A GRASP-based Memetic Algorithm with Path Relinking for the Far From Most String Problem

José E. Gallardo*, Carlos Cotta

Department Lenguajes y Ciencias de la Computación*, Universidad de Málaga, Campus de Teatinos, 29071 - Málaga, Spain.*

Abstract

The Far From Most String Problem (FFMSP) is a string selection problem. The objective is to find a string whose distance to other strings in a certain input set is above a given threshold for as many of those strings as possible. This problem has links with some tasks in computational biology and its resolution has been shown to be very hard. We propose a memetic algorithm (MA) to tackle the FFMSP. This MA exploits a heuristic objective function for the problem and features initialization of the population via a Greedy Randomized Adaptive Search Procedure (GRASP) metaheuristic, intensive recombination via path relinking and local improvement via hill climbing. An extensive empirical evaluation using problem instances of both random and biological origin is done to assess parameter sensitivity and draw performance comparisons with other state-of-the-art techniques. The MA is shown to perform better than these latter techniques with statistical significance.

Keywords: Far from most string problem, String selection problems, Bioinformatics, Metaheuristics, Memetic algorithms.

1. Introduction

From a very general point of view, *string selection problems* (SSPs) can be defined as a class of problems involving the construction or identification

*Corresponding author

Email addresses: pepeg@lcc.uma.es (José E. Gallardo), ccottap@lcc.uma.es (Carlos Cotta)

of a string of symbols satisfying certain constraints (typically distance constraints) with respect to another certain set of strings provided as input. Such problems have attracted a lot of interest for multiple reasons. From a theoretical (and even from a purely algorithmic) point of view, they constitute a clear and well-defined domain in which computational complexity issues can be analyzed and search/optimization algorithms can be put to work in challenging conditions. From a more practical point of view, there are many real-world problems which can be formalized as SSPs. Such problems are notably found in the area of computational biology, in which technological advances and the numerous initiatives are producing an unprecedented flood of data (Reichhardt, 1999) very much requiring the use of powerful computational tools to overcome the associated challenges (Meneses et al., 2005). Among such problems of interest from the perspective of SSPs we can cite discovering potential drug targets, creating diagnostic probes, designing primers, locating binding sites, or identifying consensus sequences just to name a few (Festa, 2007; Lanctot et al., 2003; Meneses et al., 2005).

SSPs can be classified in different ways. Following Pappalardo et al. (2013), we can consider median SSPs (in which the goal is to find a string that minimizes the sum of distances to strings in the input set), closest SSPs (aiming to identify a string that is close to strings in the input set or to fragments thereof), and farthest SSPs (whose goal is the recognition of differences between sequences). Distinguishing SSPs have also been defined (Lanctot et al., 2003) and can be regarded as a combination of closest and farthest SSPs in which the input set is partitioned in two subsets, and a string close to one of the partitions and far from the other one is sought. In a biological context, median and closest SSPs relate to the identification of consensus sequences or regions of similarity, a task of interest for creating diagnostic probes for bacterial infections or discovering potential drug targets (Festa, 2007) to give some examples. In this context, farthest SSPs relate to, for example, the existence of hosts in which the string identified is not preserved and hence should not be targeted.

Here we are specifically concerned with the Far From Most String Problem (FFMSP), a problem which can be roughly described as identifying a string whose distance is above a certain predefined threshold for as many as possible strings in the input set. A formal definition of the problem is provided in Section 2.1. As pointed out in Boucher et al. (2013), biological sequence data is subject to frequent random mutations and errors and it can thus be problematic to force the solution string to fit the entire input set.

In this sense, the FFMSP provides an interesting tradeoff since it tries to maximize the number of strings for which the distance constraint is fulfilled, without thus requiring that this is done for all strings in the input set. Not surprisingly, this turns out to be a task of formidable difficulty (a summary of complexity results is provided in Section 2.2) so the use of complete methods (exact or approximate) is often out of the question. Heuristic methods are therefore required to solve this problem. See Section 2.3 for an overview of related work in this direction.

We propose here a memetic approach to the FFMSP. Memetic algorithms (MAs) (Moscato, 1989) are a broad class of metaheuristics that try to blend together ideas from different optimization techniques, orchestrating the interplay between global population-based search and local search components – see Neri et al. (2012); Neri and Cotta (2012) for recent treatments of MAs; we also refer to Ong et al. (2010) for an overview of the broad area of memetic computing. Our MA is described in detail in Section 3. We have conducted an extensive empirical evaluation of different variants of this technique and –as shown in Section 4– it compares favorably to previous approaches in the literature.

2. The Far From Most String Problem

In this section we formally state the FFMSP and analyze its complexity. We then go on to summarize relevant related work.

2.1. Problem Statement

Let Σ denote a finite set of symbols (the alphabet), and let a string s taken from such an alphabet be a finite sequence of zero or more of those symbols ($s \in \Sigma^*$). We write $|s|$ for the length of string s, $s = s_1 s_2 \dots s_m$ ($s_i \in \Sigma, 1 \leqslant i \leqslant m$) to denote string s as a sequence of m symbols (s_i is i-th symbol of s), and ϵ for the empty string.

Let s and r be two finite strings of same length taken from same alphabet. The *Hamming Distance* ($\mathcal{HD}$) between those strings is the number of positions at which their symbols are different:

$$\mathcal{HD}(s,r) = \sum_{1 \leqslant i \leqslant |s|} [s_i \neq r_i] \tag{1}$$

where $[] : \mathbb{B} \to \mathbb{N}$ is an indicator function ([true] = 1 and [false] = 0).

An instance of the FFMSP is given by a triple (Σ, S, d), where Σ is a finite alphabet of symbols, $S = \{S^1, S^2, \ldots, S^n\}$ is a finite set of $n > 1$ strings taken from Σ ($S^i \in \Sigma^m, 1 \leqslant i \leqslant n$), all of them with same length (m), and $1 \leqslant d \leqslant m$ is an integer standing for the *distance threshold* of the problem.

A candidate solution x for the problem is a string of m symbols taken from Σ ($x \in \Sigma^m$), and we say that x is far from string $S^i \in S$ if $\mathcal{HD}(x, S^i) \geqslant d$ (if $\mathcal{HD}(x, S^i) < d$ we say that x is near S^i). The objective function f for a candidate solution to the problem x is the number of strings in S that are far from x:

$$f(x) = \sum_{S^i \in S} [\mathcal{HD}(x, S^i) \geqslant d] \tag{2}$$

and the FFMSP consists of maximizing such an objective function.

Note that the problem is trivial if $n < |\Sigma|$ as, in this case, a string x that is far from all strings in S can be easily constructed by taking for x_i ($1 \leqslant i \leqslant |x|$) a symbol that is not in position i for any string in S.

2.2. Computational Complexity of the FFMSP

The FFMSP is NP-hard in general by reduction from Independent Set (cf. Lanctot et al. (2003)). The proof is provided for $|\Sigma| = 3$ and can be generalized to $|\Sigma| \geqslant 4$ with some changes. Of course, NP-hardness is a worst-case result arising from a unidimensional analysis. A more careful multidimensional analysis may indicate the existence of a structural *parameter* whose value determines whether the problem's resolution is tractable or not. Such an analysis is the subject of parameterized complexity (see Fellows (2002) for a general overview). Therein, the notion of tractability is captured by class FPT (*fixed-parameter tractable*) (Downey and Fellows, 1995a), comprising parameterized problems $\langle \text{A},k \rangle$ for which a solution is computed in time $O(f(k)n^c)$ where $n = |A|$, k is the parameter, f is an arbitrary function of k only, and c is a constant independent of k. While the FFMSP has not been directly addressed from the point of view of parameterized complexity, the result in Lanctot et al. (2003) can be generalized to this domain since the reduction used is a parameterized reduction (for most purposes, a reduction between parameterized problems $\langle \text{A},k \rangle$ and $\langle \text{B},k' \rangle$ is termed a *parameterized reduction* if $B = \Phi(A, k)$, with Φ being computable in time $O(g(k)n^c)$, and $k' = h(k)$, where g, h are arbitrary functions of k and c is a constant independent of k (Downey and Fellows, 1995a)). More precisely, the reduction in Lanctot et al. (2003) is parameterized when the number of strings k that must be far from the constructed string is taken as a parameter of FFMSP and the number k' of vertices is the parameter in the Independent Set problem. Unfortunately, Independent Set is W[1]-complete for this parameterization (Downey and Fellows, 1995b) so the FFMSP turns out to be W[1]- hard. This means that –unlike FPT problems– for a fixed k there only exist algorithms running in time $O(f(k)n^{g(k)})$ with $g(k) \to \infty$ (Abrahamson et al., 1995).

Having established the general hardness of the FFMSP, attention can now be turned to its approximability. In this sense, it is known that W[1]-hard problems have no fully polynomial-time approximation scheme unless W[1]=FPT (Cai and Chen, 1997). The bad news does not end there since it has been also shown that Independent Set does not admit a PTAS (Arora et al., 1998), and hence neither does the FFMSP. As a matter of fact, Independent Set is known to be APX-hard –i.e. for some $\epsilon > 0$ finding a $(1+\epsilon)$-approximation is NP-hard– in general and even for some restricted cases (Papadimitriou and Yannakakis, 1991). Currently, it is known to be inapproximable within a factor of $n/2^{(\log n)^{3/4+\epsilon}}$ unless NP $\subseteq$ BPTIME$(2^{(\log n)^{O(1)}})$ (Khot and Ponnuswami, 2006). Note, finally, that APX-hardness of FFMSP has been also established in the case $|\Sigma| = 2$ following a result of Boucher et al. (2013).

2.3. Related Work

The FFMSP has attracted a lot of interest from the perspective of heuristics due to the aforementioned hardness results. One of the first heuristic attacks on the FFMSP was done by Meneses et al. (2005). They proposed a base heuristic which greedily picked symbols to construct a string so as to maximize the number of strings whose distance was above the threshold, and augmented it with a local search procedure based on two-exchanges (which provided an additional 1-5% improvement of the results). Later on, Festa (2007) proposed a *greedy randomized adaptive search procedure* (GRASP) (Feo and Resende, 1995) approach. GRASP tries to modulate the myopic behavior of greedy techniques by making randomized choices from among the elements of a restricted candidate list (comprising the best elements according to their greedy value). A local search component analogous to that in Meneses et al. (2005) was used. Overall, this approach provided improvements of 10-16% using a similar computational time.

The GRASP in Festa (2007) has been benchmarked in several subsequent publications. Mousavi (2010) proposed a *beam search* (BS) approach based on a heuristic objective function measuring the likelihood of candidate solutions to lead to better solutions with as few changes as possible. This BS approach was augmented with local search as well, and was shown to outperform GRASP in over 80% of test instances. The same objective function was later used within the GRASP approach itself (Mousavi et al., 2012), leading to huge improvements in solution quality. A genetic algorithm (GA) was also proposed by Festa and Pardalos (2012). The main feature of this GA was the use of a diversity-management policy by which the population was divided into three tiers: the first (tier A) comprising the best solutions in the previous generation, the second (tier B) comprising the results of applying uniform crossover to a parent from tier A and another parent of tiers B/C, and the third (tier C) comprising randomly generated solutions. This GA provided average improvements of 54-257% over the previous GRASP, albeit at a higher computational cost. Finally, Ferone et al. (2013) proposed a *variable neighborhood search* (VNS) metaheuristic and its use both as a stand-alone algorithm or within GRASP (as a local-search component). Also, *path relinking* is used as an intensification procedure, either in GRASP, VNS or hybrids thereof. The GRASP+VNS+PR hybrid was shown to provide the best results (sometimes with an up to a 5-fold increase with respect to pure GRASP), although GRASP+PR offered better performance in a scenario of limited computational budget.

3. A Memetic Algorithm for the FFMSP

As anticipated in Section 1, an MA is a metaheuristic approach based on the synergistic combination of local-search and population-based components. This is commonly achieved via the integration of an evolutionary algorithm (EA) and some local-search component(s), or in general via the intelligent interplay of different problem-specific algorithms (Moscato, 1999; Moscato and Cotta, 2010). In this sense, we have adopted an integrative perspective (cf. Puchinger and Raidl (2005)), using an EA skeleton to which several complex algorithmic add-ons are plugged. Figure 1 provides a pseudocode for the proposed memetic algorithm. This MA maintains a population of solutions to the problem (parameter *popSize* corresponds to the size of this population) that are firstly initialized and then recombined (with crossover probability p_X), mutated (with mutation probability p_m) and improved (by means of a local search procedure) until the allowed execution time is reached. At the end, the best solution found is returned.

Memetic Algorithm

```
function MA($I = (\Sigma, S = \{S^1, S^2, \ldots, S^n\}, d), \alpha, popSize, p_X, p_m$)
  for $i := 1$ to $popSize$ do
    $pop_i$ := GRASP($\alpha, I$)
    Evaluate($pop_i$)
  end for
  while allowed runtime not exceeded do
    if recombination is performed($p_X$) then
      $parent_1$ := Select($pop$)
      $parent_2$ := Select($pop$)
      $offspring$ := Recombine($parent_1$, $parent_2$)
    else
      $offspring$ := Select($pop$)
    end if
    $offspring$ := Mutate($p_m$, $offspring$)
    $offspring$ := Local Search($offspring, I$)
    Evaluate($offspring$)
    $pop$ := Replace($pop$, $offspring$)
  end while
  return best solution found
end function
```

Figure 1: Memetic Algorithm for FFMSP.

Since the problem does not pose constraints on the construction of solutions, these can be naturally represented as strings of length m over alphabet Σ. Although there exists a well-defined objective function f to be maximized, corresponding to the number of strings in the instance that are far from the current solution (see Equation 2), we have instead considered the heuristic function proposed in Mousavi et al. (2012) as it reduces the number of local optima of the resulting search landscape. This heuristic (which we henceforth refer to as h) evaluates a solution taking into account its objective value but also the *likelihood* of the solution to lead to better solutions with as few local moves as possible. One important property of h is that $f(s) > f(s')$ implies $h(s) > h(s')$. Details on the rationale of its design can be found in Mousavi

Heuristic Function

```
function $h(s, I = (\Sigma, S = \{S^1, S^2, \ldots, S^n\}, d))$
  $m$ is $|S^i|, \ \forall i \,.\, 1 \leqslant i \leqslant n$
  $near := 0$
  for $i := 1$ to $n$ do
    $d_i := \mathcal{HD}(s, S^i)$
    $c_i := m - d_i$
    if $d_i < d$ then
      $near := near + 1$
    end if
  end for
  $f := n - near$
  if $near = 0$ then
    $GpC := 0$
  else
    $sumGpC := 0$
    for $i := 1$ to $n, \ d_i < d$ do
      $g_i := 1$
      for $j := 1$ to $n, \ i \neq j$ do
        $sumP := 0$
        for $c := c_j$ to $c_i$ do
          $sumP := sumP + T[c_i, c]/|\Sigma|^{c_i}$
        end for
        $g_i := g_i + sumP$
      end for
      $sumGpC := sumGpC + g_i/c_i$
    end for
    $GpC := sumGpC/near$
  end if
  return $(n + 1) \times f + GpC$
end function
```

Figure 2: Heuristic function for FFMSP.

et al. (2012), and we reproduce its pseudo-code in Figure 2 for completeness of description of our proposal. In this pseudo-code, T is a bi-variable function from $\mathbb{N}_0 \times \mathbb{Z}$ to $\mathbb{N}_0$ that can be precomputed as follows:

$$\begin{array}{lcl} T(0,0) & = & 1 \\ T(0,k) & = & 0, \text{ if } k \neq 0 \\ T(L,k) & = & T(L-1,k-1) + (|\Sigma|-2)T(L-1,k) + T(L-1,k+1), \text{ if } L > 0 \end{array}$$

and, as a result, h can be calculated in $O(nm + n^2)$, having paid a one-off time cost of $O(m^2)$ to initialize a bi-dimensional array corresponding to T.

Initialization of individuals in the population was done by means of a Greedy Randomized Adaptive Search Procedure (GRASP) similar to the one proposed in Ferone et al. (2013). Pseudo-code for this procedure is shown in Figure 3, where $V_j(c)$ stands for the total number of occurrences of symbol c in position j in any of the strings in the given instance, and V_j^{min} (V_j^{max}) is the number of occurrences of the least (most) frequent symbol at position j. As can be seen, the procedure constructs a solution by adding a symbol to the string on each iteration. This symbol is chosen uniformly at random from a restricted candidate list (RCL) that includes candidate symbols. A parameter α ($0 \leqslant \alpha \leqslant 1$) is introduced as a generalization of the original GRASP component of the algorithm by Ferone et al. (2013), in order to control the degree of greediness used to generate the solution: smaller values for this parameter imply the generation of greedier solutions and larger values introduce more randomness. For instance, if α is 0, so is β, and then only the least frequent symbol is included in the RCL (this would correspond to a greedy selection strategy where the least frequent symbol for each position would be selected). For greater values of α, more symbols can be included in the RCL (all of them when $\beta = 1$), and then selection is more random.

Mutation of individuals has been implemented by substituting (with probability p_m) symbols in current string with a random symbol from Σ, and recombination has been performed by means of an intelligent operator that uses path relinking (PR) (Glover et al., 2000) in order to attain a sensible recombination of information from parents. PR finds new solutions by exploring paths that connect high quality solutions in neighborhood space. This is achieved by starting a search in one of those solutions (the initiating one) and performing local moves that lead to another solution (the guiding one). Figure 4 shows the pseudo-code of this procedure for combining two parent solutions (p^1 and p^2) for the FFMSP, where the worse parent (according to heuristic function h) acts as the initiating solution (s) and the

GRASP Algorithm

```
function GRASP($\alpha, I = (\Sigma, S = \{S^1, S^2, \dots, S^n\}, d)$)
  $m$ is $|S^i|, \ \forall i \,.\, 1 \leqslant i \leqslant n$
  $V_j(c) := \sum_{S^i \in S, \ S^i_j = c} 1, \ \forall c \in \Sigma, \ \forall j \,.\, 1 \leqslant j \leqslant m$
  $V_j^{min} := \min\{V_j(c) \mid c \in \Sigma\}, \ \forall j \,.\, 1 \leqslant j \leqslant m$
  $V_j^{max} := \max\{V_j(c) \mid c \in \Sigma\}, \ \forall j \,.\, 1 \leqslant j \leqslant m$
  $s := \epsilon$
  $\beta :=$ Random[0,$\alpha$]
  for $j := 1$ to $m$ do
    $RCL_j := \emptyset$
    $\mu := V_j^{min} + \beta(V_j^{max} - V_j^{min})$
    for $c \in \Sigma, \ V_j(c) \leqslant \mu$ do
      $RCL_j := RCL_j \cup \{c\}$
    end for
    $s := s$ Random($RCL_j$)
  end for
  return($s$)
end function
```

Figure 3: GRASP Algorithm for FFMSP.

better parent as the guiding one (s^*). As it can be seen, differences between both solutions (as their positions and symbols in s^*) are firstly computed (set Δ), thus defining the set of local moves to be done. Then, a path from the initiating solution towards the guiding one is generated by incorporating into the former, on each move, the component from the set Δ that leads to a better intermediate solution (the incorporation of the selected component (i^*, s^*) is done by replacing the symbol at position i^* with s^* in the current solution). This procedure is repeated until all moves have been performed (i.e., the guiding solution has been reached) and the best solution found along the path (s^*) is returned as a child.

Finally, improvement of solutions is done through a sequential hill climbing local search procedure on the search space induced by h heuristic function (see Figure 5), where a local move corresponds to modifying a single symbol in the solution with another one from the alphabet.

Path Relinking Crossover Algorithm

```
function PATH RELINKING($p^1, p^2, I = (\Sigma, S = \{S^1, S^2, \ldots, S^n\}, d)$)
  $m$ is $|S^i|$, $\forall i \,.\, 1 \leqslant i \leqslant n$
  $s := \text{ARG MIN}\{h(s, I) \mid s \in \{p^1, p^2\}\}$
  $s^* := \text{ARG MAX}\{h(s, I) \mid s \in \{p^1, p^2\}\}$
  $h^* := h(s^*, I)$
  $\Delta := \{(i, s^*_i) \mid i \in \{1 \ldots m\}, s_i \neq s^*_i\}$
  while $\Delta \neq \emptyset$ do
    $(i^*, c^*) := \text{ARG MAX}\{h(s_1 s_2 \ldots s_{i-1}\, c\, s_{i+1} \ldots s_m, I) \mid (i, c) \in \Delta\}$
    $s := s_1 s_2 \ldots s_{i^*-1}\, c^*\, s_{i^*+1} \ldots s_m$
    $\Delta := \Delta - \{(i^*, c^*)\}$
    if $h(s, I) > h^*$ then
      $s^* := s$
      $h^* := h(s, I)$
    end if
  end while
  return($s^*$)
end function
```

Figure 4: Path Relinking Crossover Algorithm for FFMSP.

4. Results and Discussion

In order to evaluate the different heuristics, we have considered two benchmark sets. Instances in the first set – henceforth referred to as RandomSet – are composed of different numbers (n) of strings of the same length (m) randomly selected according to a uniform distribution from the alphabet $\Sigma = \{\mathrm{A}, \mathrm{T}, \mathrm{C}, \mathrm{G}\}$. Eighteen subsets of instances have been defined using different numbers of strings ($n \in \{100, 200\}$), string lengths ($m \in \{300, 600, 800\}$) and distance thresholds ($d \in \{0.75 \cdot m, 0.80 \cdot m, 0.85 \cdot m\}$). For each of the eighteen subsets five different instances have been generated yielding thus a total of 90 instances.

The second set – referred to as RealSet – is also composed of 90 instances with the same combination of parameters as those in RandomSet, but instead of being randomly generated, different strings in each instance have been selected from a random segment of a real genome, namely

Local Search Algorithm

```
function LOCAL SEARCH($s, I = (\Sigma, S = \{S^1, S^2, \ldots, S^n\}, d)$)
    $m$ is $|S^i|$, $\forall i \,.\, 1 \leqslant i \leqslant n$
    $improvement$ := true
    while $improvement$ do
        $improvement$ := false
        for $j := 1$ to $m$ do
            for $c \in \Sigma,\ c \neq s_j$ do
                $s' := s_1 s_2 \ldots s_{j-1}\, c\, s_{j+1} \ldots s_m$
                if $h(s', I) > h(s, I)$ then
                    $s := s'$
                    $improvement$ := true
                end if
            end for
        end for
    end while
    return($s$)
end function
```

Figure 5: Local Search Algorithm for FFMSP.

Phytophthora Ramorum's genome (Tyler et al., 2006) (available at `http://genome.jgi.doe.gov/Phyra1_1/Phyra1_1.download.html`).

Regarding the programs used in the experiments, all algorithms were coded in C and compiled using gcc under Linux (the hybrid GRASP+VNS+PR by Ferone et al. (2013) was available from their authors and we coded remaining algorithms). With the purpose of making a fair comparison, all executions for different algorithms were performed on the same machine (HP Proliant SL170s computer, Intel Xeon X5660 2.8 GHz processor, 8GB RAM, CentOS 5.5 operating system) with a time limit of 600 seconds per execution. In all experiments, crossover probability was set to $p_X = 0.9$, mutation probability was $p_m = 1/m$ (m stands for the length of strings in the instance), population size was *popSize* $= 100$ and binary tournament was used to perform selection.

In order to report results in tables and figures uniformly, we calculate the relative percentage distance (RPD) of the solution obtained by each algorithm from the best result for the corresponding instance obtained by any of the compared algorithms (defined as $(sol_{\text{best}} - sol) \times 100/sol_{\text{best}}$), and provide statistical values for those distances.

The statistical analysis of the results of the experiments was done using software available on the web page of Research Group Soft Computing and Intelligent Information Systems at the University of Granada (`http://sci2s.ugr.es/sicidm`). With the aim of analyzing the statistical significance of the results, we used the following methodology: firstly, we performed an Aligned Friedman Rank Test (JL Hodges Jr and Lehmann, 1962), a multiple-comparison non-parametric test that aims to detect significant differences between the behavior of two or more algorithms and then ranks them from the best to worst. If, as a result of this test, the null hypothesis stating equality of rankings between the populations is rejected, we proceed to post-hoc procedures in order to compare the control algorithm (the best performing one) with the remaining algorithms. The post-hoc procedures that we have considered are Bonferroni–Dunn's (Dunn, 1961), Holm's (Holm, 1979), Hochberg's(Hochberg, 1988), Hommel's (Hommel, 1988), Holland's (Holland and Copenhaver, 1987), Rom's (Rom, 1990), Finner's (Finner, 1993) and Li's (Li, 2008) procedures. Finally, we report adjusted p-values for different post-hoc procedures (the smaller overall significance level at which the particular null hypothesis stating equality between the distributions obtained by the control algorithm and the other compared algorithm would be rejected).

4.1. Parameterization and Sensitivity Analysis

So as to analyze the influence of the different algorithmic components and parameters in the performance of the proposed MA, we firstly compared different variants of the algorithm. For these experiments, we used instances in RandomSet, and performed 10 independent executions for each algorithm and instance (thus, 50 executions per subset and algorithm).

The different components and parameters of the MA that we took into account were:

- Population initialization: we considered different values for parameter α in the GRASP algorithm (Figure 3) used during the initialization of the MA population ($\alpha \in \{0.1, 0.25, 0.5\}$). This parameter controls the greediness of GRASP, so that greater values imply less greediness. We also considered completely randomly initializing individuals.

- Recombination operator: we considered two possibilities; either using uniform crossover (UX) or using a path relinking (PR) algorithm (Figure 4).
- Local search operator: we considered the possibility of using the hill climbing operator (Figure 5) after mutation in the MA loop or not using local search at all.

In order to name different algorithms, term GRASP_p indicates using GRASP for initializing the population and setting $\alpha = p$ in this component of the algorithm. If the population was initialized randomly, the name of the algorithm instead includes the term RND. If the name includes the term PR, then path relinking was used as the crossover operator. Term UX denotes the use of uniform crossover and term HC indicates that the algorithm uses hill climbing as the local search operator. According to this convention, $\mathrm{MA}_{\mathrm{GRASP}_{0.25}+\mathrm{PR}+\mathrm{HC}}$, for instance, is a memetic algorithm that uses path relinking and hill climbing and sets $\alpha = 0.25$ in the GRASP algorithm for initializing the population. $\mathrm{MA}_{\mathrm{RND}+\mathrm{UX}}$ corresponds to using uniform crossover, no local search and random initialization of population (note that this particular algorithm is not a memetic algorithm but a genetic algorithm).

Figure 6 shows the results of these experiments as boxplots for the RPD from best solutions of results obtained by each algorithm on each subset for different numbers of strings and string lengths when the distance threshold is $d = 0.80 \cdot m$. It can be seen that, in most of the subsets, the memetic algorithms using GRASP with path relinking and hill climbing obtain best results, and overall, a small value for α leads to better quality solutions. When $n = 200$ and $m \geqslant 600$, algorithms using GRASP with uniform crossover and no local search obtain best results. This may be due to the comparatively higher computational cost of PR and HC in these particular instances, which may consume the CPU budget faster (note at any rate that the difference is small with respect to $\mathrm{MA}_{\mathrm{GRASP}+\mathrm{HC}+\mathrm{PR}}$). This is further vindicated by the comparatively worse results of UX in combination with HC, indicating the interplay of the local improvement operator is better with PR (a more intensive recombination operator, capable of efficiently locating regions of interest) than with UX (a purely explorative, information-mixing operator). For $d = 0.85 \cdot m$, results are similar (Figure 7), but best solutions are obtained by algorithms using GRASP with uniform crossover and hill climbing when $n = 200$ and $m = 600$ (these algorithms also obtain results as good as those

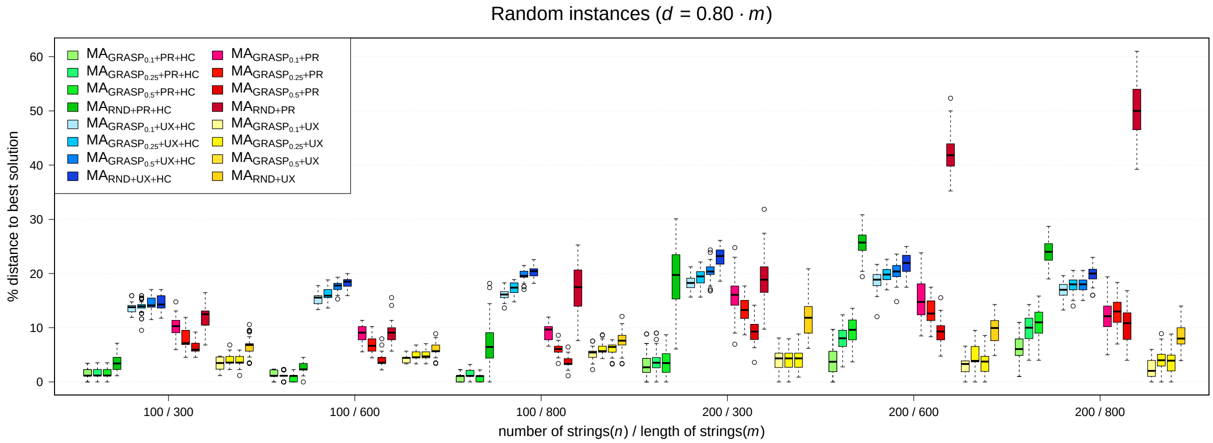

Figure 6: Box plots for relative percentage distances (RPD) from best solutions of results obtained by each memetic algorithm for instances in RandomSet and distance threshold $d = 0.80 \cdot m$. We have considered 6 datasets, each one comprising 5 different instances. Instances on each dataset are characterized by a number of strings (n) of the same length (m) and a distance threshold (d). For each combination of n/m parameters, box plots in left-to-right order correspond to algorithms in legend in top-to-bottom, left-to-right order.

obtained using path relinking when $n = 200$ and $m = 300$ or $m = 800$, but perform worse in other cases). No differences could be observed in the case of $d = 0.75 \cdot m$, as all algorithms were able to obtain optimal solutions in all runs.

To detect statistical significances in the performance of different memetic algorithms, we performed an Aligned Friedman Rank Test, with the results shown in Table 1 (distributed according to χ^2 with 15 degrees of freedom: 844.06). The p-value calculated by this test is $p = 2.44 \cdot 10^{-10}$, and this provides strong evidence for rejecting the null hypothesis that states equality of rankings between the populations. It can also be observed that MA$_{\mathrm{GRASP}_{0.1}+\mathrm{PR}+\mathrm{HC}}$ is ranked as the best performing algorithm followed by MA$_{\mathrm{GRASP}_{0.25}+\mathrm{PR}+\mathrm{HC}}$ and MA$_{\mathrm{GRASP}_{0.5}+\mathrm{PR}+\mathrm{HC}}$. As the Aligned Friedman Rank Test found statistical differences between the different algorithms, we proceeded to perform post-hoc procedures, comparing the control algorithm (MA$_{\mathrm{GRASP}_{0.1}+\mathrm{PR}+\mathrm{HC}}$) with those remaining. Table 2 shows adjusted p-values for the different procedures. The null hypothesis that states equality between the distributions obtained by the control algorithm and the other compared

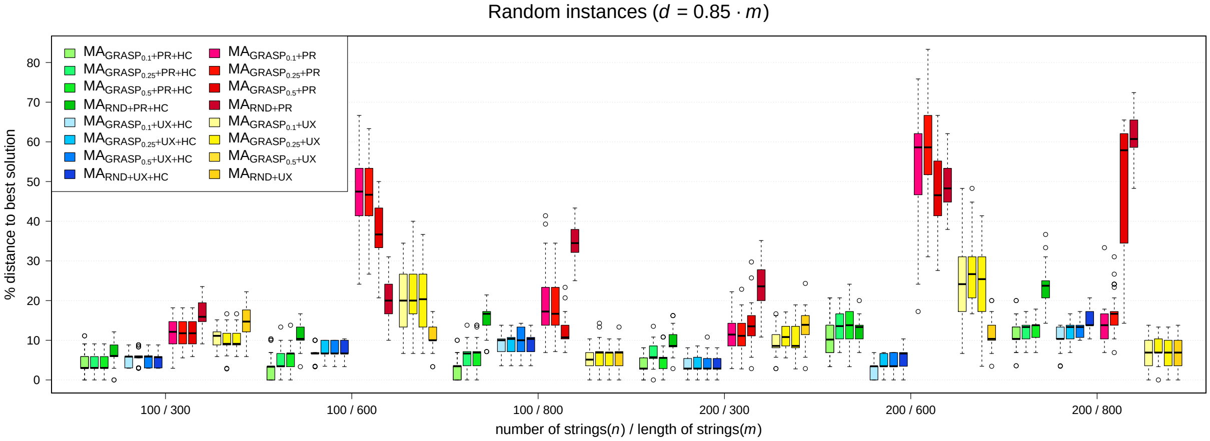

Figure 7: Box plots for relative percentage distances (RPD) from best solutions of results obtained by each memetic algorithm for instances in RandomSet and distance threshold $d = 0.85 \cdot m$. We have considered 6 datasets, each one comprising 5 different instances. Instances on each dataset are characterized by a number of strings (n) of the same length (m) and a distance threshold (d). For each combination of n/m parameters, box plots in left-to-right order correspond to algorithms in legend in top-to-bottom, left-to-right order.

algorithm was rejected by all procedures for all algorithms (results are statistically significant) except for $\text{MA}_{\text{GRASP}_{0.25}\text{+PR+HC}}$ and $\text{MA}_{\text{GRASP}_{0.5}\text{+PR+HC}}$. In the case of $\text{MA}_{\text{GRASP}_{0.5}\text{+PR+HC}}$, all procedures except for Bonferroni–Dunn's reject null hypothesis (with p-values equal to 0.02 and 0.04, thus differences are statistically significant). Finally, for $\text{MA}_{\text{GRASP}_{0.25}\text{+PR+HC}}$, none of the procedures found statistical evidence to reject the null hypothesis, thus the performance of this algorithm cannot be considered statistically different from the one of $\text{MA}_{\text{GRASP}_{0.1}\text{+PR+HC}}$.

As a result of this analysis, it can be concluded that incorporating GRASP (with an small setting for parameter α) in order to initialize the population, using path relinking as a crossover operator and performing local search by means of hill climbing yields overall best results and thus using these components in the memetic algorithm is beneficial.

4.2. Comparison to state-of-the-art algorithms

In this section we compare one of the best performing memetic algorithm ($\text{MA}_{\text{GRASP}_{0.1}\text{+PR+HC}}$ – denoted by $\text{MA}_{\text{GRASP+HC+PR}}$ henceforth for simplicity) to state-of-the-art algorithms for the FFMSP (hybrid GRASP+VNS+PR in

Table 1: Results for the Aligned Friedman Rank Test on RandomSet instances for different memetic algorithms.

position	algorithm	ranking
1st	$\mathrm{MA}_{\mathrm{GRASP}_{0.1}+\mathrm{PR}+\mathrm{HC}}$	4125.76
2nd	$\mathrm{MA}_{\mathrm{GRASP}_{0.25}+\mathrm{PR}+\mathrm{HC}}$	4482.05
3rd	$\mathrm{MA}_{\mathrm{GRASP}_{0.5}+\mathrm{PR}+\mathrm{HC}}$	4584.02
4th	$\mathrm{MA}_{\mathrm{GRASP}_{0.1}+\mathrm{UX}}$	5591.35
5th	$\mathrm{MA}_{\mathrm{GRASP}_{0.5}+\mathrm{UX}}$	5699.21
6th	$\mathrm{MA}_{\mathrm{GRASP}_{0.25}+\mathrm{UX}}$	5851.03
7th	$\mathrm{MA}_{\mathrm{RND}+\mathrm{UX}}$	6206.48
8th	$\mathrm{MA}_{\mathrm{GRASP}_{0.5}+\mathrm{PR}}$	7615.16
9th	$\mathrm{MA}_{\mathrm{RND}+\mathrm{PR}+\mathrm{HC}}$	7997.19
10th	$\mathrm{MA}_{\mathrm{GRASP}_{0.1}+\mathrm{UX}+\mathrm{HC}}$	8206.66
11th	$\mathrm{MA}_{\mathrm{GRASP}_{0.25}+\mathrm{UX}+\mathrm{HC}}$	8473.52
12th	$\mathrm{MA}_{\mathrm{GRASP}_{0.5}+\mathrm{UX}+\mathrm{HC}}$	8504.02
13th	$\mathrm{MA}_{\mathrm{GRASP}_{0.25}+\mathrm{PR}}$	8569.03
14th	$\mathrm{MA}_{\mathrm{RND}+\mathrm{UX}+\mathrm{HC}}$	8722.04
15th	$\mathrm{MA}_{\mathrm{GRASP}_{0.1}+\mathrm{PR}}$	9446.67
16th	$\mathrm{MA}_{\mathrm{RND}+\mathrm{PR}}$	11133.78

Table 2: Adjusted p-values for $N \times 1$ comparisons of control algorithm ($\mathrm{MA}_{\mathrm{GRASP}_{0.1}+\mathrm{PR}+\mathrm{HC}}$) with different memetic algorithms for the Aligned Friedman Rank Test on RandomSet instances.

algorithm	Bonf	Holm	Hoch	Homm	Holl	Rom	Finn	Li
$\mathrm{MA}_{\mathrm{GRASP}_{0.25}+\mathrm{PR}+\mathrm{HC}}$	1.00E0	7.00E-2	7.00E-2	7.00E-2	7.00E-2	7.00E-2	7.00E-2	7.00E-2
$\mathrm{MA}_{\mathrm{GRASP}_{0.5}+\mathrm{PR}+\mathrm{HC}}$	2.90E-1	4.00E-2	4.00E-2	4.00E-2	4.00E-2	4.00E-2	2.00E-2	2.00E-2
$\mathrm{MA}_{\mathrm{GRASP}_{0.1}+\mathrm{UX}}$	1.13E-12	2.25E-13	2.25E-13	2.25E-13	2.25E-13	2.25E-13	8.66E-14	8.06E-14
$\mathrm{MA}_{\mathrm{GRASP}_{0.5}+\mathrm{UX}}$	1.47E-14	3.93E-15	3.93E-15	3.93E-15	4.00E-15	3.74E-15	1.22E-15	1.05E-15
$\mathrm{MA}_{\mathrm{GRASP}_{0.25}+\mathrm{UX}}$	1.98E-17	6.61E-18	6.61E-18	6.61E-18	0.0	6.28E-18	0.0	1.42E-18
$\mathrm{MA}_{\mathrm{RND}+\mathrm{UX}}$	3.70E-25	1.48E-25	1.48E-25	1.48E-25	0.0	1.41E-25	0.0	2.65E-26
$\mathrm{MA}_{\mathrm{GRASP}_{0.5}+\mathrm{PR}}$	9.50E-70	4.43E-70	4.43E-70	4.43E-70	0.0	4.21E-70	0.0	6.80E-71
$\mathrm{MA}_{\mathrm{RND}+\mathrm{PR}+\mathrm{HC}}$	1.08E-85	5.74E-86	5.74E-86	5.74E-86	0.0	5.46E-86	0.0	7.71E-87
$\mathrm{MA}_{\mathrm{GRASP}_{0.1}+\mathrm{UX}+\mathrm{HC}}$	3.89E-95	2.34E-95	2.34E-95	2.34E-95	0.0	2.22E-95	0.0	2.79E-96
$\mathrm{MA}_{\mathrm{GRASP}_{0.25}+\mathrm{UX}+\mathrm{HC}}$	6.99E-108	4.66E-108	4.66E-108	4.66E-108	0.0	4.43E-108	0.0	5.01E-109
$\mathrm{MA}_{\mathrm{GRASP}_{0.5}+\mathrm{UX}+\mathrm{HC}}$	2.17E-109	1.59E-109	1.59E-109	1.59E-109	0.0	1.51E-109	0.0	1.55E-110
$\mathrm{MA}_{\mathrm{GRASP}_{0.25}+\mathrm{PR}}$	1.22E-112	9.78E-113	9.78E-113	9.78E-113	0.0	9.30E-113	0.0	8.76E-114
$\mathrm{MA}_{\mathrm{RND}+\mathrm{UX}+\mathrm{HC}}$	1.78E-120	1.55E-120	1.55E-120	1.55E-120	0.0	1.47E-120	0.0	1.28E-121
$\mathrm{MA}_{\mathrm{GRASP}_{0.1}+\mathrm{PR}}$	3.58E-161	3.34E-161	3.34E-161	3.34E-161	0.0	3.17E-161	0.0	2.56E-162
$\mathrm{MA}_{\mathrm{RND}+\mathrm{PR}}$	6.60E-279	6.60E-279	6.60E-279	6.60E-279	0.0	6.27E-279	0.0	4.73E-280

Ferone et al. (2013) – that we denote by $\mathrm{GRASP}_{\mathrm{FFR}}$– and GRASP algorithm in Mousavi et al. (2012) – denoted by $\mathrm{GRASP}_{\mathrm{Mou}}$). For $\mathrm{GRASP}_{\mathrm{FFR}}$ we set corresponding parameters as indicated by the authors of that algorithm, and for $\text{GRASP}_{\text{Mou}}$ we set $\gamma = 0.95$, after observing that using this setting provides better results than using the one suggested by the authors of this algorithm. We firstly made a comparison on instances in RandomSet, for which 20 independent runs on each of the instances were executed by each of the algorithms (thus, 100 executions per subset and algorithm). Results are shown numerically in Table 3 as the mean objective value obtained by each of the algorithms on each subset, together with the mean and standard deviation for RPD for best solutions and mean improvement percentage of results obtained by $\text{MA}_{\text{GRASP+HC+PR}}$ with respect to those obtained by $\text{GRASP}_{\text{FFR}}$ and $\text{GRASP}_{\text{Mou}}$.

Table 3: Results obtained by $\text{MA}_{\text{GRASP+HC+PR}}$, $\text{GRASP}_{\text{FFR}}$ and $\text{GRASP}_{\text{Mou}}$ for instances in RandomSet. We have considered 18 datasets, each one comprising 5 different instances. Instances on each dataset are characterized by a number of strings (n) of the same length (m) and a distance threshold (d). Table shows mean solution obtained by each algorithm (sol), along with statistical values (mean (μ) and standard deviation (σ)) for relative percentage distance (RPD) from best solutions of results obtained by each algorithm. For $\text{GRASP}_{\text{FFR}}$ and $\text{GRASP}_{\text{Mou}}$ algorithms, mean improvement percentage of solutions (imp.%) obtained by $\text{MA}_{\text{GRASP+HC+PR}}$ with respect to the ones obtained by these algorithms is also shown.

			$\text{MA}_{\text{GRASP+HC+PR}}$		$\text{GRASP}_{\text{FFR}}$			$\text{GRASP}_{\text{Mou}}$		
d	n	m	sol.	RPD $\mu \pm \sigma$	sol.	RPD $\mu \pm \sigma$	imp.%	sol.	RPD $\mu \pm \sigma$	imp.%
$0.75 \cdot m$	100	300	100.00	0.00 ± 0.00	100.00	0.00 ± 0.00	0.00	100.00	0.00 ± 0.00	0.00
	100	600	100.00	0.00 ± 0.00	100.00	0.00 ± 0.00	0.00	100.00	0.00 ± 0.00	0.00
	100	800	100.00	0.00 ± 0.00	100.00	0.00 ± 0.00	0.00	100.00	0.00 ± 0.00	0.00
	200	300	200.00	0.00 ± 0.00	200.00	0.00 ± 0.00	0.00	200.00	0.00 ± 0.00	0.00
	200	600	200.00	0.00 ± 0.00	200.00	0.00 ± 0.00	0.00	200.00	0.00 ± 0.00	0.00
	200	800	200.00	0.00 ± 0.00	200.00	0.00 ± 0.00	0.00	200.00	0.00 ± 0.00	0.00
$0.80 \cdot m$	100	300	84.82	1.14 ± 0.75	80.78	5.85 ± 1.51	4.71	70.99	17.27 ± 1.65	16.13
	100	600	87.08	1.27 ± 0.72	79.12	10.30 ± 1.47	9.03	70.83	19.69 ± 1.06	18.42
	100	800	89.90	1.21 ± 0.81	79.52	12.60 ± 1.75	11.40	71.08	21.88 ± 0.97	20.68
	200	300	109.58	2.34 ± 1.49	105.85	5.66 ± 2.26	3.32	83.04	25.98 ± 1.69	23.64
	200	600	101.23	2.86 ± 1.91	88.95	14.61 ± 2.60	11.76	80.90	22.29 ± 2.65	19.44
	200	800	93.82	4.26 ± 2.28	80.09	18.27 ± 3.26	14.01	79.77	18.59 ± 1.97	14.33
$0.85 \cdot m$	100	300	32.58	3.61 ± 2.28	18.41	45.60 ± 5.55	41.99	30.10	10.95 ± 1.61	7.34
	100	600	28.76	4.76 ± 3.12	4.89	83.82 ± 3.38	79.06	25.36	16.01 ± 2.12	11.25
	100	800	27.96	2.90 ± 2.61	2.58	91.03 ± 3.25	88.13	24.33	15.51 ± 1.70	12.61
	200	300	34.49	3.64 ± 2.29	14.85	58.57 ± 4.34	54.93	32.69	8.62 ± 2.80	4.98
	200	600	26.17	6.54 ± 4.39	2.26	91.93 ± 2.60	85.39	25.54	8.76 ± 2.84	2.22
	200	800	25.61	4.42 ± 2.95	0.60	97.76 ± 1.83	93.35	23.71	11.51 ± 2.97	7.10

For instances with $d = 0.75 \cdot m$, all the algorithms were always able to find optimal solutions on all instances. In the case of $d = 0.80 \cdot m$, the best results were obtained by $\text{MA}_{\text{GRASP+HC+PR}}$ followed by $\text{GRASP}_{\text{FFR}}$. The average improvement percentage of $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{FFR}}$ ranges from 3.32% for $n = 200$ and $m = 300$ to 14.01% for $n = 200$ and $m = 800$. The mean average improvement of $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{FFR}}$ is 9.04% whereas it is 18.77% for $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{Mou}}$. Figure 8 shows the results of these experiments as boxplots for the RPD from the best solutions of results obtained by each algorithm on each subset. As can be seen, the average improvement percentage of $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{FFR}}$ increases with m, both for $n = 100$ and $n = 200$. In the case of $\text{GRASP}_{\text{Mou}}$, this percentage is rather stable for $n = 100$ and seems to decrease gently for increasing m with $n = 200$, but is always larger than for $\text{GRASP}_{\text{FFR}}$. For instances with $d = 0.85 \cdot m$, the best results are again obtained by $\text{MA}_{\text{GRASP+HC+PR}}$ followed by $\text{GRASP}_{\text{Mou}}$. The average improvement percentage of $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{Mou}}$ ranges from 2.22% for $n = 200$ and $m = 600$ to 12.61% for $n = 100$ and $m = 800$. The mean average improvement of $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{Mou}}$ is 7.58% whereas for $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{FFR}}$ it is 73.81%.

Table 4 shows statistical results for an Aligned Friedman Rank Test (distributed according to χ^2 with 2 degrees of freedom: 1327.33). The p-value calculated by this test is 0, and this indicates strong evidence for rejecting the null hypothesis that states equality of rankings between the populations. We can also observe that $\text{MA}_{\text{GRASP+HC+PR}}$ is ranked as the best performing algorithm followed by $\text{GRASP}_{\text{Mou}}$ and $\text{GRASP}_{\text{FFR}}$.

Table 4: Results for the Aligned Friedman Rank Test on RANDOMSET instances for $\text{MA}_{\text{GRASP+HC+PR}}$ and different state-of-the-art algorithms.

position	algorithm	ranking
1^{st}	$\text{MA}_{\text{GRASP+HC+PR}}$	1471.59
2^{nd}	$\text{GRASP}_{\text{Mou}}$	2998.60
3^{rd}	$\text{GRASP}_{\text{FFR}}$	3631.29

Since the Aligned Friedman Rank Test found statistical differences between the different algorithms, we proceeded to perform post-hoc procedures in order to compare the control algorithm ($\text{MA}_{\text{GRASP+HC+PR}}$) with other algorithms ($\text{GRASP}_{\text{Mou}}$ and $\text{GRASP}_{\text{FFR}}$). In all cases and with all procedures, the null hypothesis that states equality between the distributions obtained by the control algorithm and the other compared algorithm was rejected,

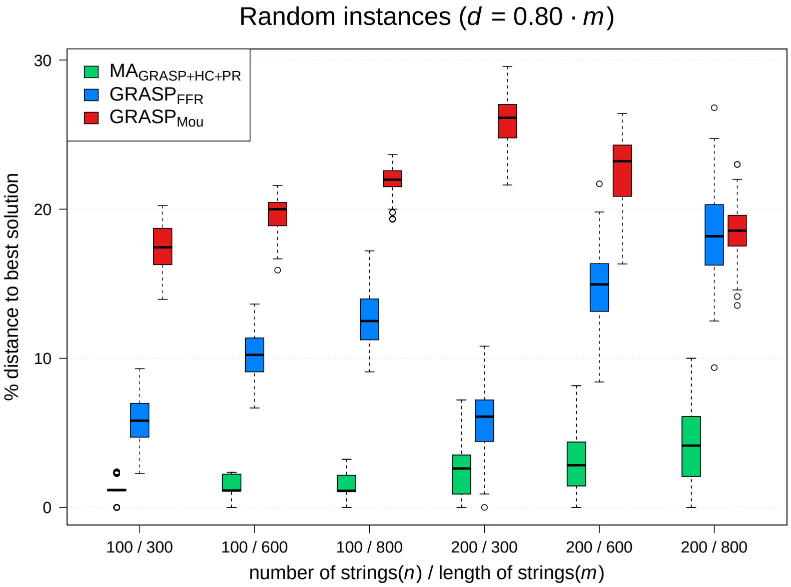

Figure 8: Box plots for relative percentage distances (RPD) from best solutions of results obtained by state-of-the-art algorithms for the FFMSP ($\mathrm{GRASP}_{\mathrm{FFR}}$ and $\mathrm{GRASP}_{\mathrm{Mou}}$) and $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ for instances in RandomSet and distance threshold $d = 0.80 \cdot m$. Instances on each dataset are characterized by a number of strings (n) of the same length (m) and a distance threshold (d). For each combination of n/m parameters, box plots in left-to-right order correspond to algorithms in legend in top-to-bottom order.

thus there are statistical differences between the control algorithm and those remaining. The adjusted p-values for different post-hoc procedures are shown in Table 5. These p-values are 0 or very close to 0 and hence the results are considered statistically significant.

Table 5: Adjusted p-values for $N \times 1$ comparisons of control algorithm ($\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$) with different state-of-the-art algorithms for the Aligned Friedman Rank Test on RandomSet instances.

algorithm	Bonf	Holm	Hoch	Homm	Holl	Rom	Finn	Li
$\mathrm{GRASP}_{\mathrm{FFR}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\mathrm{GRASP}_{\mathrm{Mou}}$	1.73E-189	8.66E-190	8.66E-190	8.66E-190	0.0	8.66E-190	0.0	8.66E-190

For instances in RealSet we also executed 20 independent runs by each of the algorithms on each instance (thus, 100 executions per subset and al-

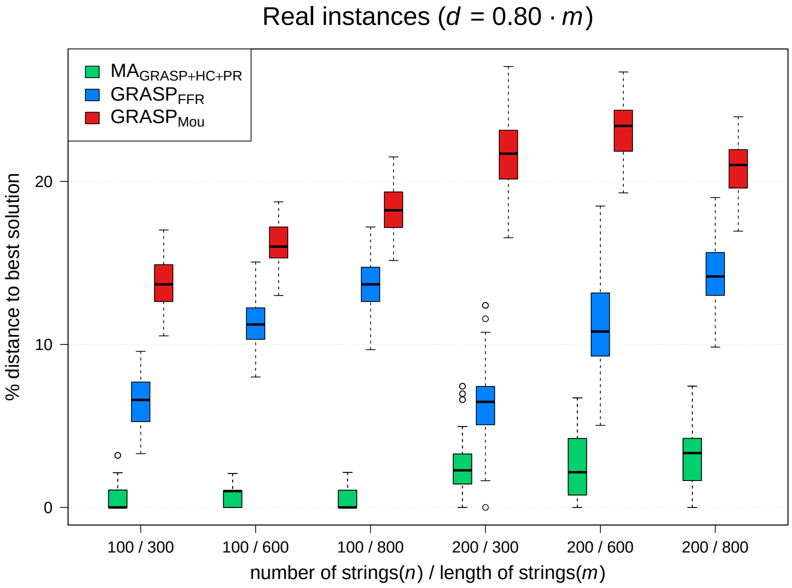

Figure 9: Box plots for relative percentage distances (RPD) from best solutions of results obtained by state-of-the-art algorithms for the FFMSP ($\mathrm{GRASP}_{\mathrm{FFR}}$ and $\mathrm{GRASP}_{\mathrm{Mou}}$) and $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ for instances in RealSet and distance threshold $d = 0.80 \cdot m$. Instances on each dataset are characterized by a number of strings (n) of the same length (m) and a distance threshold (d). For each combination of n/m parameters, box plots in left-to-right order correspond to algorithms in legend in top-to-bottom order.

gorithm). The results are also shown numerically in Table 6 as the mean objective value obtained by each of the algorithms on each subset, together with the mean and standard deviation for RPD for best solutions and mean improvement percentage of solutions obtained by $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ with respect to those obtained by $\mathrm{GRASP}_{\mathrm{FFR}}$ and $\mathrm{GRASP}_{\mathrm{Mou}}$.

The results are very similar to those on RandomSet instances. For instances with $d = 0.75 \cdot m$, all algorithms were always able to find optimal solutions on all instances. In the case of $d = 0.80 \cdot m$, the best results were obtained by $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ followed by $\mathrm{GRASP}_{\mathrm{FFR}}$. The average improvement percentage of $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ over $\mathrm{GRASP}_{\mathrm{FFR}}$ ranges from 4.06% for $n = 200$ and $m = 300$ to 13.14% for $n = 100$ and $m = 800$. The mean average improvement of $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ over $\mathrm{GRASP}_{\mathrm{FFR}}$ is 8.94% whereas for $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ over $\mathrm{GRASP}_{\mathrm{Mou}}$ it is 17.36%. Figure 9 shows the results of these experiments as boxplots for the RPD from the best solutions of results obtained by each algorithm on each subset. For instances with $d = 0.85 \cdot m$, the best results are again obtained by $\text{MA}_{\text{GRASP+HC+PR}}$ followed by $\text{GRASP}_{\text{Mou}}$. The average improvement percentage of $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{Mou}}$ ranges from 4.82% for $n = 200$ and $m = 600$ to 11.05% for $n = 100$ and $m = 600$. The mean average improvement of $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{Mou}}$ is 7.77% whereas for $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{FFR}}$ it is 59.25%. The trends are analogous as are those found in RandomSet, confirming the sustained advantage of $\text{MA}_{\text{GRASP+HC+PR}}$ and the relative performance of $\text{GRASP}_{\text{Mou}}$ and $\text{GRASP}_{\text{FFR}}$.

Table 7 shows statistical results for the Aligned Friedman Rank Test (distributed according to χ^2 with 2 degrees of freedom: 1324.00). The p-value calculated by this test is 0, and this indicates strong evidence for rejecting the null hypothesis that states equality of rankings between the populations. We can also observe that $\text{MA}_{\text{GRASP+HC+PR}}$ is ranked as the best performing algorithm followed by $\text{GRASP}_{\text{Mou}}$ and $\text{GRASP}_{\text{FFR}}$. Due to statistical differences between the different algorithms, we proceeded to perform post-hoc procedures in order to compare the control algorithm ($\text{MA}_{\text{GRASP+HC+PR}}$) with other algorithms ($\text{GRASP}_{\text{Mou}}$ and $\text{GRASP}_{\text{FFR}}$). In all cases and with all procedures, the null hypothesis that states equality between the distributions obtained by the control algorithm and the other compared algorithm was rejected, thus there are statistical differences between the control algorithm and those remaining. The adjusted p-values for different post-hoc procedures are shown in Table 8 and again the results can be considered statistically significant (let us also note *en passant* that significant results are also obtained if the analysis is replicated separately for different values of d, m and n).

Finally, with the aim of comparing the any-time behavior of the different algorithms, Figures 10–11 show the solution quality (as average RPD to best solutions) together with the execution time (SQT) provided by the different algorithms for instances in RealSet and the different settings for n, m and d (we have omitted SQTs for $d = 0.75 \cdot m$ as all algorithms reach optimal solutions right at the beginning of execution). It can be seen that $\text{MA}_{\text{GRASP+HC+PR}}$ produces better solutions than the remaining algorithms throughout the execution time in all cases, except for the case of $n = 200$, $m = 300$ and $d = 0.80 \cdot m$, for which $\text{GRASP}_{\text{FFR}}$ produces better results for the first 153 seconds of the execution.

Table 6: Results obtained by $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$, $\mathrm{GRASP}_{\mathrm{FFR}}$ and $\mathrm{GRASP}_{\mathrm{Mou}}$ for instances in RealSet. We have considered 18 datasets, each one comprising 5 different instances. Instances on each dataset are characterized by a number of strings (n) of the same length (m) and a distance threshold (d). Table shows mean solution obtained by each algorithm (sol), along with statistical values (mean (μ) and standard deviation (σ)) for relative percentage distance (RPD) from best solutions of results obtained by each algorithm. For $\mathrm{GRASP}_{\mathrm{FFR}}$ and $\mathrm{GRASP}_{\mathrm{Mou}}$ algorithms, mean improvement percentage of solutions (imp.%) obtained by $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ with respect to the ones obtained by these algorithms is also shown.

			$\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$		$\mathrm{GRASP}_{\mathrm{FFR}}$			$\mathrm{GRASP}_{\mathrm{Mou}}$		
d	n	m	sol.	RPD $\mu \pm \sigma$	sol.	RPD $\mu \pm \sigma$	imp.%	sol.	RPD $\mu \pm \sigma$	imp.%
$0.75 \cdot m$	100	300	100.00	0.00 ± 0.00	100.00	0.00 ± 0.00	0.00	100.00	0.00 ± 0.00	0.00
	100	600	100.00	0.00 ± 0.00	100.00	0.00 ± 0.00	0.00	100.00	0.00 ± 0.00	0.00
	100	800	100.00	0.00 ± 0.00	100.00	0.00 ± 0.00	0.00	100.00	0.00 ± 0.00	0.00
	200	300	200.00	0.00 ± 0.00	200.00	0.00 ± 0.00	0.00	200.00	0.00 ± 0.00	0.00
	200	600	200.00	0.00 ± 0.00	200.00	0.00 ± 0.00	0.00	200.00	0.00 ± 0.00	0.00
	200	800	200.00	0.00 ± 0.00	200.00	0.00 ± 0.00	0.00	200.00	0.00 ± 0.00	0.00
$0.80 \cdot m$	100	300	92.66	0.57 ± 0.71	87.03	6.62 ± 1.57	6.04	80.34	13.81 ± 1.36	13.24
	100	600	96.68	0.74 ± 0.57	86.30	11.41 ± 1.51	10.67	81.70	16.14 ± 1.25	15.40
	100	800	95.68	0.34 ± 0.54	83.05	13.48 ± 1.64	13.14	78.37	18.39 ± 1.46	18.05
	200	300	126.21	2.34 ± 1.58	120.92	6.40 ± 2.18	4.06	101.13	21.79 ± 2.13	19.45
	200	600	125.53	2.58 ± 1.90	114.39	11.33 ± 2.53	8.74	99.06	23.07 ± 1.77	20.49
	200	800	116.71	3.23 ± 1.87	103.43	14.24 ± 1.91	11.01	95.55	20.76 ± 1.59	17.54
$0.85 \cdot m$	100	300	37.96	2.69 ± 2.34	26.45	32.30 ± 5.84	29.61	34.92	10.47 ± 2.03	7.78
	100	600	34.23	3.81 ± 2.27	10.18	71.82 ± 6.83	68.01	30.31	14.86 ± 1.89	11.05
	100	800	31.07	4.07 ± 3.12	4.73	86.08 ± 7.17	82.01	27.61	14.65 ± 2.39	10.58
	200	300	47.98	4.43 ± 2.65	32.66	35.39 ± 8.39	30.97	44.12	12.03 ± 2.45	7.60
	200	600	40.65	4.52 ± 2.47	16.69	63.55 ± 14.51	59.02	38.50	9.34 ± 2.25	4.82
	200	800	31.48	4.00 ± 2.56	3.41	89.86 ± 4.05	85.86	29.91	8.82 ± 2.19	4.82

Table 7: Results for the Aligned Friedman Rank Test on RealSet instances for $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ and different state-of-the-art algorithms.

position	algorithm	ranking
1st	$\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$	1447.82
2nd	$\mathrm{GRASP}_{\mathrm{Mou}}$	3014.03
3rd	$\mathrm{GRASP}_{\mathrm{FFR}}$	3639.63

Table 8: Adjusted p-values for $N \times 1$ comparisons of control algorithm ($\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$) with different state-of-the-art algorithms for the Aligned Friedman Rank Test on RealSet instances.

algorithm	Bonf	Holm	Hoch	Homm	Holl	Rom	Finn	Li
$\mathrm{GRASP}_{\mathrm{FFR}}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$\mathrm{GRASP}_{\mathrm{Mou}}$	2.99E-199	1.49E-199	1.49E-199	1.49E-199	0.0	1.49E-199	0.0	1.49E-199

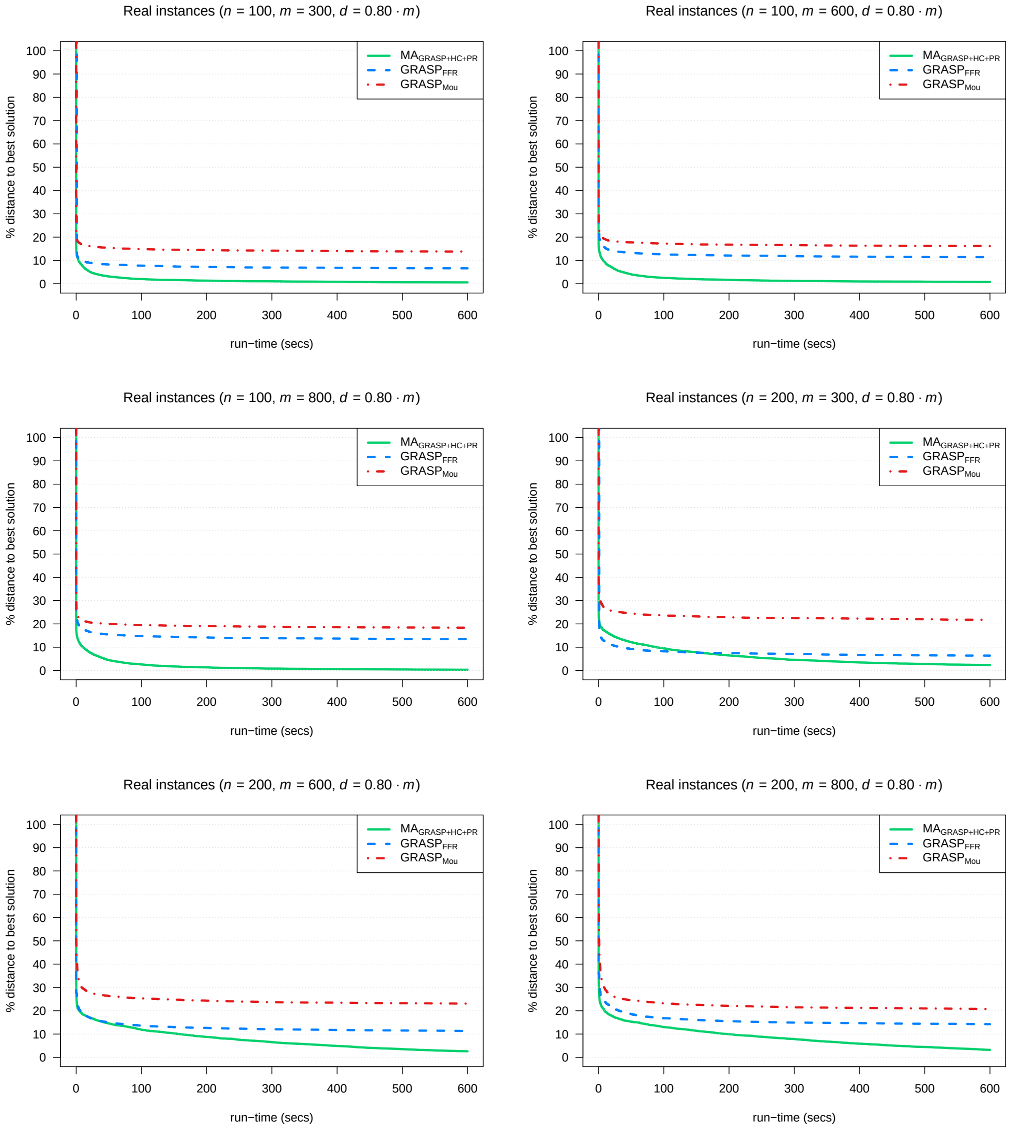

Figure 10: Solution quality over time as relative percentage distances (RPD) from best solutions of results obtained by state-of-the-art algorithms for the FFMSP ($\text{GRASP}_{\text{FFR}}$ and $\text{GRASP}_{\text{Mou}}$) and $\text{MA}_{\text{GRASP+HC+PR}}$ for instances in RealSet and distance threshold $d = 0.80 \cdot m$. Instances on each dataset are characterized by a number of strings (n) of the same length (m) and a distance threshold (d).

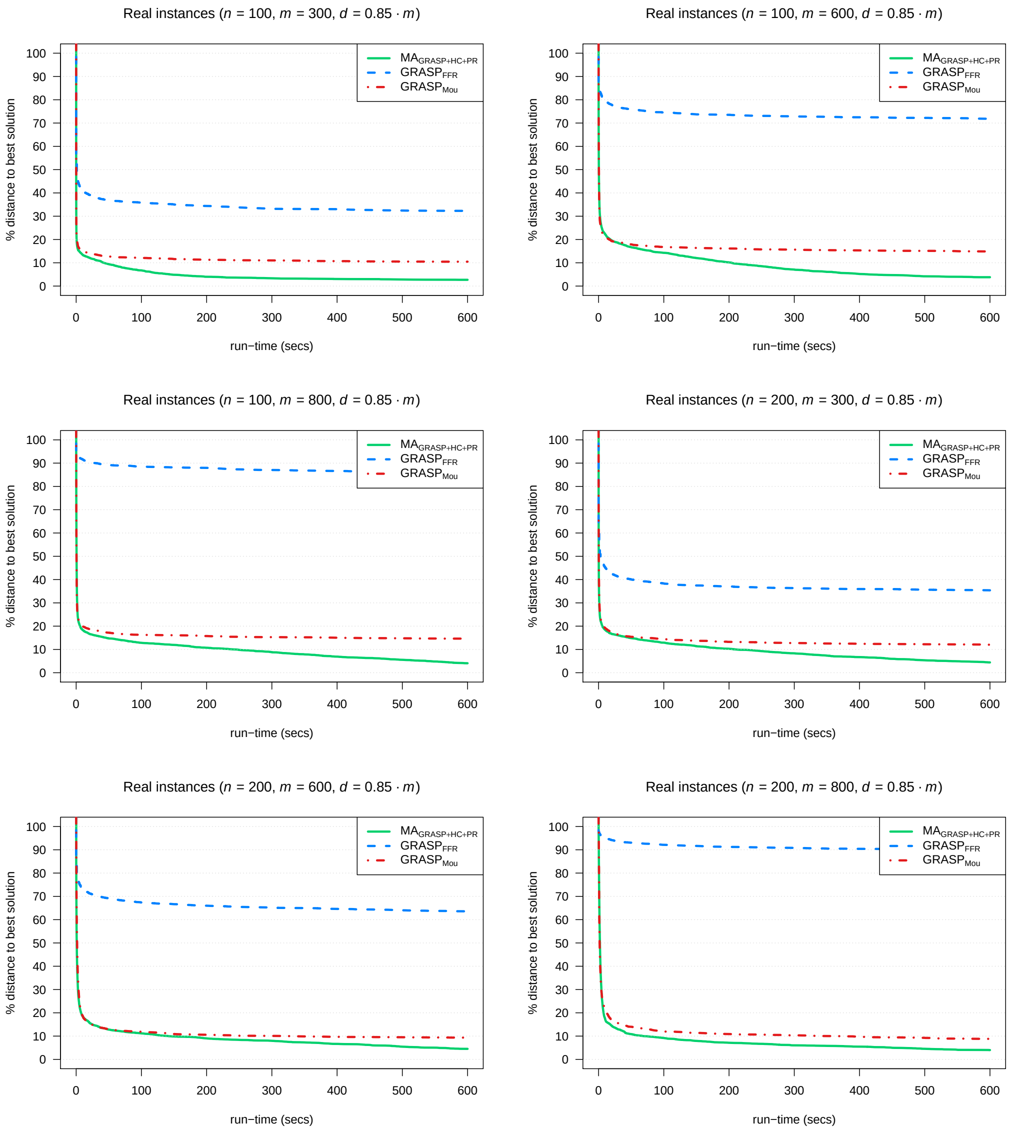

Figure 11: Solution quality over time as relative percentage distances (RPD) from best solutions of results obtained by state-of-the-art algorithms for the FFMSP ($\text{GRASP}_{\text{FFR}}$ and $\text{GRASP}_{\text{Mou}}$) and $\text{MA}_{\text{GRASP+HC+PR}}$ for instances in RealSet and distance threshold $d = 0.85 \cdot m$. Instances on each dataset are characterized by a number of strings (n) of the same length (m) and a distance threshold (d).

4.3. Very Large Real Instances

Finally, we have compared our proposal with state-of-the-art algorithms on very large real instances. For this benchmark, nine subsets of instances have been defined using 400 strings (n=400) and different string lengths ($m \in \{1600, 2400, 3200\}$) and distance thresholds ($d \in \{0.75 \cdot m, 0.80 \cdot m, 0.85 \cdot m\}$). For each of the nine subsets, five different instances have been generated yielding thus a total of 45 instances.

The results of this comparison are shown in Table 9. It can be observed that for $d = 0.75 \cdot m$ optimal solutions were found in all cases by $\text{MA}_{\text{GRASP+HC+PR}}$ and $\text{GRASP}_{\text{Mou}}$. For $d = 0.80 \cdot m$ and $d = 0.85 \cdot m$, the best results are obtained by $\text{MA}_{\text{GRASP+HC+PR}}$. The mean average improvement of $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{FFR}}$ is 18.12% whereas for $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{Mou}}$ it is 13.65%. For $d = 0.85 \cdot m$, the mean average improvement of $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{FFR}}$ is 88.20% whereas for $\text{MA}_{\text{GRASP+HC+PR}}$ over $\text{GRASP}_{\text{Mou}}$ it is 2.55%. It is notable that the performance of $\text{GRASP}_{\text{FFR}}$ substantially degrades in this case for $d = 0.85 \cdot m$, that is, the most restricted scenario. The heuristic fitness function used by $\text{MA}_{\text{GRASP+HC+PR}}$ and $\text{GRASP}_{\text{Mou}}$ seems to be able to cope with this situation better.

The results of the Aligned Friedman Rank Test for these instances are shown in Table 10 (distributed according to χ^2 with 2 degrees of freedom: 652.29). The p-value calculated by this test is $p = 2.54 \cdot 10^{-10}$, and this provides evidence for rejecting the null hypothesis that states equality of rankings between the populations. The results of post-hoc procedures comparing the control algorithm ($\text{MA}_{\text{GRASP+HC+PR}}$) with those remaining are shown in Table 11 as adjusted p-values. The null hypothesis that states equality between the distributions obtained by the control algorithm and the other compared algorithms was rejected by all procedures for all algorithms (results are statistically significant).

5. Conclusions

The FFMSP is a SSP of enormous difficulty, whose resolution demands the use of powerful heuristics. We have proposed a memetic algorithm to this end. Our MA has successfully integrated different metaheuristic components, namely, GRASP for population initialization, path relinking for recombination and hill climbing for local improvement, into an evolutionary search engine. A careful sensitivity analysis of the MA indicates that from a

Table 9: Results obtained by $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$, $\mathrm{GRASP}_{\mathrm{FFR}}$ and $\mathrm{GRASP}_{\mathrm{Mou}}$ for very large real instances. We have considered 9 datasets, each one comprising 5 different instances. Instances on each dataset are characterized by a number of strings (n) of the same length (m) and a distance threshold (d). Table shows mean solution obtained by each algorithm (sol), along with statistical values (mean (μ) and standard deviation (σ)) for relative percentage distance (RPD) from best solutions of results obtained by each algorithm. For $\mathrm{GRASP}_{\mathrm{FFR}}$ and $\mathrm{GRASP}_{\mathrm{Mou}}$ algorithms, mean improvement percentage of solutions (imp.%) obtained by $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ with respect to the ones obtained by these algorithms is also shown.

			$\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$		$\mathrm{GRASP}_{\mathrm{FFR}}$			$\mathrm{GRASP}_{\mathrm{Mou}}$		
d	n	m	sol.	RPD $\mu \pm \sigma$	sol.	RPD $\mu \pm \sigma$	imp.%	sol.	RPD $\mu \pm \sigma$	imp.%
$0.75 \cdot m$	400	1600	400.00	0.00 ± 0.00	399.94	0.02 ± 0.06	0.02	400.00	0.00 ± 0.00	0.00
	400	2400	400.00	0.00 ± 0.00	399.98	0.01 ± 0.03	0.01	400.00	0.00 ± 0.00	0.00
	400	3200	400.00	0.00 ± 0.00	399.91	0.02 ± 0.07	0.02	400.00	0.00 ± 0.00	0.00
$0.80 \cdot m$	400	1600	131.70	4.55 ± 2.31	112.80	18.76 ± 6.69	14.21	115.28	16.38 ± 2.74	11.83
	400	2400	143.12	5.34 ± 3.25	119.22	22.64 ± 11.44	17.30	118.99	20.88 ± 4.48	15.54
	400	3200	134.94	7.58 ± 3.43	102.35	30.43 ± 6.45	22.85	114.83	21.15 ± 3.82	13.58
$0.85 \cdot m$	400	1600	22.42	9.01 ± 5.88	0.61	97.51 ± 2.44	88.50	21.76	11.69 ± 5.70	2.67
	400	2400	20.67	13.21 ± 5.91	0.16	99.31 ± 1.59	86.10	20.20	15.20 ± 5.67	1.99
	400	3200	18.18	9.97 ± 5.76	0.00	100.00 ± 0.00	90.03	17.58	12.98 ± 5.62	2.98

Table 10: Results for the Aligned Friedman Rank Test on very large real instances for $\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$ and different state-of-the-art algorithms.

position	algorithm	ranking
1st	$\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$	725.28
2nd	$\mathrm{GRASP}_{\mathrm{Mou}}$	1307.96
3rd	$\mathrm{GRASP}_{\mathrm{FFR}}$	2018.26

Table 11: Adjusted p-values for $N \times 1$ comparisons of control algorithm ($\mathrm{MA}_{\mathrm{GRASP+HC+PR}}$) with different state-of-the-art algorithms for the Aligned Friedman Rank Test on very large real instances.

algorithm	Bonf	Holm	Hoch	Homm	Holl	Rom	Finn	Li
$\mathrm{GRASP}_{\mathrm{FFR}}$	7.06E-271	7.06E-271	7.06E-271	7.06E-271	0.0	7.06E-271	0.0	7.06E-271
$\mathrm{GRASP}_{\mathrm{Mou}}$	2.56E-56	1.28E-56	1.28E-56	1.28E-56	0.0	1.28E-56	0.0	1.28E-56

global perspective this full-fledged memetic approach (i) exhibits a statistically significant superiority to versions in which some of the components are substituted by standard operators, and (ii) provides better results when the population initialization is done with more intensive (i.e. higher greediness) GRASP. The comparison with state-of-the-art algorithms from Mousavi et al. (2012) and Ferone et al. (2013) is also favorable to the MA, which provides average improvements of 8% and 74%, respectively on the hardest random problem instances and of 8% and 60% on the hardest real-world instances. Statistical tests confirm the significance of these improvements.

Future work will be directed towards testing the scalability of the MA. For this purpose, we will consider the use of parallel versions of the technique. We also plan to transfer design ideas to related SSPs, in particular to distinguishing SSPs, leading to the actual application of the technique in practical bioinformatic problems.

6. Acknowledgements

Authors wish to thank D. Ferone, Dr. Festa and Dr. Resende for providing us with an implementation of their hybrid GRASP algorithm for the FFMSP and to anonymous reviewers for their suggestions and comments. Support from projects ANYSELF (TIN2011-28627-C04-01 – http://anyself.wordpress.com) of MICINN and DNEMESIS (TIC-6083 – http://dnemesis.lcc.uma.es/wordpress) of Junta de Andalucía is acknowledged.

References

Abrahamson, K.A., Downey, R.G., Fellows, M.R., 1995. Fixed-parameter tractability and completeness IV: On completeness for W[P] and PSPACE analogues. Annals of Pure and Applied Logic 73, 235 – 276.

Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M., 1998. Proof verification and the hardness of approximation problems. Journal of the ACM 45, 501–555.

Boucher, C., Landau, G.M., Levy, A., Pritchard, D., Weimann, O., 2013. On approximating string selection problems with outliers. Theoretical Computer Science 498, 107 – 114.

Cai, L., Chen, J., 1997. On fixed-parameter tractability and approximability of NP optimization problems. Journal of Computer and System Sciences 54, 465 – 474.

Downey, R.G., Fellows, M.R., 1995a. Fixed-parameter tractability and completeness I: Basic results. SIAM Journal on Computing 24, 873–921.

Downey, R.G., Fellows, M.R., 1995b. Fixed-parameter tractability and completeness II: On completeness for W[1]. Theoretical Computer Science 141, 109 – 131.

Dunn, O.J., 1961. Multiple comparisons among means. Journal of the American Statistical Association 56, 52–64.

Fellows, M.R., 2002. Parameterized complexity: The main ideas and connections to practical computing, in: Fleischer, R., Moret, B., Schmidt, E. (Eds.), Experimental Algorithmics. Springer Berlin Heidelberg. volume 2547 of *Lecture Notes in Computer Science*, pp. 51–77.

Feo, T.A., Resende, M.G., 1995. Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109–133.

Ferone, D., Festa, P., Resende, M.G.C., 2013. Hybrid metaheuristics for the far from most string problem, in: Blesa, M.J., Blum, C., Festa, P., Roli, A., Sampels, M. (Eds.), Hybrid Metaheuristics, Springer. pp. 174–188.

Festa, P., 2007. On some optimization problems in molecular biology. Mathematical Biosciences 207, 219 – 234.

Festa, P., Pardalos, P.M., 2012. Efficient solutions for the far from most string problem. Annals of Operations Research 196, 663–682.

Finner, H., 1993. On a monotonicity problem in step-down multiple test procedures. Journal of the American Statistical Association 88, 920–923.

Glover, F., Laguna, M., Martí, R., 2000. Fundamentals of scatter search and path relinking. Control and Cybernetics 39, 653–684.

Hochberg, Y., 1988. A Sharper Bonferroni Procedure for Multiple Tests of Significance. Biometrika 75, 800–802.

Holland, B.S., Copenhaver, M.D., 1987. An Improved Sequentially Rejective Bonferroni Test Procedure. Biometrics 43, 417–423.

Holm, S., 1979. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6, 65–70.

Hommel, G., 1988. A stagewise rejective multiple test procedure based on a modified Bonferroni test. Biometrika 75, 383–386.

JL Hodges Jr, J., Lehmann, E., 1962. Rank methods for combination of independent experiments in analysis of variance. The Annals of Mathematical Statistics 33, 482–497.

Khot, S., Ponnuswami, A.K., 2006. Better inapproximability results for max-clique, chromatic number and min-3lin-deletion, in: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (Eds.), Automata, Languages and Programming. Springer Berlin Heidelberg. volume 4051 of *Lecture Notes in Computer Science*, pp. 226–237.

Lanctot, J.K., Li, M., Ma, B., Wang, S., Zhang, L., 2003. Distinguishing string selection problems. Information and Computation 185, 41 – 55.

Li, J.D., 2008. A two-step rejection procedure for testing multiple hypotheses. Journal of Statistical Planning and Inference 138, 1521–1527.

Meneses, C., Oliveira, C., Pardalos, P., 2005. Optimization techniques for string selection and comparison problems in genomics. IEEE Engineering in Medicine and Biology Magazine 24, 81–87.

Moscato, P., 1989. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report 826. Caltech Concurrent Computation Program.

Moscato, P., 1999. Memetic algorithms: A short introduction, in: Corne, D., Dorigo, M., Glover, F. (Eds.), New Ideas in Optimization. McGraw-Hill, pp. 219–234.

Moscato, P., Cotta, C., 2010. A modern introduction to memetic algorithms, in: Gendreau, M., Potvin, J.Y. (Eds.), Handbook of Metaheuristics. Springer US. volume 146 of *International Series in Operations Research & Management Science*, pp. 141–183.

Mousavi, S.R., 2010. A hybridization of constructive beam search with local search for far from most strings problem. International Journal of Computational and Mathematical Science 4, 340–348.

Mousavi, S.R., Babaie, M., Montazerian, M., 2012. An improved heuristic for the far from most strings problem. J. Heuristics 18, 239–262.

Neri, F., Cotta, C., 2012. Memetic algorithms and memetic computing optimization: A literature review. Swarm and Evolutionary Computation 2, 1 – 14.

Neri, F., Cotta, C., Moscato, P., 2012. Handbook of Memetic Algorithms. volume 379 of *Studies in computational Intelligence*. Springer-Verlag.

Ong, Y.S., Lim, M.H., Chen, X., 2010. Memetic computation-past, present and future. IEEE Computational Intelligence Magazine 5, 24–31.

Papadimitriou, C.H., Yannakakis, M., 1991. Optimization, approximation, and complexity classes. Journal of Computer and System Sciences 43, 425 – 440.

Pappalardo, E., Pardalos, P., Stracquadanio, G., 2013. String selection problems, in: Optimization Approaches for Solving String Selection Problems. Springer New York. SpringerBriefs in Optimization, pp. 27–49.

Puchinger, J., Raidl, G.R., 2005. Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification, in: Mira, J., Álvarez, J.R. (Eds.), Artificial Intelligence and Knowledge Engineering Applications: a Bioinspired Approach, Springer-Verlag, Berlin Heidelberg. pp. 41–53.

Reichhardt, T., 1999. It's sink or swim as a tidal wave of data approaches. Nature 399, 517–520.

Rom, D.M., 1990. A sequentially rejective test procedure based on a modified Bonferroni inequality. Biometrika 77, 663–665.

Tyler, B.M., Tripathy, S., Zhang, X., Dehal, P., Jiang, R.H.Y., Aerts, A., Arredondo, F.D., Baxter, L., Bensasson, D., Beynon, J.L., Chapman, J., Damasceno, C.M.B., Dorrance, A.E., Dou, D., Dickerman, A.W., Dubchak, I.L., Garbelotto, M., Gijzen, M., Gordon, S.G., Govers, F., Grunwald, N.J., Huang, W., Ivors, K.L., Jones, R.W., Kamoun, S., Krampis, K., Lamour, K.H., Lee, M.K., McDonald, W.H., Medina, M., Meijer, H.J.G., Nordberg, E.K., Maclean, D.J., Ospina-Giraldo, M.D., Morris, P.F., Phuntumart, V., Putnam, N.H., Rash, S., Rose, J.K.C., Sakihama, Y., Salamov, A.A., Savidor, A., Scheuring, C.F., Smith, B.M., Sobral, B.W.S., Terry, A., Torto-Alalibo, T.A., Win, J., Xu, Z., Zhang, H., Grigoriev, I.V., Rokhsar, D.S., Boore, J.L., 2006. Phytophthora genome sequences uncover evolutionary origins and mechanisms of pathogenesis. Science 313, 1261–1266.