3d protein folding simulations, in: *International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993, pp. 581–588.
 - [41] C. Chira, D. Horvath, D. Dumitrescu, An Evolutionary Model Based on Hill-Climbing Search Operators for Protein Structure Prediction, in: *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, Vol. 6023 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2010, pp. 38–49.
 - [42] C. Chira, A Hybrid Evolutionary Approach to Protein Structure Prediction with Lattice Models, in: *IEEE Congress on Evolutionary Computation*, New Orleans, LA, USA, 2011, pp. 2300–2306.
 - [43] V. Cutello, G. Morelli, G. Nicosia, M. Pavone, G. Scollo, On Discrete Models and Immunological Algorithms for Protein Structure Prediction, *Natural Computing* 10 (1) (2011) 91–102.
 - [44] N. Lesh, M. Mitzenmacher, S. Whitesides, A Complete and Effective Move Set for Simplified Protein Folding, in: *International Conference on Research in Computational Molecular Biology*, ACM, Berlin, Germany, 2003, pp. 188–195.
 - [45] C. Thachuk, A. Shmygelska, H. Hoos, A Replica Exchange Monte Carlo Algorithm for Protein Folding in the HP Model, *BMC Bioinformatics* 8 (1) (2007) 342.
 - [46] C. M. Johnson, A. Katikireddy, A Genetic Algorithm with Backtracking for Protein Structure Prediction, in: *Genetic and Evolutionary Computation Conference*, ACM, Seattle, WA, USA, 2006, pp. 299–300.
 - [47] E. Talbi, *Metaheuristics: From Design to Implementation*, Wiley Publishing, 2009.
 - [48] M. Khimasia, P. Coveney, Protein Structure Prediction as a Hard Optimization Problem: The Genetic Algorithm Approach, *Molecular Simulation* 19 (4) (1997) 205–226.
 - [49] H. Lopes, M. Scapin, An Enhanced Genetic Algorithm for Protein Structure Prediction Using the 2D Hydrophobic-Polar Model, in: *Artificial Evolution*, Vol. 3871 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Lille, France, 2006, pp. 238–246.
 - [50] J. Knowles, R. Watson, D. Corne, Reducing Local Optima in Single-Objective Problems by Multi-objectivization, in: *Evolutionary Multi-Criterion Optimization*, Springer-Verlag, Zurich, Switzerland, 2001, pp. 269–283.
 - [51] J. Handl, S. Lovell, J. Knowles, Multiobjectivization by Decomposition of Scalar Cost Functions, in: *Parallel Problem Solving from Nature*, Vol. 5199 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Dortmund, Germany, 2008, pp. 31–40.
 - [52] M. Jensen, Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation, *Journal of Mathematical Modelling and Algorithms* 3 (4) (2004) 323–347.
 - [53] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, E. Zitzler, Do Additional Objectives Make a Problem Harder?, in: *Genetic and Evolutionary Computation Conference*, ACM, London, England, 2007, pp. 765–772.
 - [54] I. Vite-Silva, N. Cruz-Cortés, G. Toscano-Pulido, L. de la Fraga, Optimal Triangulation in 3D Computer Vision Using a Multi-objective Evolutionary Algorithm, in: *Applications of Evolutionary Computing*, Vol. 4448 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Valencia, Spain, 2007, pp. 330–339.
 - [55] R. Day, J. Zydallis, G. Lamont, Solving the Protein Structure Prediction Problem Through a Multi-Objective Genetic Algorithm, in: *IEEE/DARPA International Conference on Computational Nanoscience*, San Juan, PR, USA, 2002, pp. 32–35.
 - [56] V. Cutello, G. Narzisi, G. Nicosia, A Class of Pareto Archived Evolution Strategy Algorithms Using Immune Inspired Operators for Ab-Initio Protein Structure Prediction, in: *Applications of Evolutionary Computing*, Vol. 3449 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Lausanne, Switzerland, 2005, pp. 54–63.
 - [57] V. Cutello, G. Narzisi, G. Nicosia, A Multi-Objective Evolutionary Approach to the Protein Structure Prediction Problem, *Journal of The Royal Society Interface* 3 (6) (2006) 139–151.
 - [58] V. Cutello, G. Narzisi, G. Nicosia, Computational Studies of Peptide and Protein Structure Prediction Problems via Multiobjective Evolutionary Algorithms, in: *Multiobjective Problem Solving from Nature*, *Natural Computing Series*, Springer Berlin Heidelberg, 2008, pp. 93–114.
 - [59] J. Handl, S. Lovell, J. Knowles, Investigations into the Effect of Multiobjectivization in Protein Structure Prediction, in: *Parallel Problem Solving from Nature*, Vol. 5199 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Dortmund, Germany, 2008, pp. 702–711.
 - [60] D. Becerra, A. Sandoval, D. Restrepo-Montoya, L. Nino, A Parallel Multi-Objective Ab Initio Approach for Protein Structure Prediction, in: *IEEE International Conference on Bioinformatics and Biomedicine*, Hong Kong, China, 2010, pp. 137–141.
 - [61] C. Soares Brasil, A. Botazzo Delbem, D. Ferraz Bonetti, Investigating Relevant Aspects of MOEAs for Protein Structures Prediction, in: *Genetic and Evolutionary Computation Conference*, ACM, Dublin, Ireland, 2011, pp. 705–712.
 - [62] B. Olson, A. Shehu, Multi-Objective Stochastic Search for Sampling Local Minima in the Protein Energy Surface, in: *International Conference on Bioinformatics, Computational Biology and Biomedical Informatics*, ACM, Washington DC, USA, 2013, pp. 430–439.
 - [63] M. Garza-Fabre, E. Rodriguez-Tello, G. Toscano-Pulido, Multiobjectivizing the HP Model for Protein Structure Prediction, in: *Evolutionary Computation in Combinatorial Optimization*, Vol. 7245 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Málaga, Spain, 2012, pp. 182–193.
 - [64] M. Garza-Fabre, G. Toscano-Pulido, E. Rodriguez-Tello, Locality-based Multiobjectivization for the HP Model of Protein Structure Prediction, in: *Genetic and Evolutionary Computation Conference*, ACM, Philadelphia, PA, USA, 2012, pp. 473–480.
 - [65] M. Garza-Fabre, E. Rodriguez-Tello, G. Toscano-Pulido, An Improved Multiobjectivization Strategy for HP Model-Based Protein Structure Prediction, in: *Parallel Problem Solving from Nature*, Vol. 7492 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Taormina, Italy, 2012, pp. 82–92.
 - [66] D. Greiner, J. Emperador, G. Winter, B. Galván, Improving Computational Mechanics Optimum Design Using Helper Objectives: An Application in Frame Bar Structures, in: *Evolutionary Multi-Criterion Optimization*, Vol. 4403 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Matshushima, Japan, 2007, pp. 575–589.
 - [67] M. Jähne, X. Li, J. Branke, Evolutionary Algorithms and Multi-

- Objectivization for the Travelling Salesman Problem, in: Genetic and Evolutionary Computation Conference, ACM, Montreal, Canada, 2009, pp. 595–602.
- [68] D. Lochtefeld, F. Ciarallo, Helper-Objective Optimization Strategies for the Job-Shop Scheduling Problem, *Applied Soft Computing* 11 (6) (2011) 4161–4174.
- [69] D. Lochtefeld, F. Ciarallo, Multiobjectivization via Helper-Objectives With the Tunable Objectives Problem, *IEEE Transactions on Evolutionary Computation* 16 (3) (2012) 373–390.
- [70] L. Bui, M. Nguyen, J. Branke, H. Abbass, Tackling Dynamic Problems with Multiobjective Evolutionary Algorithms, in: *Multiobjective Problem Solving from Nature*, Natural Computing Series, Springer Berlin Heidelberg, 2008, pp. 77–91.
- [71] E. Segredo, C. Segura, C. Leon, A Multiobjectivised Memetic Algorithm for the Frequency Assignment Problem, in: *IEEE Congress on Evolutionary Computation*, New Orleans, LA, USA, 2011, pp. 1132–1139.
- [72] J. Mouret, Novelty-Based Multiobjectivization, in: *New Horizons in Evolutionary Robotics*, Vol. 341 of *Studies in Computational Intelligence*, Springer Berlin / Heidelberg, 2011, pp. 139–154.
- [73] S. Wessing, M. Preuss, G. Rudolph, Niching by Multiobjectivization with Neighbor Information: Trade-offs and Benefits, in: *IEEE Congress on Evolutionary Computation*, Cancún, México, 2013, pp. 103–110.
- [74] T. Tran, D. Brockhoff, B. Derbel, Multiobjectivization with NSGA-II on the Noiseless BBOB Testbed, in: *Genetic and Evolutionary Computation Conference*, ACM, Amsterdam, The Netherlands, 2013, pp. 1217–1224.
- [75] D. Sharma, K. Deb, N. Kishore, Customized Evolutionary Optimization Procedure for Generating Minimum Weight Compliant Mechanisms, *Engineering Optimization* 46 (1) (2014) 39–60.
- [76] S. Watanabe, K. Sakakibara, Multi-objective Approaches in a Single-objective Optimization Environment, in: *IEEE Congress on Evolutionary Computation*, Vol. 2, Edinburgh, UK, 2005, pp. 1714–1721.
- [77] D. Saxena, K. Deb, Trading on Infeasibility by Exploiting Constraints Criticality Through Multi-objectivization: A System Design Perspective, in: *IEEE Congress on Evolutionary Computation*, Singapore, 2007, pp. 919–926.
- [78] H. Singh, T. Ray, R. Sarker, Optimum Oil Production Planning Using Infeasibility Driven Evolutionary Algorithm, *Evolutionary Computation* 21 (1) (2013) 65–82.
- [79] A. Churchill, P. Husbands, A. Philippides, Multi-objectivization of the Tool Selection Problem on a Budget of Evaluations, in: *Evolutionary Multi-Criterion Optimization*, Vol. 7811 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2013, pp. 600–614.
- [80] M. Marmion, C. Dhaenens, L. Jourdan, A. Liefooghe, S. Vérel, On the Neutrality of Flowshop Scheduling Fitness Landscapes, in: *Learning and Intelligent Optimization*, Vol. 6683 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2011, pp. 238–252.
- [81] H. Lourenço, O. Martin, T. Stützle, Iterated Local Search: Framework and Applications, in: *Handbook of Metaheuristics*, Vol. 146 of *International Series in Operations Research & Management Science*, Springer US, 2010, pp. 363–397.
- [82] C. Reidys, P. Stadler, Neutrality in Fitness Landscapes, *Applied Mathematics and Computation* 117 (2-3) (2001) 321–350.
- [83] K. Dill, C. H., From Levinthal to Pathways to Funnels, *Nature Structural Biology* 4 (1) (1997) 10–19.
- [84] L. Vanneschi, M. Tomassini, P. Collard, S. Vérel, Y. Pirola, G. Mauri, A Comprehensive View of Fitness Landscapes with Neutrality and Fitness Clouds, in: *Genetic Programming*, Vol. 4445 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Valencia, Spain, 2007, pp. 241–250.
- [85] J. Knowles, D. Corne, Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors, *IEEE Transactions on Evolutionary Computation* 7 (2) (2003) 100–116.
- [86] M. López-Ibáñez, J. Knowles, M. Laumanns, On Sequential Online Archiving of Objective Vectors, in: *Evolutionary Multi-Criterion Optimization*, Vol. 6576 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Ouro Preto, Brazil, 2011, pp. 46–60.
- [87] E. Mezura-Montes, C. Coello Coello, E. Tun-Morales, Simple Feasibility Rules and Differential Evolution for Constrained Optimization, in: *MICA 2004: Advances in Artificial Intelligence*, Vol. 2972 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, México City, México, 2004, pp. 707–716.
- [88] A. Muñoz Zavala, A. Hernández Aguirre, E. Villa Diharce, Constrained Optimization via Particle Evolutionary Swarm Optimization Algorithm (PESO), in: *Genetic and Evolutionary Computation Conference*, Washington DC, USA, 2005, pp. 209–216.
- [89] A. Mani, C. Patvardhan, A Novel Hybrid Constraint Handling Technique for Evolutionary Optimization, in: *IEEE Congress on Evolutionary Computation*, Trondheim, Norway, 2009, pp. 2577–2583.
- [90] X. Yuan, A. Su, Y. Yuan, H. Nie, L. Wang, An Improved PSO for Dynamic Load Dispatch of Generators with Valve-Point Effects, *Energy* 34 (1) (2009) 67–74.
- [91] K. Deb, An Efficient Constraint Handling Method for Genetic Algorithms, *Computer Methods in Applied Mechanics and Engineering* 186 (2-4) (2000) 311–338.
- [92] A. Menchaca-Mendez, C. A. Coello Coello, A New Proposal to Hybridize the Nelder-Mead Method to a Differential Evolution Algorithm for Constrained Optimization, in: *IEEE Congress on Evolutionary Computation*, Trondheim, Norway, 2009, pp. 2598–2605.
- [93] K. Deb, S. Agrawal, A. Pratab, T. Meyarivan, A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II, in: *Parallel Problem Solving from Nature*, Vol. 1917 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Paris, France, 2000, pp. 849–858.
- [94] J. Zhang, S. C. Kou, J. S. Liu, Biopolymer Structure Simulation and Optimization via Fragment Regrowth Monte Carlo, *The Journal of Chemical Physics* 126 (22) (2007) 225101.
- [95] T. Wüst, Y. Li, D. Landau, Unraveling the Beautiful Complexity of Simple Lattice Model Polymers and Proteins Using Wang-Landau Sampling, *Journal of Statistical Physics* 144 (2011) 638–651.
- [96] M. Garza-Fabre, E. Rodríguez-Tello, G. Toscano-Pulido, Comparative Analysis of Different Evaluation Functions for Protein Structure Prediction under the HP Model, *Journal of Computer Science and Technology* 28 (5) (2013) 868–889.
- [97] T. Runarsson, X. Yao, Constrained Evolutionary Optimization: The Penalty Function Approach, in: *Evolutionary Optimization*, Vol. 48 of *International Series in Operations Research & Management Science*, Springer US, 2003, pp. 87–113.