A genetic algorithm for the phylogeny problem using an optimized crossover strategy based on path-relinking

Celso C. Ribeiro* and Dalessandro S. Vianna**

Department of Computer Science, Catholic University of Rio de Janeiro, Rio de Janeiro, RJ 22453-900, Brazil

Abstract. A phylogenetic tree relates taxonomic units using their similarities over a set of characters. We propose a new genetic algorithm for the problem of building a phylogenetic tree under the parsimony criterion. It makes use of an innovative optimized crossover strategy which is an extension of the path-relinking intensification technique originally applied in the context of implementations of other metaheuristics such as tabu search and GRASP. Computational results are reported for benchmark instances and randomly generated test problems, illustrating the effectiveness of the genetic algorithm.

Keywords: Phylogeny problem, genetic algorithms, path-relinking, optimized crossover.

1 Introduction

A phylogeny is a tree that relates taxons [16, 17] using their similarities over a set of independent characters. Each taxon is defined by a set of characters. Binary characters are those who have only two possible states. Instances of the phylogeny problem with binary characters are defined by 0-1 matrices, in which each element (i, j) corresponds to the state of character j within taxon i.

Each state change along a branch of a phylogenetic tree is counted as an evolutionary step. The parsimony criterion states that the best phylogeny is the one that can be explained by the minimum number of evolutionary steps [9, 15]. The phylogeny problem is that of finding a phylogenetic tree with the minimum number of evolutionary steps.

A phylogenetic tree s for the operational taxons under analysis belongs to the set S of unrooted trees with n leaves (each of them corresponding to an operational taxon) and all internal nodes with degree three. Let $f : S \rightarrow \mathbb{R}$ be a function which associates each phylogeny $s \in S$ to its parsimony value. The phylogeny problem consists in finding a phylogeny $s^\star \in S$ such that $f(s^\star) = \min_{s \in S} f(s)$. Polynomial algorithms running in $O(mn)$ time for the computation of the parsimony value of a given phylogeny are described in [5–7], where n

* celso@inf.puc-rio.br

** vianna@inf.puc-rio.br

is the number of operational characters and m the number of binary characters. Andreatta and Ribeiro [2] compared the computational results obtained by a variety of heuristics on a set of eight benchmark problems. Ribeiro and Vianna [13] proposed a GRASP heuristic which improved the best known solutions for some benchmark instances.

The structure of the new genetic algorithm is described in Section 2. A new optimized crossover strategy based on path-relinking is reported in Section 3. Computational results for benchmark instances from the literature and for randomly generated test problems are reported in Section 4. Concluding remarks are made in the last section.

2 Genetic algorithm: Population dynamics

A genetic algorithm is a population-based metaheuristic for combinatorial problems. A population is a set of solutions which are combined (through crossover) and perturbed (by mutation) to produce a new generation of solutions. Attributes of high-quality solutions have a greater probability to be passed down to the next generation. This process is repeated over many generations as long as the quality of the solutions in the new population improves over time.

The initial population of the genetic algorithm proposed in this work is formed by 100 solutions built by the greedy randomized algorithm `Gstep_wR` [2].

The population is partitioned into three sets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. The best solutions are kept in $\mathcal{A}$, while the worst ones are in $\mathcal{C}$. Class $\mathcal{A}$ is formed by the 30% best solutions of each generation, while $\mathcal{C}$ is formed by the 20% worst. All solutions in $\mathcal{A}$ are promoted to the next generation. Solutions in $\mathcal{B}$ are replaced by crossover of one parent from $\mathcal{A}$ with another from $\mathcal{B} \cup \mathcal{C}$ using the *random keys* crossover scheme of Bean [3], as already successfully used in [4]. All solutions in $\mathcal{C}$ are replaced by new ones created by the same algorithm used to build the initial population.

In the random keys scheme, crossover is carried out on a selected pair of parent solutions to produce an offspring solution. Each selected pair consists of an elite parent and a non-elite parent. The elite parent is selected, at random, uniformly from solutions in $\mathcal{A}$, while the non-elite parent is selected from those in $\mathcal{B} \cup \mathcal{C}$. We use a path-relinking strategy to generate the offspring, as described in Section 3.

This genetic algorithm also applies periodically at every seven generations a local improvement procedure to all solutions obtained by crossover. This local search procedure is a first-improving strategy using the neighborhood SPR (Subtree Pruning and Regrafting), already described in [13].

3 Optimized crossover by path-relinking

Path-relinking is an intensification strategy originally proposed by Glover [8] to explore trajectories between elite solutions obtained by tabu search or scatter search. Using one or more elite solutions, paths in the solution space leading to other elite solutions are explored in the search for better solutions. To generate paths, moves are selected to introduce attributes in the current solution that appear in the elite guiding solution.

Extensions, improvements, and successful applications of path-relinking in the context of GRASP implementations have been reported in the literature [10–12]. In this work, we perform the crossover operation using a path-relinking strategy. Given two parent solutions s_1 and s_2 ramdomly selected, the former from class $\mathcal{A}$ and the latter from $\mathcal{B} \cup \mathcal{C}$, bidirectional path-relinking between them is performed [11] and the best overall solution found is returned.

This mechanism is an extension of the traditional crossover operation. Instead of producing only one offspring, it investigates a bunch of solutions sharing the characteristics of their parents. The solution found by path-relinking is the best offspring which can be obtained by a conventional crossover operation.

Path-relinking between two solutions s_1 and s_2 is performed as follows. In the first phase, the binary tree s_1 is selected as the initial solution and s_2 as the guiding one. Let n_1 and n_2 be the current nodes being explored respectively in s_1 and s_2, which are initially set as their roots. Let l_1 and r_1 be the subtrees of s_1 rooted at each of the children of n_1 and consider the subtrees l_2 and r_2 of s_2 rooted at each of the children of n_2. Denote by L_1, R_1, L_2, and R_2, respectively, the sets of operational taxons which are leaves of l_1, r_1, l_2, and r_2.

The next step consists in computing which of the subtrees L_2 or R_2 is more similar to L_1, in terms of their number of leaves corresponding to the same taxons. Without loss of generality, suppose that $|L_1 \cap L_2| > |L_1 \cap R_2|$, i.e., L_1 share more taxons in common with L_2 than with R_2. Subtrees l_1 and l_2 will be associated one to the other.

For each taxon $v \in L_1 \setminus L_2$ incorrectly placed in subtree l_1, the algorithm computes its reconnection cost at each possible edge of subtree r_1 in time $O(1)$, as described in [13]. Analogously, for each taxon $v \in R_1 \setminus R_2$ incorrectly placed in subtree l_1, the algorithm also computes its reconnection cost at each possible edge of subtree l_1. Let $v^* \in L_1 \setminus L_2 \cup R_1 \setminus R_2$ be the node with the smaller reconnection cost (i.e., with the greatest decrease – or smallest increase – in solution value). Node v^* is eliminated from its current subtree and reconnected in the appropriate position to the other subtree. The current solution is updated. These steps are repeated until $L_1 = L_2$ and $R_1 = R_2$.

Once the left and right subtrees of n_1 contain the same nodes respectively in the left and right subtrees of n_2, these steps are recursively applied to the roots of subtrees l_1 and l_2 and to those of r_1 and r_2. This procedure is repeated with the roles of s_1 and s_2 interchanged. The path-relinking procedure returns the best solution obtained by crossover.

4 Computational results

The computational experiments were performed on a 2 GHz Pentium IV processor with 512 Mbytes of RAM memory. Heuristic `AG+PR` described in the previous sections was 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 [14].

We compare the new heuristic `AG+PR` with algorithm `GRASP+VND` proposed in [13] using eight benchmark instances and 20 randomly generated instances.

In the first experiment, the same computation time was given to each algorithm. Ten runs of 1,000 seconds each were performed for each instance. The computational results are reported in Table 1. For each instance, we first give its number of taxons (n) and its number of characteristics (m). Next, we give the average and the best solution values obtained over ten runs of each algorithm. We indicate in boldface whenever one of the algorithms found strictly better results than the other. The new heuristic obtained strictly better average solution values for five out of the eight benchmark instances and for all randomly generated instances. `AG+PR` also found strictly best solutions for two out of the eight benchmark instances and for all but one randomly generated instance.

In the second experiment, we compare the robustness of both algorithms using instance SCHU. One hundred independent runs for each algorithm were performed. Execution was terminated when a solution of value less than or equal to a difficult target specified as 760 was found, which corresponds to the best known solution before that reported by Ribeiro and Vianna [13] (the currently best known solution value for this instance is 759). The empirical probability distribution for the time to target solution value is plotted in Figure 2. To plot the empirical distribution for each algorithm, we followed the procedure described in [1]. We associate with the i-th smallest running time t_i a probability $p_i = (i - \frac{1}{2})/100$, and plot the points $z_i = (t_i, p_i)$, for $i = 1, \ldots, 100$. This plot shows that heuristic `AG+PR` is able to find solutions with the same value of those obtained by `GRASP+VND` in much smaller computation times. The new heuristic is more robust.

5 Concluding remarks

Approximate and exact (for small problems) algorithms for the computation of phylogenetic trees are dispersed through the scientific literature. We proposed in this paper a genetic algorithm for the phylogeny problem. This heuristic uses an innovative strategy based on path-relinking to implement the crossover operation, which is also combined with local search. This strategy can be easily extended to implementations of genetic algorithms for solving other problems.

We also notice that this genetic algorithm outperformed the best algorithms currently available [2, 13] in terms of solution quality and robustness (better solutions in smaller computation times).

References

1. R.M. Aiex, M.G.C. Resende, and C.C. Ribeiro, "Probability distribution of solution time in GRASP: An experimental investigation", *Journal of Heuristics* 8 (2002), 343–373.

			Average solution value (over ten runs)		Best solution value (over ten runs)	
Instance	n	m	`GRASP+VND`	`AG+PR`	`GRASP+VND`	`AG+PR`
GRIS	47	93	172.0	172.0	172	172
ANGI	49	59	216.0	216.0	216	216
TENU	56	179	682.0	682.0	682	682
ETHE	58	86	372.8	**372.4**	372	372
ROPA	75	82	326.0	**325.8**	325	325
GOLO	77	97	498.2	**496.2**	497	**496**
SCHU	113	146	761.0	**759.2**	759	759
CARP	117	110	552.4	**548.6**	550	**548**
TST01	45	61	550.8	**549.6**	549	549
TST02	47	151	1367.2	**1363.6**	1361	**1358**
TST03	49	111	843.8	**840.6**	843	**838**
TST04	50	97	599.2	**595.0**	598	**592**
TST05	52	75	796.8	**794.0**	793	**790**
TST06	54	65	606.6	**605.4**	605	**603**
TST07	56	143	1288.6	**1280.6**	1283	**1276**
TST08	57	119	873.0	**867.4**	868	**863**
TST09	59	93	1159.2	**1154.2**	1156	**1150**
TST10	60	71	732.0	**728.6**	730	**725**
TST11	62	63	554.8	**546.8**	554	**544**
TST12	64	147	1242.0	**1233.0**	1237	**1229**
TST13	65	113	1532.4	**1530.6**	1529	**1526**
TST14	67	99	1182.0	**1177.4**	1180	**1174**
TST15	69	77	774.8	**766.4**	769	**765**
TST16	70	69	550.6	**547.6**	547	**545**
TST17	71	159	2479.0	**2470.8**	2475	**2468**
TST18	73	117	1554.6	**1548.2**	1548	**1542**
TST19	74	95	1041.0	**1033.0**	1035	**1028**
TST20	75	79	685.4	**678.8**	682	**676**

Fig. 1. Comparative results for ten runs of algorithms `GRASP+VND` and `AG+PR`.

2. A.A. Andreatta and C.C. Ribeiro, "Heuristics for the phylogeny problem", *Journal of Heuristics* 8 (2002), 429–447.
3. J.C. Bean, "Genetic algorithms and random keys for sequencing and optimization", *ORSA Journal on Computing* 6 (1994), 154–160.
4. L.S. Buriol, M.G.C. Resende, C.C. Ribeiro, and M. Thorup, "A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing", submitted for publication, 2003.
5. J.S. Farris, "Methods for computing Wagner trees", *Systematic Zoology* 19 (1970), 83– 92.
6. W.M. Fitch, "Towards defining the course of evolution: Minimum chances for a specific tree topology", *Systematic Zoology* 20 (1971), 406–419.
7. W.M. Fitch and J.S. Farris, "Evolutionary trees with minimum nucleotide replacements from amino acid sequences", *Journal of Molecular Evolution* 3 (1974), 263–278.

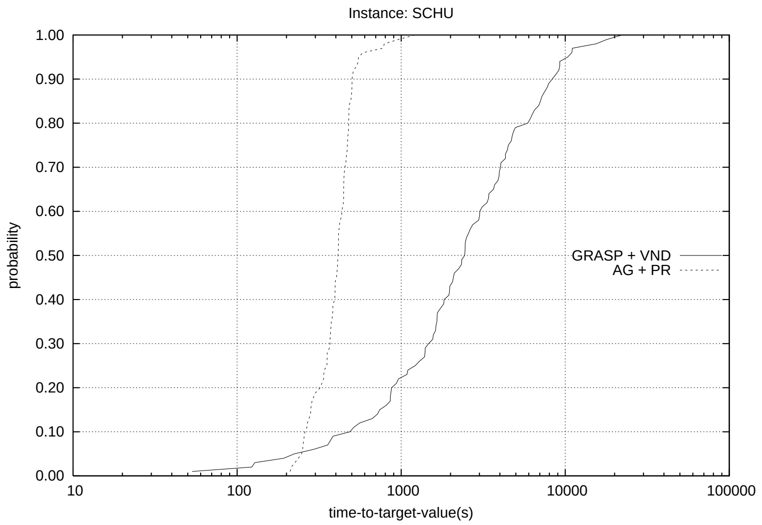

Fig. 2. Time to target solution value.

8. F. GLOVER, "Tabu search and adaptive memory programing – Advances, applications and challenges", *Interfaces in Computer Science and Operations Research* (R.S. Barr, R.V. Helgason, and J.L. Kennington, eds.), pages 1–75. Kluwer, 1996.
9. W. HENNIG, *Phylogenetic systematics*, University of Illinois Press, 1966, Urbana.
10. M.G.C. RESENDE and C.C. RIBEIRO,"GRASP with path-relinking for private virtual circuit routing", *Networks* 41 (2003), 104–114.
11. M.G.C. RESENDE and C.C. RIBEIRO, "Greedy randomized adaptive search procedures", *Handbook of Metaheuristics* (F. Glover and G. Kochenberger, eds.), Kluwer, pages 219–249, 2003.
12. M.G.C. RESENDE and C.C. RIBEIRO,"GRASP with path-relinking: Advances and applications", *Extended Abstracts of the 5th Metaheuristics International Conference*, Kyoto, 2003.
13. C.C. RIBEIRO and D.S. VIANNA, "A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure", submitted for publication, 2003.
14. L. SCHRAGE, "A more portable FORTRAN random number generator", *ACM Transactions on Mathematical Software* 5 (1979), 132-138.
15. E. SOBER, "Parsimony, likelihood and the principle of the common cause", *Philosophy of Science* 54 (1987), 465–469.
16. D.L. SWOFFORD and G. OLSEN, "Phylogeny reconstruction", *Molecular systematics* (D.M. Hillis and C. Moritz, eds.), Sinauer, 1990, Sunderland.
17. E.O. WILEY, D. SIEGEL-CAUSEY, D.R. BROOKS, and V.A. FUNK, *The compleat cladist: A primer of phylogenetic procedures*, Special publication no. 19, University of Kansas, Museum of Natural History, 1991.