

Hybrid GRASP heuristics for the phylogeny problem combining path-relinking and genetic algorithm as an intensification strategy

Dalessandro Soares Vianna^{a*}, Marcilene de Fátima Dianin Vianna^b

^{a*}dalessandro@ucam-campos.br, UCAM, UFF, Brasil

^bmarcilenedianin@gmail.com, UFF, Brasil

Abstract

A phylogeny is a tree that relates taxonomic units based on their similarity over a set of characteristics. The phylogeny problem under the parsimony criterion consists in finding a phylogeny with a minimum number of evolutionary steps. We propose hybrid heuristic methods – based on GRASP, path-relinking and genetic algorithm methodologies – to build a phylogeny while minimizing parsimony. Computational experiments using benchmark conditions are reported, and the results obtained by the proposed hybrid heuristics are compared with the solutions obtained by a traditional GRASP (without hybridization) heuristic and with previously reported solutions in the literature. The experimental results illustrate that the proposed heuristics are efficient in terms of solution quality and time-to-target-value.

Keywords

Phylogeny problem. Evolutionary trees. GRASP. Path-relinking. Genetic algorithm. Heuristics.

1. Introduction

One of the central problems in comparative biology is that of establishing ancestry relations among species, groups of species, populations of distinct species, or homologous genes in populations of distinct species (AYALA, 1995; SWOFFORD, 1982). These entities can be classified as *taxons*. Ancestry relations are represented by a rooted tree, in which leaves represent the taxons under analysis and internal nodes represent hypothetical ancestral. This tree is called *phylogeny* (RIBEIRO; VIANNA, 2005). Taxons under analysis are called *operational taxons*, while taxons associated to internal nodes of a phylogeny are called *hypothetical taxons*.

Each taxon has a set of attributes called *characteristics*. A characteristic can represent a morphological attribute or a position of amino acids or nucleotides in a protein. Binary characteristics are those who have only two possible states, which represent the presence or the absence of some attribute. Instances of the phylogeny problem with binary characteristics are characterized by 0-1 matrices,

in which each element (i,j) corresponds to the state of characteristic j within taxon i .

An evolutionary step is associated with each change of state along a branch of a phylogeny. The parsimony criterion states that the best phylogeny is the more parsimonious, i.e., the one that can be explained by the minimum number of evolutionary steps (EDWARDS; CAVALLI-SFORZA, 1964; HENNIG, 1966; SWOFFORD, 1982). Parsimony is often legitimated as the most appropriate evaluation criterion for phylogenies, since evolutionary changes occur with very small probabilities (PENNY; FOULDS; HENDY, 1982; SOBER, 1987). Given a set of taxons defined by a set of characteristics, the *phylogeny problem* is that of finding a phylogeny with the minimum number of evolutionary steps. It is NP-hard in general and in common restricted cases (BODLAENDER; FELLOWS; WARNOW, 1992; DAY; JOHNSON; SANKOFF, 1986; FOULDS; GRAHAM, 1982a; FOULDS; GRAHAM, 1982b).

Figure 1(a), adapted from Kitching et al. (1998), gives an example with $n=4$ taxons (lamprey, shark,

salmon, and lizard) and $m=6$ binary characteristics: (a) paired fins, (b) jaws, (c) large dermal bones, (d) fin rays, (e) lungs, and (f) rasping tongue. We assume that the operational taxons descend from a common ancestral, in which all characteristics are absent. Changes in the characteristics are marked in the branches of the tree. Figure 1b and Figure 1c show two phylogenetic trees with different hypothetical taxons (internal nodes) explaining the ancestry relations. The solution in Figure 1(c) is that with the minimum parsimony value.

The first work for computing phylogenies under the parsimony criterion was proposed by Luckow and Pimentel (1985). They proposed different mainframe programs. Later and Platnick (1987, 1989) compared phylogeny inference systems running on microcomputers. Andreatta and Ribeiro (2002) proposed a framework (under the paradigm of object oriented programming) for the development of local search heuristics and metaheuristics for combinatorial optimization problems. This framework was used for the development of a family of local search heuristics for the phylogeny problem under the parsimony criterion. It made possible a fair comparison of heuristics running under the same conditions on a set of eight benchmark problems (ANDREATTA; RIBEIRO, 2002). Ribeiro and Vianna (2005) proposed a GRASP+VND heuristic that found the best known

solutions to date for benchmark instances. This heuristic makes use of the k -SPR neighborhood. A move in the latter may be seen as the combination of k consecutive moves in the SPR (Subtree Pruning and Regrafting) neighborhood (ANDREATTA, 1998). A genetic algorithm was proposed by Ribeiro and Vianna (2009).

Path-relinking was originally proposed as an intensification strategy to explore trajectories connecting elite solutions obtained by tabu search or scatter search (GLOVER, 1996, 2000; GLOVER; LAGUNA, 1997; GLOVER; LAGUNA; MARTÍ, 2000). It was also successfully used to introduce memory mechanisms in implementations of GRASP (FESTA; RESENDE, 2009a, b; MAURI; LORENA, 2009; RESENDE; RIBEIRO, 2003, 2005; VIANNA et al., 2007). Cotta (2006) used path-relinking within an implementation of scatter search for solving the phylogeny problem under a distance-based optimization criterion.

The aim of this paper is to propose hybrid GRASP heuristics for the phylogeny problem, which make use of path-relinking and genetic algorithm as an intensification strategy. We evaluate experimentally the proposed heuristics and we compare the solutions obtained by them with previously known solutions in the literature.

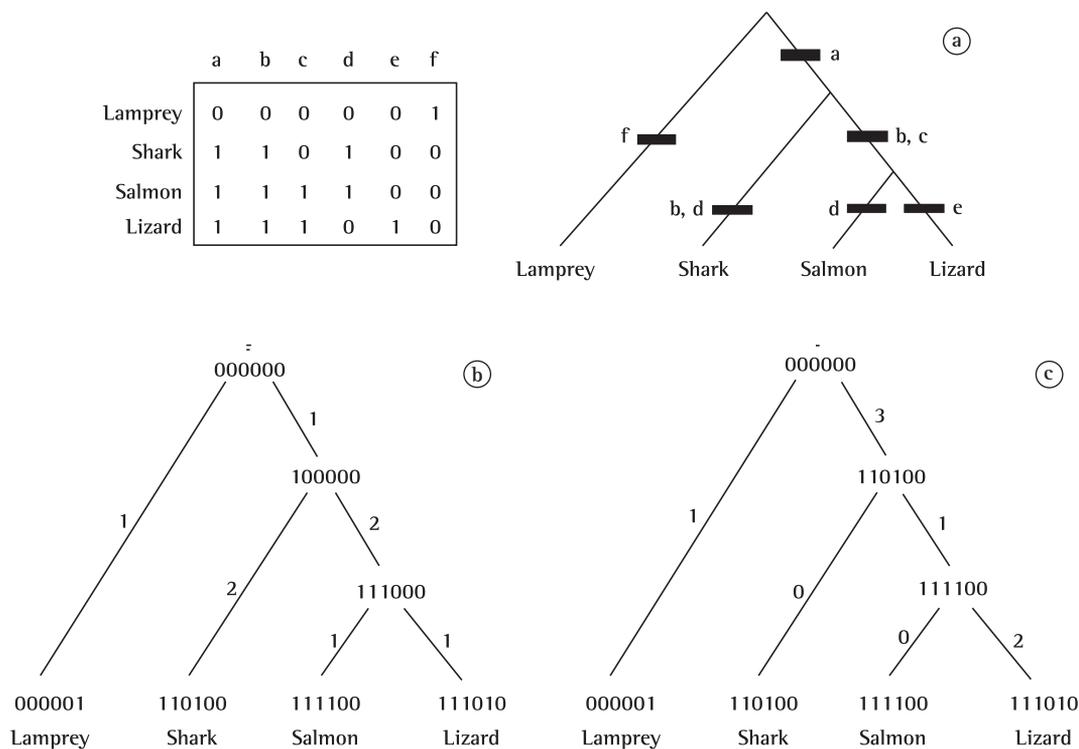


Figure 1. Example: taxons, characteristics, and two possible phylogenetic trees.

The remainder of the paper is organized as follows. In Section 2 is described the hybrid GRASP heuristics proposed. Experimental results, with benchmark instances, are presented in Section 3. Finally, concluding remarks are made in Section 4.

2. Hybrid GRASP heuristics proposed

GRASP – Greedy Randomized Adaptive Search Procedure (FEO; RESENDE, 1995; RESENDE; RIBEIRO, 2003) is a multi-start or iterative process, in which each iteration consists of two phases: construction and local search. The construction phase builds a trial solution using a greedy randomized adaptive algorithm, whose neighborhood is investigated until a local minimum is found during the local search phase. The best solution from all iterations is returned as result.

We propose two hybrid GRASP heuristics: HGRASP and HGRASP+LS. Both heuristics work through a pool of elite solutions P found during the execution of GRASP iterations. They have an additional phase, called intensification phase, which uses P with the objective of integrating features of good solutions, found during the iterations of GRASP, into new solutions generated in subsequent iterations. In the intensification phase, genetic algorithm is used.

In traditional GRASP, i.e., GRASP without the intensification phase, all iterations are independent and therefore most good solutions are simply forgotten. The intensification phase retains previous solutions and using them as guides to speed up convergence to a good quality solution.

The remainder of this section is organized as follows. In Subsections 2.1 and 2.2, we state, respectively, the constructive and local search algorithms. In Subsection 2.3 is detailed the intensification phase, which uses the path-relinking based crossover. The heuristics HGRASP and HGRASP+LS are presented in Subsections 2.4 and 2.5, respectively.

2.1. Greedy randomized adaptive construction

In the construction phase, it was used the randomized version GStep_wR of the greedy GStep heuristic (ANDREATTA; RIBEIRO, 2002). Numerical results have shown that this algorithm finds better solutions than others, although at the cost of slightly higher computation times.

Whenever a taxon i is inserted into a partial phylogeny under construction by algorithm GStep, all branches of the latter are evaluated (as illustrated in Figure 2). The chosen branch is that minimizing the insertion cost of taxon i (i.e., the most parsimonious).

Since there are still $n-(k-1)$ unselected taxons in iteration k and $2k-5$ possible branches for each insertion, the overall complexity of each construction using algorithm GStep is

$$\sum_{k=3}^n O(mk)(2k-5)[n-(k-1)] = O(mn^4) \quad (1)$$

In its randomized version, GStep_wR, the branch where taxon i will be inserted is randomly selected from among all those with cost at most $\alpha=10\%$ higher than the most parsimonious incremental.

2.2. Local search method

The local search method explores the SPR neighborhood (SWOFFORD; OLSEN, 1990; SWOFFORD et al., 1996), also described in (ANDREATTA; RIBEIRO, 2002; RIBEIRO; VIANNA, 2005). First, a subtree of the current phylogeny is disconnected. Next, it is reconnected in a different position. Figure 3

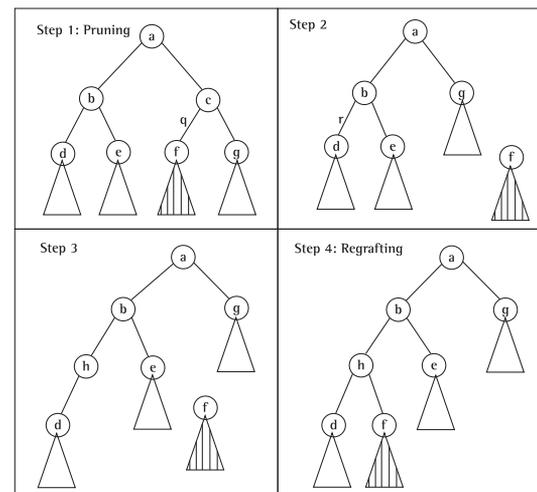


Figure 3. Illustration of a move in neighborhood SPR – adapted from (RIBEIRO; VIANNA, 2005).

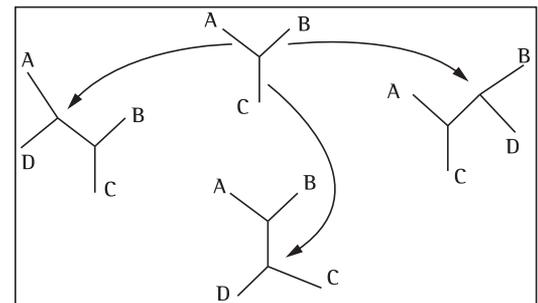


Figure 2. Alternatives for the insertion of a new taxon into a phylogeny with three taxa (ANDREATTA, 1998).

illustrates an example of a move in this neighborhood, in accordance with the steps below (RIBEIRO; VIANNA, 2009):

- Step 1: An edge $q = (c, f)$ of the current phylogeny is selected and removed. The subtree containing node c is the *base subtree*, while that containing node f is the *pending subtree*;
- Step 2: Node c is destroyed in the base subtree and its two adjacent nodes are directly connected by a new edge (a, g) , which results from collapsing the original edges (a, c) and (c, g) . Next, an edge $r = (b, d)$ of the base subtree is selected for reconnecting the pending subtree;
- Step 3: A new node h is created and edge $r = (b, d)$ is replaced by two edges (b, h) and (h, d) in the base subtree;
- Step 4: The pending subtree is reconnected to the base subtree through node h created in the previous step.

A phylogeny has $O(n)$ potential subtrees and each of them can be reconnected by $O(n)$ possible edges. Therefore, each solution has $O(n^2)$ neighbors in the SPR neighborhood. The local search heuristic follows a best-improving strategy: all neighbors are evaluated and the best improving move is selected. The search stops when no improving move exist.

2.3. Intensification method

The intensification method proposed is a genetic algorithm that uses path-relinking as a progressive crossover operator. The use of path-relinking within a GRASP, as an intensification strategy applied to each locally optimal solution, was first used by Laguna and Marti (1999). Surveys of GRASP with path-relinking are given in (RESENDE; RIBEIRO, 2005; RIBEIRO; RESENDE, 2012). The path-relinking crossover method used in this work is described in Subsection 2.3.1.

Figure 4 gives the algorithmic description of procedure Intensification which receives as input parameters the population of solutions, pop , and the size of it. The loop in lines 2 to 17 performs iterations (generations) while the stop criterion defined is not satisfied. During the generations, the population evolves with the goal of obtaining better quality individuals. We assume that the procedure finishes when during a generation no offspring solution replaces a parent solution in pop . During a generation, $size_pop/2$ crossover operations are executed, which can be visualized in the loop in lines 4 to 16. Two parent solutions are randomly selected in lines 5 and 6. The path-relinking based crossover method, detailed in Subsection 2.3.1, is applied in line 7, generating the offspring solution s . If one parent solution is worse than s , it is replaced by s in

```

Procedure intensification (pop, size_pop)
1. finished ← false;
2. while not finished do
3.   finished ← true;
4.   for k ← 1 to size_pop/2 do
5.      $i_1 \leftarrow \text{Random}(\text{size\_pop})$ ;
6.      $i_2 \leftarrow \text{Random}(\text{size\_pop})$ ; //  $i_2 \neq i_1$ 
7.      $s \leftarrow \text{PR\_Crossover}(\text{pop}[i_1], \text{pop}[i_2])$ ;
8.     if  $s$  is better than  $\text{pop}[i_1]$  or  $s$  is better than
       pop  $[i_2]$  then
9.       finished ← false;
10.      if pop  $[i_1]$  is better than pop  $[i_2]$  then
11.        pop  $[i_2] \leftarrow s$ ;
12.      else
13.        pop  $[i_1] \leftarrow s$ ;
14.      endif
15.    endif
16.  endfor
17. endwhile
18. Let  $s^*$  be the best solution in pop;
19. return  $s^*$ ;
end intensification.

```

Figure 4. Procedure Intensification.

pop (lines 10 to 14). If s is better than both parent solutions, s replaces the worst one. Finally, the best solution in pop is returned in line 19.

2.3.1. Crossover by path relinking

Path relinking was suggested as an approach to integrate intensification and diversification strategies in the context of tabu search (GLOVER, 1996; GLOVER; LAGUNA, 1997). Path-relinking is usually carried out between two solutions: one is called the *initial solution*, while the other is the *guiding solution*. One or more paths in the solution space graph connecting these solutions are explored in the search for better solutions. To generate the paths, moves applied to the initial solution introduce attributes contained in the guiding solution.

Given two parent solutions s_1 and s_2 , a bidirectional path relinking (RESENDE; RIBEIRO, 2005) is conducted: one path is computed leading from s_1 to s_2 and another leading from s_2 to s_1 . The best solution along them is returned as the offspring resulting from crossover. This mechanism is an extension of the traditional crossover operation: instead of producing only one offspring, defined by one single combination of two parents, it investigates many solutions that share the same characteristics of the selected parents.

Let s_1 and s_2 be, respectively, the initial and the guiding phylogenies. Let N_1 be the root of the phylogeny s_1 and N_2 the root of s_2 . By applying path-relinking to s_1 and s_2 , the set of operational taxons in the left subtree of N_1 will be made equal to the set of operational taxons in the left subtree

of N_2 . Consequently, the operational taxons in the right subtree of N_1 will become the same appearing in the right subtree of N_2 . This process is initiated at the roots of trees s_1 and s_2 and top-down propagated until their leaves are reached.

Figure 5 illustrates the application of crossover by path-relinking to phylogenies s_1 and s_2 . The phylogeny s_1 in Figure 5-I will be progressively modified until it becomes equal to s_2 . The nodes marked in s_1 and s_2 are those corresponding to N_1 and N_2 at each step, respectively. The final situation is described in Figure 5-VI, where both phylogenies coincide.

2.4. Heuristic HGRASP

The heuristic HGRASP makes GRASP iterations until the stop criterion be satisfied. At the end of each

iteration, it is verified if the obtained solution s will be inserted in the pool P of solutions, that is, if there is no other solution in P with the same parsimony value than s . When P reaches a predetermined size, it is initiated the intensification phase described in Subsection 2.3.

Figure 6 gives the algorithmic description of procedure HGRASP, which receives as input parameters the randomness percentage, α , used at the construction phase and the size of the pool P , $size_P$. In lines 1 to 3 the variables s^* , P and acc_P are initialized. s^* store the current best solution found; P is a data structure used to store the elite solutions; and acc_P signalsizes the total of elite solutions stored in P . The loop in lines 4 to 22 performs GRASP iterations until the stop criterion defined be satisfied. In line 5, an initial solution s is built (using the constructive

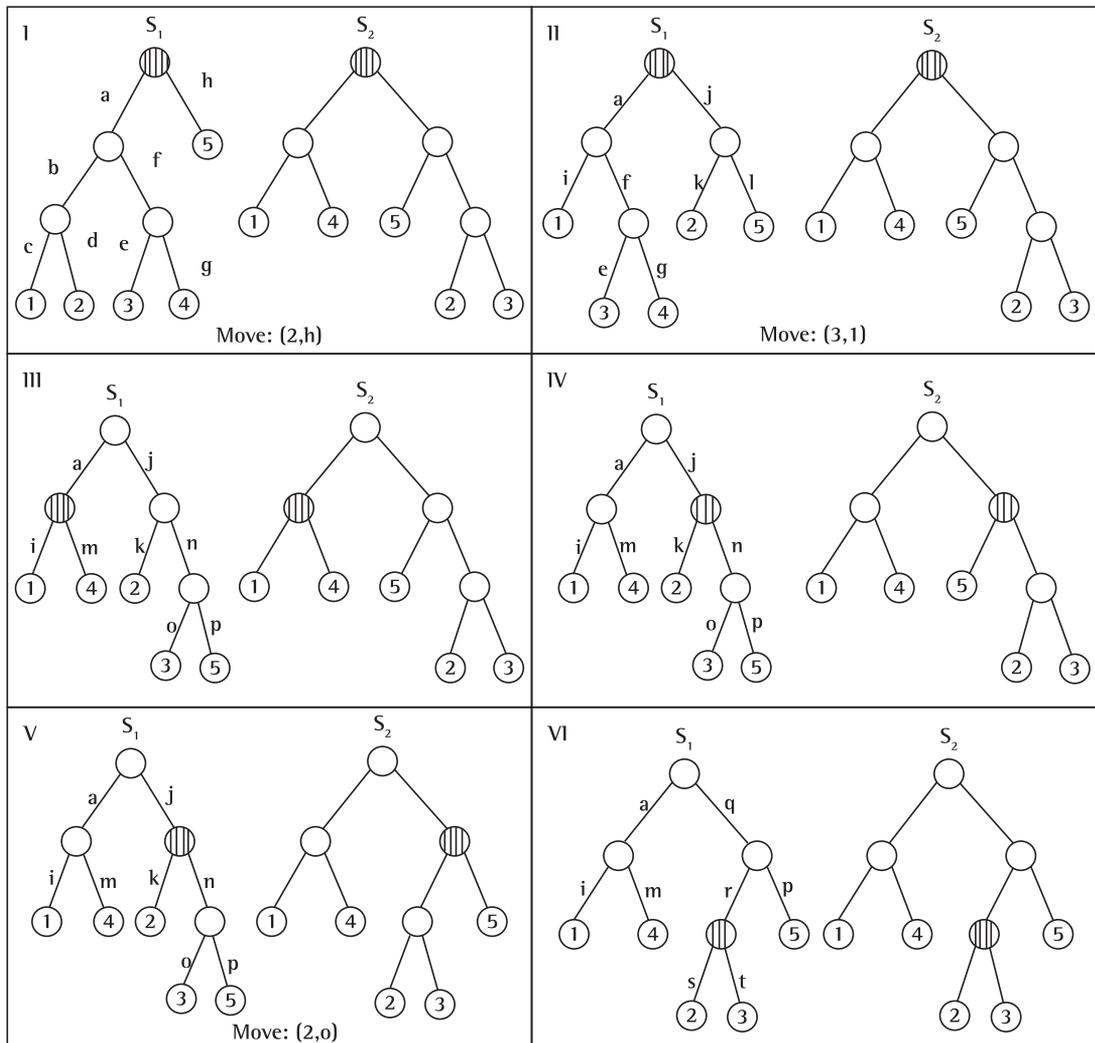


Figure 5. Example of the application of path-relinking to two phylogenies s_1 and s_2 .

method described in Subsection 2.1), in which a local search procedure (described in Subsection 2.2) is applied generating the solution s' . In case a better (more parsimonious) solution is found, s^* is updated in line 8. If s' satisfies the acceptance criterion, it is inserted in P in line 12. A solution is accepted to be inserted in P if there is no other solution in P with the same parsimony value (number of evolutionary steps). If P is complete, the intensification phase begins. The procedure *Intensification*, detailed in Subsection 2.3, works as a genetic algorithm, in which P represents the population of solutions and *size_P* the population size. In case a better solution is found, s^* is updated in line 16. The variables P and *acc_P* are reinitialized, respectively, in lines 18 and 19. Finally, the best solution found is returned in line 23.

2.5. Heuristic HGRASP+LS

The heuristic HGRASP+LS is similar to heuristic HGRASP. The difference is that the local search procedure, described in Subsection 2.2, is applied to the offspring solution s obtained by the crossover operator of procedure *Intensification*. This strategy is based on the metaheuristic ILS – Iterated Local Search (LOURENÇO; MARTIN; STUETZLE, 2002), in which a solution is perturbed to be refined by a local search procedure. In the heuristic HGRASP+LS, the path-relinking based crossover works as a perturbation method. The use of local search procedures with genetic algorithms is also known as memetic algorithms (NERI; COTTA; MOSCATO, 2012).

3. Computational results

All computational experiments were performed on a 2.13 GHz Intel Core™ Duo P7450 with 4 Gbytes of RAM memory. The heuristics HGRASP and HGRASP+LS were implemented in C using version 6.0 of the Microsoft Visual C++ compiler. We used an implementation in C of the random number generator described in (SCHRAGE, 1979).

We used two sets of benchmark instances. The first one is composed by eight real-life instances presented in (LUCKOW; PIMENTEL, 1985) and the second one by twenty randomly generated instances presented in (RIBEIRO; VIANNA, 2005).

We use $\alpha=10\%$ and $SizeP=20$, which represent, respectively, the randomness percentage used in constructive phase and the size of the pool P used in the intensification phase.

The heuristics HGRASP and HGRASP+LS were compared with a traditional GRASP heuristic, i.e.,

a GRASP heuristic that uses the construction phase described in Subsection 2.1 and the local search phase described at Subsection 2.2, but does not use the intensification phase proposed in Subsection 2.3. The proposed heuristics were also compared with the currently best known solutions for each test instance (RIBEIRO; VIANNA, 2005).

In the first experiment, the number of GRASP iterations was established as a stopping criterion. In this experiment, 100 iterations were performed. Each instance was run ten times with different seeds. For each test problem, we report in Table 1 its identification, the number of taxons (n), the number of characteristics (m) and the average computational time (in seconds) spent by each heuristic. Table 2 presents, for each instance, the parsimony value of the currently best known solution (*current*), followed by the best and average parsimony values obtained by each heuristic over ten runs. It is also report the percentage difference (*gap*) between each solution found and the currently best known solution. In the last line are presented the average *gaps* obtained by each heuristic. Results marked in bold face are the best obtained for each instance.

As can be seen in Table 1, the heuristic GRASP (traditional GRASP) obtained the best solutions

Table 1. Comparative computational times (100 GRASP iterations).

Instance	n	m	GRASP	HGRASP	HGRASP+LS
ANGI	49	59	6.7	6.7	6.7
GRIS	47	93	8.3	8.5	8.7
TENU	56	179	31.9	31.9	32.9
ETHE	58	86	17.9	18.0	18.6
ROPA	75	82	37.3	37.8	38.9
GOLO	77	97	43.2	46.4	51.7
CARP	117	110	183.7	187.3	189.9
SCHU	113	146	351.6	391.4	390.1
TST01	45	61	4.5	7.9	7.4
TST02	47	151	9.5	17.9	20.3
TST03	49	11	9.8	18.2	18.3
TST04	50	97	10.1	15.9	17.4
TST05	52	75	7.7	14.6	16.1
TST06	54	65	9.4	15.5	16.6
TST07	56	143	17.7	31.0	36.2
TST08	57	119	17.1	27.7	30.3
TST09	73	93	13.4	25.1	27.8
TST10	60	71	13.6	22.1	26.8
TST11	63	63	17.2	21.4	24.8
TST12	64	147	30.7	47.1	51.7
TST13	65	113	24.5	42.2	46.6
TST14	67	99	26.0	43.0	47.8
TST15	69	77	22.5	36.8	40.8
TST16	70	69	23.2	37.8	39.1
TST17	71	159	38.0	66.2	75.8
TST18	73	117	34.8	57.5	64.4
TST19	74	95	33.3	51.3	57.5
TST20	75	79	28.8	44.9	51.0

Table 2. Comparative results (100 GRASP iterations).

Instance	current	Best solution values						Average solution values					
		GRASP	HGRASP	HGRASP+LS	GRASP	HGRASP	HGRASP+LS						
ANGI	216	216	0.0%	216	0.0%	216	0.0%	216.6	0.3%	216.8	0.4%	216.5	0.2%
GRIS	172	172	0.0%	172	0.0%	172	0.0%	172.0	0.0%	172.0	0.0%	172.0	0.0%
TENU	682	682	0.0%	682	0.0%	682	0.0%	682.0	0.0%	682.0	0.0%	682.0	0.0%
ETHE	372	373	0.3%	373	0.3%	372	0.0%	373.0	0.3%	373.2	0.3%	372.6	0.2%
ROPA	325	326	0.3%	326	0.3%	326	0.3%	326.6	0.5%	326.7	0.5%	326.0	0.3%
GOLO	496	499	0.6%	499	0.6%	496	0.0%	500.0	0.8%	500.3	0.9%	496.5	0.1%
CARP	548	551	0.6%	550	0.4%	550	0.4%	551.0	0.6%	550.3	0.4%	550.5	0.5%
SCHU	759	760	0.1%	759	0.0%	759	0.0%	760.4	0.2%	759.0	0.0%	759.4	0.1%
TST01	547	553	1.1%	548	0.2%	548	0.2%	553.8	1.2%	553.3	1.1%	552.2	0.9%
TST02	1361	1367	0.4%	1365	0.3%	1360	-0.1%	1367.0	0.4%	1365.0	0.3%	1360.0	-0.1%
TST03	836	845	1.1%	841	0.6%	841	0.6%	845.0	1.1%	847.6	1.4%	844.1	1.0%
TST04	590	601	1.9%	601	1.9%	600	1.7%	601.0	1.9%	601.1	1.9%	600.3	1.8%
TST05	792	799	0.9%	800	1.0%	795	0.4%	801.5	1.2%	801.2	1.2%	796.2	0.5%
TST06	600	605	0.8%	606	1.0%	603	0.5%	607.5	1.3%	606.7	1.1%	603.8	0.6%
TST07	1274	1289	1.2%	1288	1.1%	1281	0.6%	1291.4	1.4%	1290.3	1.3%	1282.1	0.6%
TST08	862	876	1.6%	874	1.4%	870	0.9%	877.2	1.8%	875.5	1.6%	870.9	1.0%
TST09	1147	1157	0.9%	1160	1.1%	1153	0.5%	1157.7	0.9%	1161.3	1.3%	1155.2	0.7%
TST10	722	737	2.1%	734	1.7%	727	0.7%	737.5	2.2%	734.0	1.7%	728.3	0.9%
TST11	546	555	1.6%	553	1.3%	552	1.1%	555.5	1.7%	554.0	1.5%	552.9	1.3%
TST12	1224	1235	0.9%	1238	1.1%	1227	0.3%	1237.1	1.1%	1239.7	1.3%	1230.2	0.5%
TST13	1524	1537	0.9%	1534	0.7%	1527	0.2%	1538.5	0.9%	1535.6	0.8%	1530.7	0.4%
TST14	1171	1184	1.1%	1175	0.3%	1171	0.0%	1184.1	1.1%	1178.8	0.7%	1179.7	0.7%
TST15	758	770	1.6%	772	1.9%	768	1.3%	772.0	1.9%	772.7	1.9%	768.0	1.3%
TST16	537	553	3.0%	551	2.6%	546	1.7%	553.7	3.1%	552.0	2.8%	546.0	1.7%
TST17	2468	2481	0.5%	2484	0.6%	2471	0.1%	2484.7	0.7%	2484.4	0.7%	2472.7	0.2%
TST18	1531	1554	1.5%	1549	1.2%	1542	0.7%	1559.7	1.8%	1554.8	1.5%	1542.0	0.7%
TST19	1024	1036	1.2%	1037	1.3%	1032	0.8%	1039.6	1.5%	1037.9	1.4%	1035.2	1.1%
TST20	671	680	1.3%	681	1.5%	678	1.0%	681.0	1.5%	682.0	1.6%	679.8	1.3%
			1.0%		0.9%		0.5%		1.2%		1.1%		0.7%

for four instances and the best average values for two instances. Heuristic HGRASP obtained the best solutions for eight instances and the best average values for five. Heuristic HGRASP+LS obtained the best solutions for all instances and the best average values for 25 out of the 28 instances. Comparing the *gaps*, the proposed heuristics obtained solutions very close to the currently best know solutions.

Since the computational times spent by the proposed heuristics were a little greater than the computational times spent by the traditional GRASP heuristic (see Table 1), a second experiment was done, in which ten runs of each heuristic were performed on each instance, fixing the computational time of each run. Tables 3 and 4 present the obtained results with the computation times limited to 60 and 300 seconds, respectively.

With the computational time limited to 60 seconds, heuristic GRASP obtained the best solutions for

five instances and the best average values for four instances. Heuristic HGRASP obtained the best solutions for eight instances and the best average values for seven. Heuristic HGRASP+LS obtained the best solutions for 27 out of 28 instances and the best average values for 26 out of the 28 instances. With the computational time limited to 300 seconds, heuristic GRASP obtained the best solutions for eight instances and the best average values for four instances. Heuristic HGRASP obtained the best solutions for ten instances and the best average values for five. Heuristic HGRASP+LS obtained the best solutions for all instances and the best average values for 27 out of the 28 instances.

Comparing the obtained *gaps*, the propose heuristics obtained solutions very close to the currently best know solutions. The heuristic HGRASP+LS improved the best known solution for four instances (TST02, TST05, TST06 and TST17).

Table 3. Comparative results (60 seconds).

Instance	Best solution values							Average solution values					
	<i>current</i>	GRASP		HGRASP		HGRASP+LS		GRASP		HGRASP		HGRASP+LS	
ANGI	216	216	0.0%	216	0.0%	216	0.0%	216.0	0.0%	216.0	0.0%	216.0	0.0%
GRIS	172	172	0.0%	172	0.0%	172	0.0%	172.0	0.0%	172.0	0.0%	172.0	0.0%
TENU	682	682	0.0%	682	0.0%	682	0.0%	682.0	0.0%	682.0	0.0%	682.0	0.0%
ETHE	372	372	0.0%	372	0.0%	372	0.0%	372.3	0.1%	372.2	0.1%	372.2	0.1%
ROPA	325	326	0.3%	326	0.3%	326	0.3%	326.5	0.5%	326.0	0.3%	326.0	0.3%
GOLO	496	497	0.2%	497	0.2%	496	0.0%	498.8	0.6%	497.4	0.3%	496.3	0.1%
CARP	548	551	0.6%	549	0.2%	551	0.6%	551.9	0.7%	549.9	0.3%	551.4	0.6%
SCHU	759	761	0.3%	760	0.1%	760	0.1%	761.4	0.3%	761.4	0.3%	761.6	0.3%
TST01	547	549	0.4%	550	0.6%	548	0.2%	550.1	0.6%	550.5	0.6%	548.8	0.3%
TST02	1361	1365	0.3%	1365	0.3%	1361	0.0%	1365.8	0.3%	1365.0	0.3%	1361.0	0.0%
TST03	836	842	0.7%	843	0.8%	839	0.4%	843.2	0.9%	844.4	1.0%	839	0.4%
TST04	590	595	0.9%	597	1.2%	594	0.7%	597.8	1.3%	599.2	1.6%	594.9	0.8%
TST05	792	795	0.5%	797	0.6%	794	0.3%	797.7	0.7%	799.0	0.9%	794.3	0.3%
TST06	600	605	0.8%	602	0.3%	602	0.3%	606.3	1.0%	606.0	1.0%	604.4	0.7%
TST07	1274	1288	1.1%	1288	1.1%	1279	0.4%	1289.6	1.2%	1291.1	1.3%	1281.7	0.6%
TST08	862	868	0.7	872	1.2%	867	0.6%	869.2	0.8%	872.8	1.3%	867.0	0.6%
TST09	1147	1157	0.9%	1157	0.9%	1151	0.3%	1157.6	0.9%	1159.2	1.1%	1153.8	0.6%
TST10	722	732	1.4%	732	1.4%	726	0.6%	733.0	1.5%	735.2	1.8%	728.2	0.9%
TST11	546	553	1.3%	554	1.5%	549	0.6%	555.2	1.7%	555.5	1.7%	549.3	0.6%
TST12	1224	1238	1.1%	1238	1.1%	1234	0.8%	1237.2	1.1%	1238	1.1%	1234.7	0.9%
TST13	1524	1533	0.6%	1534	0.7%	1530	0.4%	1534.8	0.7%	1536.8	0.8%	1530.9	0.4%
TST14	1171	1181	0.9%	1183	1.0%	1175	0.3%	1182.4	1.0%	1184.1	1.1%	1176.0	0.4%
TST15	758	772	1.9%	773	2.0%	765	0.9%	772.9	2.0%	775.7	2.3%	766.5	1.1%
TST16	537	551	2.6%	547	1.9%	542	0.9%	552.0	2.8%	547.6	2.0%	547.5	2.0%
TST17	2468	2483	0.6%	2478	0.4%	2466	-0.1%	2484.2	0.7%	2480.8	0.5%	2469.4	0.1%
TST18	1531	1554	1.5%	1551	1.3%	1545	0.9%	1554.1	1.5%	1553.8	1.5%	1545.0	0.9%
TST19	1024	1039	1.5%	1033	0.9%	1028	0.4%	1039.8	1.5%	1035.2	1.1%	1028.6	0.4%
TST20	671	684	1.8%	682	1.6%	676	0.8%	684.8	2.1%	684.2	2.0%	677.6	1.0%
			0.8%		0.7%		0.4%		1.0%		0.9%		0.5%

To effectively compare the proposed heuristics with the traditional GRASP heuristic, we compared the behavior of each heuristic on instances ROPA and TST15 using the methodology proposed by Aiex, Resende and Ribeiro (2002). Instance TST15 is harder than instance ROPA. Two hundred independent runs for each heuristic were done for each instance. Each run was terminated when a solution of value less than or equal to a certain target value was found. The target values were set at 327 for instance ROPA and 772 for instance TST15. Although each of these sub-optimal values was chosen such that the slowest heuristic could terminate in a reasonable amount of computation time, the relative behavior of the two heuristics is not affected by this choice. Empirical probability distributions for the time-to-target-solution-value are plotted in Figures 7 and 8. To plot the empirical distribution for each algorithm, we follow the procedure described in (AIEX; RESENDE; RIBEIRO, 2002). We associate with the i -th smallest running time t_i a probability $p_i = \left(i - \frac{1}{2}\right) / 200$, and

plot the points $z_{i-} (t_i, p_i)$, for $i=1, \dots, 200$.

Procedure HGRASP (α , size_P)

```

1.  $s^* \leftarrow \emptyset$ ;
2.  $P \leftarrow \emptyset$ ;
3.  $acc\_P \leftarrow 0$ ;
4. while stop criterion not satisfied do
5.    $s \leftarrow$  Construction ( $\alpha$ );
6.    $s' \leftarrow$  Localsearch ( $s$ );
7.   if  $s'$  is better than  $s^*$  then
8.      $s^* \leftarrow s'$ ;
9.   endif
10.  if  $s'$  satisfies the pool acceptance criterion then
11.     $acc\_P \leftarrow acc\_P + 1$ ;
12.     $P [acc\_P] \leftarrow s'$ ;
13.    if  $acc\_P = size\_P$  then
14.       $s'' \leftarrow$  Intensification ( $P$ , size_P);
15.      if  $s''$  is better than  $s^*$  then
16.         $s^* \leftarrow s''$ ;
17.      endif
18.     $P \leftarrow \emptyset$ ;
19.     $acc\_P \leftarrow 0$ ;
20.  endif
21. endif
22. endwhile
23. return  $s^*$ ;
end HGRASP

```

Figure 6. Procedure HGRASP.

Table 4. Comparative results (300 seconds).

Instance	current	Best solution values						Average solution values					
		GRASP		HGRASP		HGRASP+LS		GRASP		HGRASP		HGRASP+LS	
ANGI	216	216	0.0%	216	0.0%	216	0.0%	216.0	0.0%	216.0	0.0%	216.0	0.0%
GRIS	172	172	0.0%	172	0.0%	172	0.0%	172.0	0.0%	172.0	0.0%	172.0	0.0%
TENU	682	682	0.0%	682	0.0%	682	0.0%	682.0	0.0%	682.0	0.0%	682.0	0.0%
ETHE	372	372	0.0%	372	0.0%	372	0.0%	372.0	0.0%	372.0	0.0%	372.0	0.0%
ROPA	325	325	0.0%	325	0.0%	325	0.0%	325.8	0.3%	326.0	0.3%	325.4	0.1%
GOLO	496	497	0.2%	497	0.2%	496	0.0%	498.0	0.4%	497.2	0.2%	496	0.0%
CARP	548	549	0.2%	549	0.2%	548	0.0%	550.4	0.4%	549.0	0.2%	548.0	0.0%
SCHU	759	761	0.3%	759	0.0%	759	0.0%	761.4	0.3%	760.2	0.2%	759.2	0.0%
TST01	547	549	0.4%	548	0.2%	547	0.0%	549.4	0.4%	549.6	0.5%	547.8	0.2%
TST02	1361	1363	0.2%	1362	0.1%	1360	-0.1%	1363.6	0.2%	1362.6	0.1%	1360.8	0.0%
TST03	836	840	0.5%	839	0.4%	836	0.0%	840.6	0.6%	840.8	0.6%	837.4	0.2%
TST04	590	592	0.5%	592	0.5%	592	0.5%	594.8	0.8%	595.4	0.9%	593.0	0.5%
TST05	792	795	0.4%	795	0.4%	791	-0.1%	795.8	0.5%	795.8	0.5%	791.0	-0.1%
TST06	600	603	0.5%	602	0.3%	599	-0.2%	604.2	0.7%	604.4	.7%	599.0	-0.2%
TST07	1274	1283	0.7%	1282	0.6%	1276	0.2%	1284.2	0.8%	1284.2	0.8%	1280.0	0.5%
TST08	862	868	0.7%	867	0.6%	864	0.2%	870.6	1.0%	869.8	0.9%	864.2	0.3%
TST09	1147	1155	0.7%	1151	0.3%	1148	0.1%	1155.4	0.7%	1152.2	0.4%	1148.0	0.1%
TST10	722	730	1.1%	730	1.1%	726	0.6%	730.0	1.1%	730.4	1.2%	726.2	0.6%
TST11	546	551	0.9%	548	0.4%	548	0.4%	552.0	1.1%	548.0	0.4%	548.4	0.4%
TST12	1224	1232	0.6%	1234	0.8%	1230	0.5%	1233.8	0.8%	1235.2	0.9%	1231.4	0.6%
TST13	1524	1533	0.6%	1534	0.7%	1526	0.1%	1534.2	0.7%	1535.0	0.7%	1527.2	0.2%
TST14	1171	1178	0.6%	1179	0.7%	1175	0.3%	1179.4	0.7%	1179.8	0.8%	1175.0	0.3%
TST15	758	769	1.5%	765	0.9%	764	0.8%	769.8	1.6%	769.8	1.6%	764.6	0.9%
TST16	537	547	1.9%	545	1.5%	541	0.7%	547.4	1.9%	545.0	1.5%	541.8	0.9%
TST17	2468	2474	0.2%	2469	0.0%	2467	-0.1%	2478.8	0.4%	2470.8	0.1%	2469.2	0.1%
TST18	1531	1540	0.6%	1540	0.6%	1540	0.6%	1542.8	0.8%	1542.8	0.8%	1542.6	0.8%
TST19	1024	1032	0.8%	1032	0.8%	1029	0.5%	1034.4	1.0%	1032.8	0.9%	1029.8	0.6%
TST20	671	676	0.8%	676	0.8%	676	0.8%	677.6	1.0%	681.2	1.5%	676.4	0.8%
			0.5%		0.4%		0.2%		0.7%		0.6%		0.3%

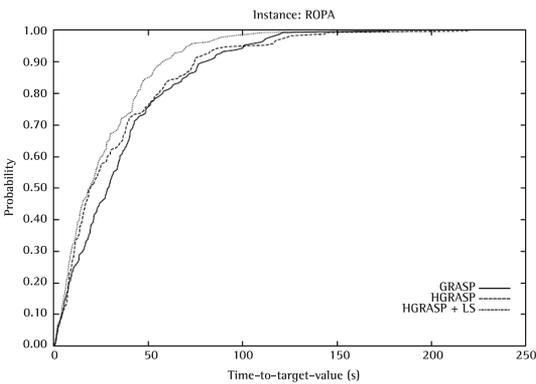


Figure 7. Time-to-target experiment (instance ROPA).

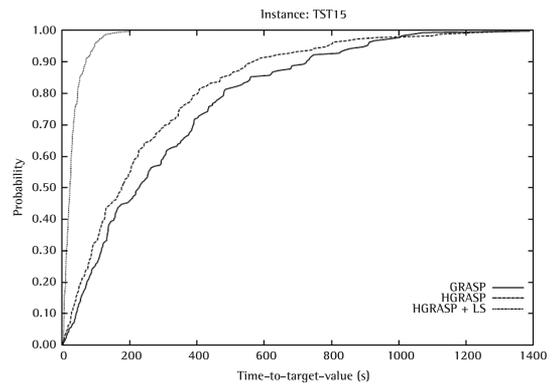


Figure 8. Time-to-target experiment (instance TST15).

The obtained results show that the heuristics HGRASP and HGRASP+LS systematically find better solutions than the traditional GRASP heuristic in smaller computation times. The heuristic HGRASP+LS is the most robust, presenting a much smaller variability in the computation times over different runs.

According to Table 2, for the ROPA instance, the Average Solution Values obtained by all approaches

are smaller than 327, results achieved in less than 38.9 seconds (Table 1). The Figure 7 shows that, in 50 seconds, the probability of obtaining a solution with a cost less or equal to 327 is smaller than 90%. This shows how important is the observation of statistical metrics. The same occurs in the Figure 8. In the Tables 2 and 3, the GRASP is better than HGRASP, for the TST15 instance; but, in time-to-target plot, this does not occur.

4. Conclusion

In this paper, we proposed hybrid GRASP heuristics for the phylogeny problem and showed how an intensification strategy, based on path-relinking and genetic algorithm, can be used to improve the performance of the greedy randomized search.

Computational experiments were done with 28 benchmark instances for the phylogeny problem. The proposed heuristics, HGRASP and HGRASP+LS, systematically found better solution than traditional GRASP heuristic in smaller computational times. They also found better solutions for most benchmark instances.

Comparing with the currently best known solutions, the proposed heuristics found solutions very close to them. The heuristic HGRASP+LS, running 300 seconds, improved the best known solutions for four benchmark instances, matched the best results for ten benchmark instances and found very close solutions for the others (average *gap* = 0.2%).

References

- AIEX, R. M.; RESENDE, M. G. C.; RIBEIRO, C. C. *Probability distribution of solution time in GRASP: An experimental investigation*. *Journal of Heuristics*, v. 8, p. 343-373, 2002. <http://dx.doi.org/10.1023/A:1015061802659>
- ANDREATTA, A. A. *Uma arquitetura abstrata de domínio para o desenvolvimento de heurísticas de busca local com uma aplicação ao problema da filogenia*. 1998. Tese (Doutorado em Informática)-Pontifícia Universidade Católica, Rio de Janeiro, 1998.
- ANDREATTA, A. A.; RIBEIRO, C. C. Heuristics for the phylogeny problem. *Journal of Heuristics*, v. 8, p. 429-447, 2002. <http://dx.doi.org/10.1023/A:1015439913121>
- AYALA, F. J. The myth of Eve: Molecular biology and human origins. *Science*, v. 270, p. 1930-1939, 1995. PMID:8533083. <http://dx.doi.org/10.1126/science.270.5244.1930>
- BODLAENDER, H.; FELLOWS, M.; WARNOW, T. Two strikes against the perfect phylogeny problem. In: INTERNATIONAL CONFERENCE ON ALGORITHMS, LANGUAGES AND PROGRAMMING, 1992, Wien. *Proceedings...* Springer-Verlag, 1992. p. 273-283.
- COTTA, C. Scatter search with path relinking for phylogenetic inference. *European Journal of Operational Research*, v. 169, p. 520-532, 2006. <http://dx.doi.org/10.1016/j.ejor.2004.08.013>
- DAY, W. H. E.; JOHNSON, D. S.; SANKOFF, D. The computational complexity of inferring rooted phylogenies by parsimony. *Mathematical Biosciences*, v. 81, p. 33-42, 1986. [http://dx.doi.org/10.1016/0025-5564\(86\)90161-6](http://dx.doi.org/10.1016/0025-5564(86)90161-6)
- EDWARDS, A.; CAVALLI-SFORZA, L. Reconstruction of evolutionary trees. *Phenetic and Phylogenetic Classification*, v. 6, p. 67-76, 1964.
- FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, p. 109-133, 1995. <http://dx.doi.org/10.1007/BF01096763>
- FESTA, P.; RESENDE, M. G. C. An annotated bibliography of GRASP - part I: Algorithms. *International Transactions in Operational Research*, v. 16, p. 1-24, 2009a. <http://dx.doi.org/10.1111/j.1475-3995.2009.00663.x>
- FESTA, P.; RESENDE, M. G. C. An annotated bibliography of GRASP - part II: Applications. *International Transactions in Operational Research*, v. 16, p. 131-172, 2009b. <http://dx.doi.org/10.1111/j.1475-3995.2009.00664.x>
- FOULDS, L. R.; GRAHAM, R. L. The Steiner problem in phylogeny is NP-Complete. *Advances in Applied Mathematics*, v. 3, p. 43-49, 1982a. [http://dx.doi.org/10.1016/S0196-8858\(82\)80004-3](http://dx.doi.org/10.1016/S0196-8858(82)80004-3)
- FOULDS, L. R.; GRAHAM, R. L. Unlikelihood that minimal phylogenie for a realistic biological study can be constructed in reasonable computational time. *Mathematical Biosciences*, v. 60, p. 133-142, 1982b. [http://dx.doi.org/10.1016/0025-5564\(82\)90125-0](http://dx.doi.org/10.1016/0025-5564(82)90125-0)
- GLOVER, F. Tabu search and adaptive memory programming - Advances, applications and challenges. In: BARR, R. S.; HELGASON, R. V.; KENNINGTON, J. L. (Eds.). *Interfaces in Computer Science and Operations Research*. Kluwer, 1996. p. 1-75. PMID:8842811.
- GLOVER, F. Multi-start and strategic oscillation methods - Principles to exploit adaptive memory. In: LAGUNA, M.; GONZÁLES-VELARDE, J. L. (Eds.). *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*. Kluwer, 2000. p. 1-24. PMID:10757191. http://dx.doi.org/10.1007/978-1-4615-4567-5_1
- GLOVER, F.; LAGUNA, M. *Tabu Search*. Kluwer, 1997. <http://dx.doi.org/10.1007/978-1-4615-6089-0>
- GLOVER, F.; LAGUNA, M.; MARTÍ, R. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, v. 39, p. 653-684, 2000.
- HENNIG, W. *Phylogenetic systematics*. Urbana: University of Illinois Press, 1966.
- KITCHING, I. J. et al. *Cladistics: The theory and practice of parsimony analysis*. London: Oxford University Press, 1998.
- LAGUNA, M.; MARTÍ, R. GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, v. 11, p. 44-52, 1999. <http://dx.doi.org/10.1287/ijoc.11.1.44>
- LOURENÇO, H. R.; MARTIN, O.; STUETZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. (Eds.). *Handbook of Metaheuristics*. Norwell: Kluwer Academic Publishers, 2002. chapt. 7, p. 321-353.
- LUCKOW, M.; PIMENTEL, R. A. An empirical comparison of numerical Wagner computer programs. *Cladistics*, v. 1, p. 47-66, 1985. <http://dx.doi.org/10.1111/j.1096-0031.1985.tb00410.x>
- MAURI, G. R.; LORENA, L. A. N. Uma nova abordagem para o problema dial-a-ride. *Produção*, v. 19, n. 1, p. 41-54, 2009. <http://dx.doi.org/10.1590/S0103-65132009000100004>
- NERI, F.; COTTA, C.; MOSCATO, P. *Handbook of Memetic Algorithms*. Springer, 2012. v. 379. (Series: Studies in Computational Intelligence).
- PENNY, D.; FOULDS, L. R.; HENDY, M. D. Testing the theory of evolution by comparing phylogenetic trees constructed from five different protein sequences. *Nature*, v. 247, p. 197-200, 1982. <http://dx.doi.org/10.1038/297197a0>

- PLATNICK, N. I. An empirical comparison of microcomputer parsimony programs. *Cladistics*, v. 3, p. 121-144, 1987. <http://dx.doi.org/10.1111/j.1096-0031.1987.tb00502.x>
- PLATNICK, N. I. An empirical comparison of microcomputer parsimony programs II. *Cladistics*, v. 5, p. 145-161, 1989. <http://dx.doi.org/10.1111/j.1096-0031.1989.tb00561.x>
- RESENDE, M. G. C.; RIBEIRO, C. C. Greedy randomized adaptive search procedures. In: GLOVER, F.; KOCHENBERGER, G. (Eds.). *Handbook of Metaheuristics*. Kluwer, 2003.
- RESENDE, M. G. C.; RIBEIRO, C. C. GRASP with path-relinking: Recent advances and applications. In: IBARAKI, T.; NONOBE, K.; YAGIURA, M. (Eds.). *Metaheuristics: Progress as Real Problem Solvers*. Springer, 2005. p. 29-63. http://dx.doi.org/10.1007/0-387-25383-1_2
- RIBEIRO, C. C.; RESENDE, M. G. C. Path-relinking intensification methods for stochastic local search algorithms. *Journal of Heuristics*, v. 18, p. 193-214, 2012. <http://dx.doi.org/10.1007/s10732-011-9167-1>
- RIBEIRO, C. C.; VIANNA, D. S. A hybrid genetic algorithm for the phylogeny problem using path-relinking as a progressive crossover strategy. *International Transactions in Operational Research*, v. 16, p. 641-657, 2009. <http://dx.doi.org/10.1111/j.1475-3995.2009.00699.x>
- RIBEIRO, C. C.; VIANNA, D. S. A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure. *International Transactions in Operational Research*, v. 12, p. 325-338, 2005. http://dx.doi.org/10.1111/j.1475-3995.2005.498_1.x
- SCHRAGE, L. A more portable FORTRAN random number generator. *ACM Transactions on Mathematical Software*, v. 5, p. 132-138, 1979. <http://dx.doi.org/10.1145/355826.355828>
- SOBER, E. Parsimony, likelihood and the principle of the common cause. *Philosophy of Science*, v. 54, p. 465-469, 1987. <http://dx.doi.org/10.1086/289394>
- SWOFFORD, D. L. *Wagner procedure program*. Champaign: Illinois Natural History Survey, 1982. PMID:20372399.
- SWOFFORD, D. L.; OLSEN, G. Phylogeny reconstruction. In: HILLIS, D. M.; MORITZ, C. (Eds.). *Molecular Systematics*. Sinauer, 1990. p. 411-501.
- SWOFFORD, D. L. et al. *Phylogeny inference*. In: HILLIS, D. M.; MORITZ, C.; MABLE, B. K. (Eds.). *Molecular Systematics*. 2nd ed. Sinauer, 1996. p. 407-514.
- VIANNA, D. S. et al. Parallel strategies for a multi-criteria GRASP algorithm. *Produção*, v. 17, p. 1-12, 2007. <http://dx.doi.org/10.1590/S0103-65132007000100006>