PATH-RELINKING INTENSIFICATION METHODS FOR STOCHASTIC LOCAL SEARCH ALGORITHMS

CELSO C. C. RIBEIRO AND MAURICIO G. C. RESENDE

Abstract. Path-relinking is major enhancement to heuristic search methods for solving combinatorial optimization problems, leading to significant improvements in both solution quality and running times. We review its fundamentals and implementation strategies, as well as advanced hybridizations with more elaborate metaheuristic schemes such as genetic algorithms and scatter search. Numerical examples are discussed and algorithms compared based on their run time distributions.

1. Introduction and motivation

We consider in this paper a combinatorial optimization problem (COP), defined by a finite ground set $E = \{1, \ldots, n\}$, a set of feasible solutions $F \subseteq 2^E$, and an objective function $f : 2^E \to \mathbb{R}$. In its minimization version, we seek a global optimum $x^* \in F$ such that $f(x^*) \leq f(x)$, $\forall x \in F$, with each solution being represented by its characteristic vector $x \in \{0,1\}^{|E|}$. The ground set E, the cost function f, and the set of feasible solutions F are defined for each specific problem.

In the case of the classical *traveling salesman problem*, the ground set E is that of all edges connecting the cities to be visited, $f(x)$ is the sum of the costs of all edges in a solution x, and F is formed by all edge subsets that determine a Hamiltonian cycle.

The *Ising model* has been a subject of great interest in Physics. In spite of its simplicity, it retains many of the characteristics of real systems. As the temperature decreases, the system moves to a *ground state* of minimum energy. Finding a ground state amounts to solve an equivalent *max-cut* combinatorial optimization problem (Barahona, 1994). Given an undirected graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ is the set of vertices and E is the set of edges, and weights w_{ij} associated with the edges $(i, j) \in E$, the max-cut problem consists in finding a subset of vertices S such that the weight of the cut $(S, \bar{S})$ given by

$$w(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} w_{ij}$$

is maximized. Applications are found in VLSI design and statistical physics, see e.g. Barahona et al. (1988), Chang and Du (1987), Chen et al. (1983), and Pinter (1984) among others. The max-cut problem can be formulated as the following integer quadratic program:

$$\max \frac{1}{2} \sum_{1 \leq i < j \leq n} w_{ij}(1 - y_i y_j)$$

Date: April 1, 2010.
Key words and phrases. Path-relinking, metaheuristics, hybrid metaheurisics.

subject to

$$y_i \in \{-1, 1\} \quad \forall\, i \in V.$$

Each set $S = \{i \in V : y_i = 1\}$ induces a cut $(S, \bar{S})$ with weight

$$w(S, \bar{S}) = \frac{1}{2} \sum_{1 \le i < j \le n} w_{ij}(1 - y_i y_j).$$

Since the decision version of the max-cut problem was proved to be NP-complete by Karp (1972), it is very unlikely that exact and efficient polynomial-time algorithms exist for handling large grids in magnetic fields.

The algorithms considered in this paper play a major role in finding quasi-optimal solutions for such hard combinatorial optimization problems, as shown in Festa et al. (2006) for the max-cut problem.

A *neighborhood* of a solution $x \in F$ is any set $N(x) \subseteq F$. Each solution $y \in N(x)$ is reachable from x by an operation called *move*. Normally, two neighbor solutions x and $y \in N(x)$ differ by only a few elements.

We define an undirected graph $G = (F, M)$ associated with the search space of COP, where the nodes in F correspond to feasible solutions and the edges in M correspond to moves in the neighborhood structure, i.e. $(x, y) \in M$ if and only if $x \in F$, $y \in F$, $x \in N(y)$, and $y \in N(x)$.

A solution x' is a *local optimum* with respect to a given neighborhood N if $f(x') \le f(x), \forall x \in N(x')$. *Stochastic local search* (SLS) algorithms are based on the exploration of solution neighborhoods, searching for improving solutions until a local optimum is found. Different high-level strategies can be implemented in stochastic local search methods to avoid entrapment or premature convergence to local minima which are not globally optimal solutions. These high-level strategies are often referred to as *metaheuristics*, which are general procedures that coordinate simple heuristics and rules to find good (often optimal) approximate solutions to computationally difficult combinatorial optimization problems. Among them, we find simulated annealing (Kirkpatrick et al., 1983), the cross-entropy method (Rubinstein and Kroese, 2004), greedy randomized adaptive search procedures (GRASP) (Festa and Resende, 2009a;b), variable neighborhood search (VNS) (Mladenović and Hansen, 1997; Hansen and Mladenović, 2002), genetic algorithms (Holland, 1975), scatter search (Glover et al., 2003), and others. These methods can be grouped in two major classes: trajectory-based or population-based.

Trajectory-based stochastic local search algorithms start from a feasible solution x^0 corresponding to a node of the search space graph $G = (F, M)$. At any iteration k, they basically search for an improving solution $x^{k+1} \in N(x^k)$ in the neighborhood of the current solution x^k, such that $f(x^{k+1}) < f(x^k)$. In the case of a *first improving strategy*, any improving solution $x^{k+1} \in N(x^k)$ may be used. If a *best improving strategy* is used, the improving solution x^{k+1} is the best in the neighborhood, i..e $f(x^{k+1}) = \min\{f(x) : x \in N(x^k)\}$. Each stochastic local search algorithm makes use of a distinct paradigm and offers different mechanisms to escape from locally optimal solutions, going beyond the first local optimum found.

On the other hand, *population-based* metaheuristics are based on the computation of samples of the solutions (or nodes) of the search space graph. Starting from an initial population $X^0 \subseteq F$, they attempt to build a new population $X^{k+1} \subseteq F$ at each iteration k, such that $\min\{f(x) : x \in X^{k+1}\} < \min\{f(x) : x \in X^k\}$.

Both trajectory-based stochastic local search algorithms and population-based metaheuristics visit a subset of *elite* or *reference* solutions of COP formed by some of its best local optima.

Good solutions for a combinatorial optimization problem often share a significant portion of their attributes. Examples of such attributes include edges and nodes of a graph, sequence positions in a schedule, subsets of a partition or a cover of an item set, membership in a subset of potential locations, or basic variables in the solutions of linear programming problems. Paths between a pair of nodes x and y in the search space graph $G = (F, M)$ traverse other solutions that share attributes contained in x and y. The underlying assumption of *path-relinking* is that undiscovered high-quality solutions can be found by exploring paths connecting previously found high-quality elite solutions. To generate the desired paths, it is only necessary to select moves that upon starting from an initiating solution x progressively introduce attributes contributed by a guiding solution y. The construction of such paths in the search space graph characteristically connects previous points in ways not achieved in the previous search history (Glover, 1999), often leading to better local optimal.

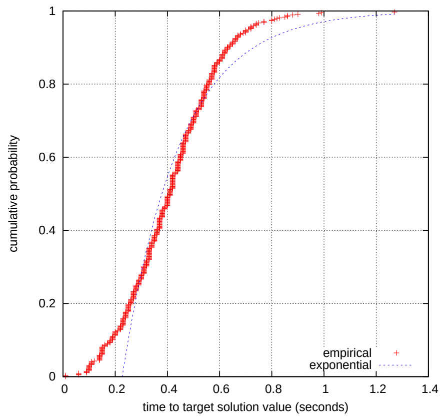
The remainder of this paper is organized as follows. In the next section, we introduce the reader to run time distributions, which are of major importance in the evaluation and comparison of stochastic local search algorithms. Next, we describe the template of the basic path-relinking algorithm. We review its mechanics and different strategies for generating paths to, from, or between elite solutions. Hybridizations of path-relinking with different metaheuristics such as GRASP, VNS, genetic algorithms, and scatter search are reviewed in the following. Illustrative numerical results profiling the improvements obtained with the use of path-relinking are also reported along the paper. Concluding remarks are drawn in the last section.

2. Run time distributions

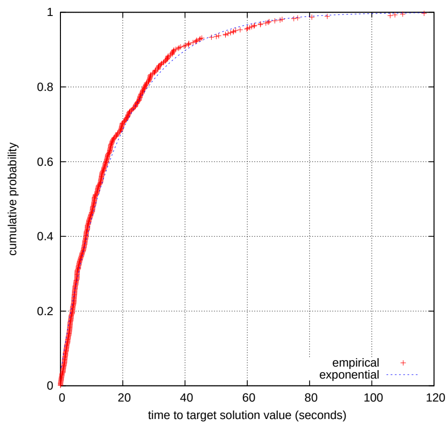
Run time distributions or time-to-target plots display on the ordinate axis the probability that an algorithm will find a solution at least as good as a given target value within a given running time, shown on the abscissa axis. Time-to-target plots were first used by Feo et al. (1994). Run time distributions have been advocated by (Hoos and Stützle, 1998; Hoos and Stützle, 1998) as a way to characterize the running times of stochastic algorithms for combinatorial optimization.

Aiex et al. (2007) describe a perl program to create time-to-target plots for measured times that are assumed to fit a shifted exponential distribution, closely following Aiex et al. (2002). Such plots are very useful in the comparison of different algorithms or strategies for solving a given problem and have been widely used as a tool for algorithm design and comparison.

Basically, to plot the empirical run time distribution of a given stochastic local search algorithm, we fix a solution target value and run each algorithm N times, recording the running time when a solution with cost at least as good as the target value is found. For each algorithm, we associate with the i-th sorted running time t_i a probability $p_i = (i - \frac{1}{2})/N$ and plot the points $z_i = (t_i, p_i)$, for $i = 1, \ldots, N$. Typically, we consider a sample of $N = 200$ runs of each algorithm to be evaluated.



(a) Algorithm A_1



(b) Algorithm A_2

FIGURE 1. Run time distributions for algorithms A_1 and A_2.

To illustrate the use of run time distributions in the comparison of two stochastic local search algorithms, we consider algorithms A_1 and A_2 for solving the same problem. Figure 1 depicts the run time distributions of each algorithm, obtained after $N = 500$ runs with different seeds.

To further explore the comparison of the two algorithms, their run time distributions are superimposed in Figure 2. Since the run time distribution of algorithm A_1 is far to the left of that of algorithm A_2, we conclude that the former performs much better than the latter, since A_1 finds same quality solutions as A_2 in much smaller running times. We denote by X_1 (resp. X_2) the continuous random variable representing the time needed by algorithm A_1 (resp. A_2) to find a solution for the problem under consideration. The tool provided in Ribeiro et al. (2009) is used to compute the probability that the running time of algorithm A_1 is smaller than or equal to that of A_2 and we get $Pr(X_1 \leq X_2) = 0.943516$, consistent with Figure 2.

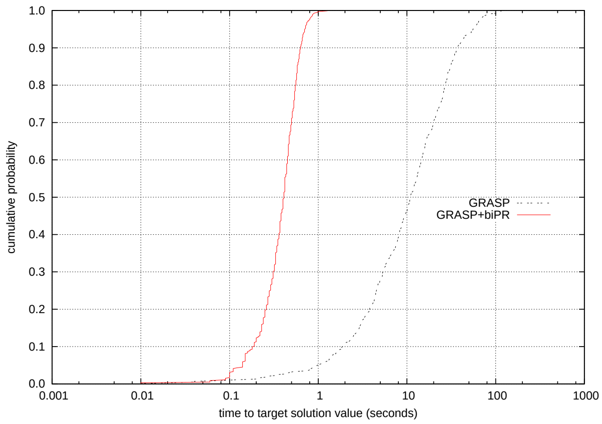


FIGURE 2. Superimposed run time distributions of algorithms A_1 and A_2.

3. TEMPLATE AND MECHANICS OF PATH-RELINKING

Path-relinking was originally proposed by Glover (1996) as an intensification strategy to explore trajectories connecting elite solutions obtained by tabu search or scatter search (Glover, 2000; Glover and Laguna, 1997; Glover et al., 2000). In the remainder of this section, we focus on path-relinking, including its template and mechanics, implementation issues, randomization, the use of elite sets to hybridize path-relinking with other heuristic methods, and evolutionary path-relinking.

We consider the undirected search space graph $G = (F, M)$ associated with a combinatorial optimization problem: $(x, y) \in M$ if and only if $x \in F$, $y \in F$, $x \in N(y)$, and $y \in N(x)$. Path-relinking is usually carried out between two solutions in F: one is the *initial solution*, while the other is the *guiding solution*. One or more paths in the search space graph connecting these solutions are explored in the search for better solutions. Local search may be applied to the best solution in each of these paths, since there is no guarantee that this solution is locally optimal.

Let $x \in F$ be a node on the path between an initial solution x and a guiding solution $g \in F$. Not all solutions in the neighborhood $N(x)$ are allowed to follow x on the path from x to g. We restrict the choice to those solutions in $N(x)$ that are more similar to g than x is. We denote by $N_g(x)$ this restricted neighborhood. This is accomplished by selecting moves from x that introduce attributes contained in the guiding solution g. Therefore, path-relinking may be viewed as a strategy that seeks to incorporate attributes of high quality solutions (i.e. the guiding solutions), by favoring these attributes in the selected moves. After evaluating each potential move, the most common strategy is to select the move resulting in the best-quality restricted neighbor of x. The restricted neighbors of x are those that incorporate an attribute of the guiding solution x that is not present in x.

3.1. **Basic implementation strategies for path-relinking.** Several strategies for path-relinking have been considered and combined in recent implementations. These include forward, backward, back-and-forward, mixed, truncated, greedy randomized adaptive, and evolutionary path-relinking. All these strategies involve trade-offs between computation time and solution quality.

Suppose that path-relinking is applied to a minimization problem between solutions x^1 and x^2 such that $f(x^1) \leq f(x^2)$, where $f(\cdot)$ denotes the objective function to be minimized. In *forward* path-relinking, the initial and guiding solutions are set, respectively, to $x = x^2$ and $g = x^1$. Conversely, in *backward* path-relinking, we set $x = x^1$ and $g = x^2$. In *back-and-forward* path-relinking, backward path-relinking is applied first, followed by forward path-relinking. Path-relinking explores the neighborhood of the initial solution more thoroughly than the neighborhood of the guiding solution because, as it moves along the path, the size of the restricted neighborhood progressively decreases. Consequently, backward path-relinking tends to do better than forward path-relinking. Back-and-forward path-relinking does at least as well as either backward or forward path-relinking, but takes about twice as long to compute since two paths are traversed. Computational experiments in Aiex et al. (2005) and Resende and Ribeiro (2003) have shown that backward path-relinking usually outperforms forward path-relinking, while back-and-forward path-relinking finds solutions at least as good as forward or backward path-relinking, but at the expense of longer run times.

1. **Algorithm** Path-Relinking(x^1, x^2)
2. $\Delta \leftarrow \{j = 1, \ldots, n \ : \ x^1_j \neq x^2_j\}$;
3. $x^* \leftarrow \text{argmin}\{f(x^1), f(x^2)\}$;
4. $f^* \leftarrow \min\{f(x^1), f(x^2)\}$;
5. $x \leftarrow x^1$;
6. **while** $|\Delta| > 1$ **do**
7. $\quad \ell^* \leftarrow \text{argmin}\{f(x \oplus \ell) \ : \ \ell \in \Delta\}$;
8. $\quad \Delta \leftarrow \Delta \setminus \{\ell^*\}$;
9. $\quad x \leftarrow x \oplus \ell^*$;
10. $\quad$ **if** $f(x) < f^*$ **then do**;
11. $\qquad x^* \leftarrow x$;
12. $\qquad f^* \leftarrow f(x)$;
13. $\quad$ **end**;
14. $\quad x^1 \leftarrow x^2$;
15. $\quad x^2 \leftarrow x^2$;
16. **end**;
17. Apply local search to improve best solution x^*;
18. **return** x^*;
19. **end**

FIGURE 3. Template of a mixed path-relinking algorithm for minimization problems.

In applying *mixed path-relinking* (Glover et al., 2004; Resende and Ribeiro, 2010a; Ribeiro and Rosseti, 2009) between two feasible solutions x^1 and x^2, two paths are started simultaneously: one at x^1 leading to x^2 and the other at x^2 leading to x^1. These two paths meet at some feasible solution, thus connecting x^1 and x^2 with a single path. Figure 3 shows a template for a mixed path-relinking algorithm between solutions x^1 and x^2 of a minimization problem. The set $\Delta = \{j = 1, \ldots, n \ : \ x^1_j \neq x^2_j\}$ of positions in which x^1 and x^2 differ is computed in line 2. In the case of combinatorial optimization problems whose solutions are represented by 0-1 vectors, $|\Delta|$ is the *Hamming distance* between x^1 and x^2. The best solution, x^*, among x^1 and x^2 and its cost, $f^* = f(x^*)$, are determined in lines 3 and 4, respectively. The current path-relinking solution, x, is initialized to x^1 in line 5. The loop in lines 6 to 16 progressively determines the next solution in the path connecting x^1 and x^2, until the entire path is traversed. For every position $\ell \in \Delta$, we define $x \oplus \ell$ to be the solution obtained from x by changing its ℓ-th position by that of x^2. Line 7 determines the component ℓ^* of Δ for which $x \oplus \ell$ results in the least-cost restricted neighbor. This component is removed from Δ in line 8 and the current solution is updated in line 9 by changing the value of its ℓ-th position. If the test in line 10 detects that the new solution x improves the best solution x^* in the path, then x^* and its cost are updated in lines 11 and 12, respectively. The roles of the starting and target solutions are swapped in lines 14 and 15 to implement the mixed path-relinking strategy and a new iteration starts. If $|\Delta| = 0$, then a local search procedure is applied in line 17 to the best solution in the path and the resulting locally optimal solution x^* is returned in line 18.

Like back-and-forward path-relinking, the mixed variant thoroughly explores both neighborhoods $N_{x^2}(x^1)$ and $N_{x^1}(x^2)$. Furthermore, it is faster than back-and-forward path-relinking and usually takes as long as the backward or forward variants. Mixed path-relinking was suggested by Glover (1996) and was first implemented and tested in the context of the 2-path network design problem (Resende and Ribeiro, 2010a), for which it was shown to outperform forward, backward, and back and forward path-relinking. Figure 4 illustrates the comparison of a pure GRASP heuristic with four variants combined with path-relinking and applied to the 2-path network design problem: forward, backward, back and forward, and mixed. The run-time distribution plots show that GRASP with mixed path-relinking has the best running time profile among the variants compared.

One can expect to see most solutions produced by path-relinking to come from subpaths close to either the initiating or guiding solutions. Resende et al. (2010) showed that this occurs, for example, in instances of the max-min diversity problem. In that experiment, a back and forward path-relinking scheme was tested. It was experimentally shown that exploring the subpaths near the extremities may produce solutions about as good as those found by exploring the entire path, since there is a higher concentration of better solutions close to the initial solutions explored by path-relinking. It is straightforward to adapt path-relinking to explore only the neighborhoods close to the extremes. *Truncated* path-relinking can be applied to either forward, backward, backward and forward, or mixed path-relinking: instead of exploring the entire path, it just explores a fraction of the path and, consequently, takes a fraction of the time to run. Truncated path-relinking was also applied in Andrade and Resende (2007a).

3.2. **Minimum distance required for path-relinking.** We assume that we want to connect two locally optimal solutions x^1 and x^2 with path-relinking. If x^1 and x^2 differ by only one of their components, then the path directly connects the two solutions and no solution, other than x^1 and x^2, is visited.

Since x^1 and x^2 are both local minima, then $f(x^1) \leq f(r)$ for all $r \in N(x^1)$ and $f(x^2) \leq f(r)$ for all $r \in N(x^2)$. If x^1 and x^2 differ by exactly two components, then the Hamming distance between them is $|\Delta| = 2$ and any path between x^1 and x^2 visits exactly one intermediary solution $r \in N(x^1) \cap N(x^2)$. Consequently r cannot be better than either x^1 or x^2. Likewise, if $|\Delta| = 3$ then any path between x^1 and

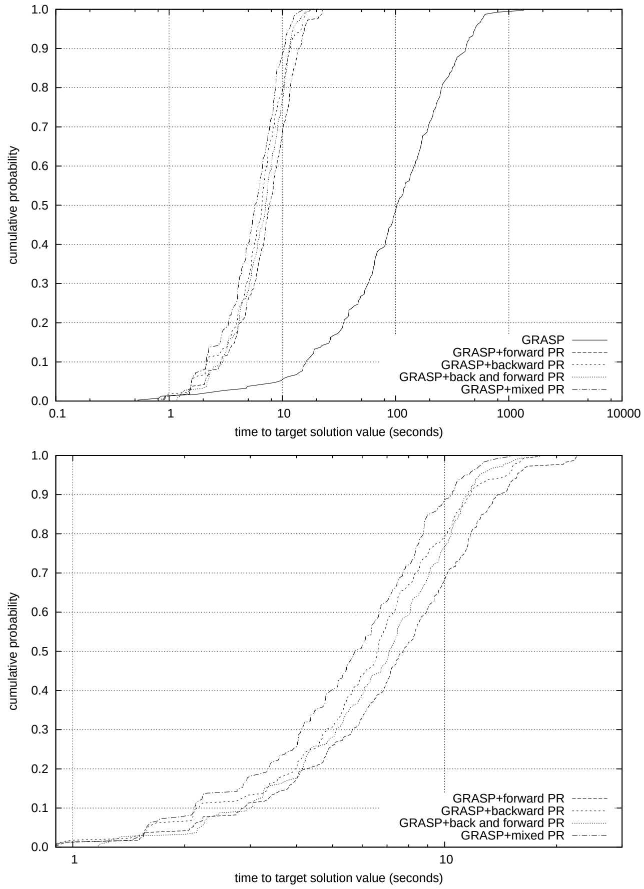


FIGURE 4. Time-to-target plots for pure GRASP and four variants of GRASP with path-relinking (forward, backward, back and forward, and mixed) on an instance of the 2-path network design problem.

x^2 visits two intermediary solutions $r^1 \in N(x^1)$ and $r^2 \in N(x^2)$ and, consequently, neither r^1 nor r^2 can be better than both x^1 and x^2.

Therefore, things only get interesting for $|\Delta| \geq 4$ and we may discard the application of path-relinking to pairs of solutions differing by less than four components.

3.3. **Randomization in path-relinking.** All previously described path-relinking strategies follow a greedy criterion to select the best move at each of their iterations. Therefore, path-relinking is limited to exploring a single path from a set of exponentially many paths between any pair of solutions. By adding randomization to path-relinking, *greedy randomized adaptive* path-relinking is not constrained to explore a single path. Instead of always selecting the move that results in the best solution, a restricted candidate list is constructed with the moves that result in promising solutions with costs in an interval that depends on the values of the best and worst moves, as well as on a parameter α. One move is selected at random from this set to produce the next solution in the path. By applying this strategy several times to the initial and guiding solutions, several paths can be explored. This strategy is useful when path-relinking is applied more than once to the same pair of solutions as it may occur in evolutionary path-relinking. Greedy randomized adaptive path-relinking has been applied in Andrade and Resende (2007a), Faria Jr. et al. (2005), and Resende et al. (2010).

4. Trajectory-based hybridization

Path-relinking is a major enhancement to trajectory-based stochastic local search algorithms that generate a sequence of locally optimal solutions. They include, but are not limited to, metaheuristics such as GRASP, VNS, tabu search, and simulated annealing. To hybridize path-relinking with them, one makes use of an *elite set* to collect a diverse pool of high-quality solutions found during the search. The elite set starts empty and is limited in size. Each new locally optimal solution produced by the heuristic is relinked with one or more solutions from the elite set.

Each solution resulting from path-relinking is considered as a candidate to be inserted in the elite set, where it can replace an elite solution of worse value. If the latter is not full, then the candidate is simply added to the elite set if it differs from all members. If the pool is full and the candidate is better than the best elite solution, then it replaces a solution in the pool. In case the candidate is better than the worst elite solution but not better than the best one, then it replaces a solution in the pool if it is sufficiently different from every other solution currently in the elite set. To balance the impact on pool quality and diversity, the solution selected to be replaced may be the one that is most similar to the entering solution among those elite solutions of quality no better than the entering solution (Resende and Werneck, 2004).

Given a local optimum x^1 produced by the stochastic local search heuristic, we need to select at random from the elite set a solution x^2 to be connected with x^1 by path-relinking. In principle, any solution in the elite set could be selected. However, one should avoid elite solutions that are too similar to x^1, because relinking two similar solutions limits the scope of the path-relinking search. Therefore, one should privilege pairs of solutions differing by a large number of components. A strategy introduced in (Resende and Werneck, 2004) is to select the elite solution x^2 at random with probability proportional to the number of components by which it differs from the local optimum x^1. This strategy also favors pool solutions that have a large number of paths connecting them with x^1.

After determining which solution (x^1 or x^2) will be designated the initial solution and which will be the guiding solution, any of the strategies previously described may be used. As noticed before, this choice involves trade-offs between computation time and solution quality.

The template of the algorithm in Figure 5 illustrates the pseudo-code of a hybrid trajectory-based heuristic that uses path-relinking for minimization. In line 2, the elite set P is initially empty. The loop in lines 3 to 12 makes up an iteration of the hybrid algorithm, until some stopping criterion is met. In line 4, x is a new locally optimal solution generated by the trajectory-based heuristic. If the elite set is empty, then x is inserted into the pool in line 5. Otherwise, x becomes the initial solution x^1 in line 7 and a guiding solution x^2 is selected at random from the pool in line 8. The initiating and guiding solutions are relinked in line 9 and the resulting solution is tested for inclusion into the elite set in line 10. The hybrid

1. **Algorithm** Trajectory-based Heuristic with Path-Relinking
2. $P \leftarrow \emptyset$;
3. **while** stopping criterion not satisfied **then do**;
4.   Build new local optimum x with trajectory heuristic;
5.   **if** $P = \emptyset$ **then** $P \leftarrow \{x\}$;
6.   **else do**;
7. $x^1 \leftarrow x$;
8.   Choose an elite set solution $x^2 \in P$ at random;
9. $x \leftarrow$ Path-Relinking(x^1, x^2);
10.   Update the elite set P with x;
11.   **end**;
12. **end**;
13. **return** elite set P and best solution $x^* \in P$;
14. **end**.

FIGURE 5. Template for the hybridization of path-relinking with a trajectory-based heuristic that generates locally optimal solutions.

procedure returns the set of elite solutions and the best solution found during the search in line 11.

4.1. **GRASP and VNS.** GRASP (Greedy Randomized Adaptive Search Procedure) is a multi-start metaheuristic for combinatorial optimization problems, in which each iteration consists basically of two phases: construction and local search (Resende and Ribeiro, 2002; 2010a;b). The construction phase builds a feasible solution following a greedy randomized criterion, whose neighborhood is investigated until a local minimum is found during the local search phase. Its basic implementation is memoryless, because it does not make use of information collected in previous iterations.

The use of path-relinking as an intensification strategy applied to each locally optimal solution obtained by a GRASP heuristic may lead to significant improvements in both solution times and running times. It was first proposed by Laguna and Martí (1999) and followed by several extensions, improvements, and successful applications. In this context, path-relinking is applied to each local minimum produced by the heuristic with a randomly selected elite solution. The resulting solution is a candidate for inclusion into the elite set, as in the template presented in Figure 5.

Enhancing GRASP with path-relinking almost always improves the performance of the heuristic. As an illustration, Figure 6 shows time-to-target plots (or run time distributions) for GRASP with and without path-relinking implementations corresponding to four different applications. These plots show the empirical cumulative probability distributions of the *time-to-target* random variable, i.e., the time needed to find a solution at least as good as a given target value. For all problems, the plots show that GRASP with path-relinking is able to find target solutions faster than the memoryless basic algorithm.

VNS (Variable Neighborhood Search) is based on the investigation of progressively more complex neighborhoods, in which each iteration consists basically of

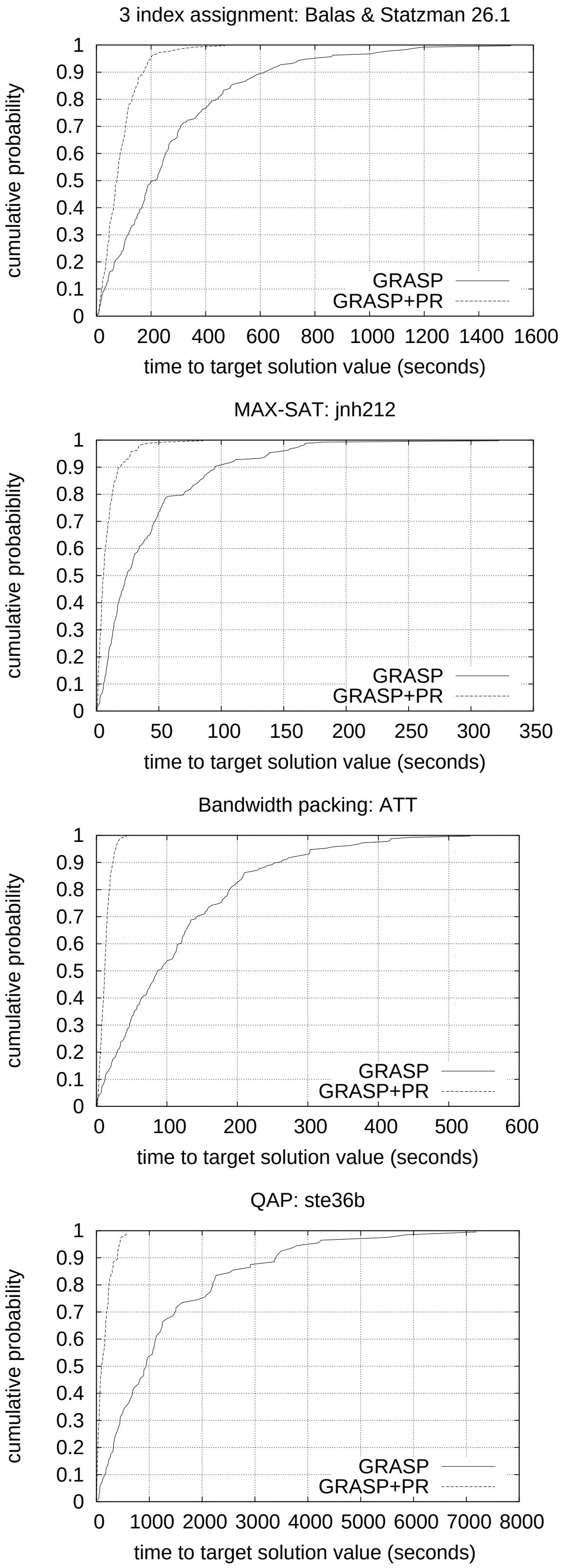


Figure 6. Time to target plots comparing running times of GRASP with and without path-relinking on distinct problems: three-index assignment (Aiex et al., 2005), maximum satisfiability (Festa et al., 2006), bandwidth packing, and quadratic assignment (Oliveira et al., 2004).

two phases: shaking (random selection of a solution in the current neighborhood of the current solution) and local search (Mladenović and Hansen, 1997; Hansen and Mladenović, 2002). Path-relinking may be embedded as an intensification strategy into a VNS heuristic following the same strategies above described, see e.g. Festa et al. (2002) for an application to the max-cut problem.

Path-relinking may also be used as a post-optimization procedure, applied to every pair of elite solutions produced by the GRASP or VNS heuristic. Aiex et al. (2005) applied path-relinking to all pairs of elite solutions as an intensification scheme to improve pool quality and as a post-optimization step. The post-optimization phase may also submit the elite set to an evolutionary process, as described next.

4.2. **Evolutionary path-relinking.** Path-relinking can also be applied to pairs of elite set solutions to search for new high-quality solutions and to improve the quality of the elite set. This can be done in a post-optimization phase, after the main heuristic stops, or periodically, when the main heuristic is still being applied (Aiex et al., 2005; Resende et al., 2010; Resende and Werneck, 2004).

Evolutionary path-relinking take as input the elite set and returns either the same elite set or a renewed one with an improved average cost.

Resende and Werneck (2004; 2006) described an evolutionary path-relinking scheme applied to pairs of elite solutions and used as a post-optimization step. It works with a population that evolves over a number of generations. The initial population is the input elite set. Population $k+1$ is initially empty. Path-relinking is applied to all pairs of solutions in population k. Each solution output from the path-relinking operation is a candidate for inclusion in population $k+1$. The usual rules for inclusion into the elite set are also adopted in evolutionary path-relinking. If population $k+1$ is not yet full, then the solution is accepted if it differs from all solutions in the population. After population $k+1$ is full, the solution is accepted if either it is better than the best solution in the population or it is better than the worst and is sufficiently different from all solutions in the population. Once a solution is accepted for inclusion into population $k+1$, it replaces the solution in population $k+1$ that does not have a better cost and that is most similar to it. The procedure halts when the best solution in population $k+1$ does not improve upon the best solution in population k. Andrade and Resende (2007b) used this evolutionary scheme as an intensification process every 100 GRASP iterations. During the intensification phase, every solution in the pool is relinked with the two best ones. Since two elite solutions might be relinked more than once in different calls to the intensification process, greedy randomized adaptive path-relinking was used.

A variation of the above scheme is described by Resende et al. (2010). In that scheme, while there exists a pair of solutions in the elite set for which path-relinking has not yet been applied, the two solutions are combined with path-relinking and the resulting solution is tested for membership in the elite set. If it is accepted, it then replaces the elite solution most similar to it among all solutions having worse cost. This variant outperformed several other heuristics using GRASP with path-relinking, simulated annealing, and tabu search for the max-min diversity problem.

Since some elite solutions may remain in the elite set over several applications of evolutionary path-relinking, greedy randomized adaptive path-relinking can be used in evolutionary path-relinking to avoid repeated explorations of the same paths in the solution space in different applications of the procedure.

5. Population-based hybridization

Path-relinking can also be applied to population-based stochastic local search as an advanced crossover or combination operator.

5.1. **Progressive crossover in genetic algorithms.** Path-relinking was first hybridized with a genetic algorithm by Ribeiro and Vianna (2003) to implement a progressive crossover operator in the context of its application to the phylogeny problem. The original proposal was extended and improved in Ribeiro and Vianna (2009), where a back and forward path-relinking strategy was used and the best solution along the two paths is returned as the offspring resulting from crossover. This mechanism is an extension of the traditional crossover operator: instead of producing only one offspring, defined by one single combination of two parents, it investigates many solutions that share characteristics of the selected parents. The solution found by path-relinking corresponds to the best offspring that can be obtained by applying the standard crossover to the parents.

The experiments reported in Ribeiro and Vianna (2009) made use of the results obtained for a randomly generated instance (TST17) of the phylogeny problem to assess the evolution of the solutions found by three different genetic algorithms in one hour (3,600 seconds) of computations: the random-keys genetic algorithm RKGA (Ribeiro and Vianna, 2003), the proposed genetic algorithm GA+PR using path-relinking to implement the progressive crossover operator, and the simpler genetic algorithm GAUni using uniform crossover. Figure 7 presents the solution value at the end of each generation for each of the 100 individuals in the population. Since the original random-keys genetic algorithm RKGA made use of elitism, the solution values are restricted to a smaller interval ranging between 2500 and 2620. The solution values obtained by the two other algorithms show more variability. The solutions found by algorithm GA+PR are better than those obtained by RKGA and GAUni, illustrating the contribution of the strategy based on path-relinking to implement the crossover operator.

Path-relinking was also applied by Zhang and Lai (2006) following the strategy proposed in Ribeiro and Vianna (2003) in the implementation of a genetic algorithm for the multiple-level warehouse layout problem. Their approach also makes use of path-relinking when the genetic algorithm seems to be trapped in a locally optimal solution. Once again, path-relinking was used by Vallada and Ruiz (2010) as a progressive crossover operator within a genetic algorithm for the minimum tardiness permutation flowshop problem. It was also applied as an intensification strategy after a number of generations without improvement to the best solution. The selected individuals are marked in order to not be selected again during the application of path-relinking. Path-relinking was also hybridized with a genetic algorithm as a post-optimization procedure (Ranjbar et al., 2008). In that paper, the solutions in the final population produced by the genetic algorithm are progressively combined and refined.

5.2. **Scatter Search.** Scatter search basically performs iterations over a set of good solutions identified as the reference set. In this context, path-relinking may be used as a solution combination method to transform a given subset of solutions produced by the subset generation method. We refer the reader to Laguna and Martí (2003) and to the special issue of EJOR (Martí, (Editor), 2006), where a number of successful applications of scatter search and path-relinking to problems

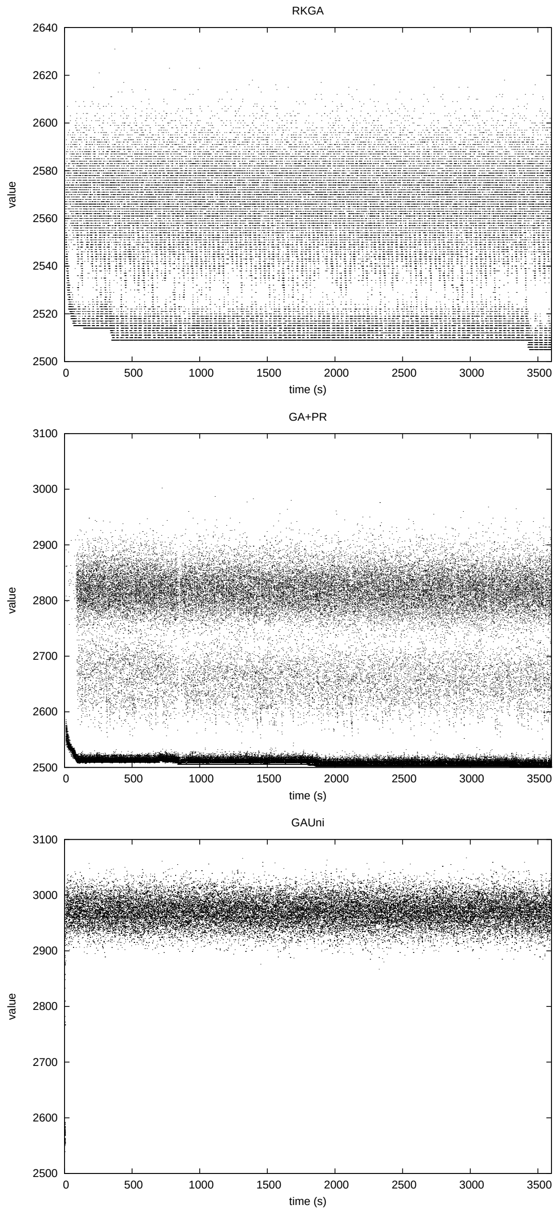


Figure 7. Solutions obtained by genetic algorithms for random instance TST17 for 3,600 seconds of computations.

in different domains can be found. These applications include neural networks, multi- and single-objective routing problems, graph drawing, scheduling, coloring, prediction, clustering, and nonlinear optimization,

6. Concluding remarks

Path-relinking is a very effective strategy to improve solution quality and to reduce computation times of stochastic local search algorithms for combinatorial optimization problems, leading to more robust implementations. Any available knowledge about the problem structure should be used in the development of efficient algorithms to explore the most attractive strategy for path-relinking.

Path-relinking may also be viewed as a constrained local search strategy applied to the initial solution, in which only a limited set of moves can be performed and non-improving moves are allowed. Several implementation strategies have been applied and combined in successful implementations of path-relinking in conjunction with different metaheuristics such as greedy randomized adaptive search procedures, variable neighborhood search, genetic algorithms, and scatter search. Some of these successful applications may be found in references such as (Aiex et al., 2003; 2005; Andrade and Resende, 2007a; Canuto et al., 2001; Festa et al., 2006; Oliveira et al., 2004; Reghioui et al., 2007; Resende and Ribeiro, 2003; Resende and Werneck, 2004; 2006; Ribeiro and Rosseti, 2007; Ribeiro et al., 2002; Scaparra and Church, 2005; Resende and Ribeiro, 2005; 2010a; Festa et al., 2002; Ribeiro and Vianna, 2009), among many others.

Parallel implementations of stochastic local search algorithms are quite robust and lead to linear speedups both in independent and cooperative strategies. Cooperative strategies are based on the collaboration between processors through path-relinking and a global pool of elite solutions. This allows the use of more processors to find better solutions in less computation time. Therefore, path-relinking offers a very practical strategy to improve the speedups of parallel implementations in clusters and grids (Resende and Ribeiro, 2010a; Ribeiro and Rosseti, 2007).

References

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

R.M. Aiex, S. Binato, and M.G.C. Resende. Parallel GRASP with path-relinking for job shop scheduling. *Parallel Computing*, 29:393–430, 2003.

R.M. Aiex, P.M. Pardalos, M.G.C. Resende, and G. Toraldo. GRASP with path-relinking for three-index assignment. *INFORMS J. on Computing*, 17:224–247, 2005.

R.M. Aiex, M.G.C. Resende, and C.C. Ribeiro. TTTPLOTS: A perl program to create time-to-target plots. *Optimization Letters*, 1:355–366, 2007.

D.V. Andrade and M.G.C. Resende. GRASP with path-relinking for network migration scheduling. In *Proceedings of the International Network Optimization Conference*, 2007a.

D.V. Andrade and M.G.C. Resende. GRASP with evolutionary path-relinking. Technical Report TD-6XPTS7, AT&T Labs Research, Florham Park, 2007b.

F. Barahona. Ground-state magnetization of Ising spin glasses. *Physical Review B*, 49:12864–12867, 1994.

F. Barahona, M. Grötschel, M. Jürgen, and G. Reinelt. An application of combinatorial optimization to statistical optimization and circuit layout design. *Operations Research*, 36:493–513, 1988.

S.A. Canuto, M.G.C. Resende, and C.C. Ribeiro. Local search with perturbations for the prize-collecting Steiner tree problem in graphs. *Networks*, 38:50–58, 2001.

K.C. Chang and D.-Z. Du. Efficient algorithms for layer assignment problems. *IEEE Transactions on Computer-Aided Design*, CAD-6:67–78, 1987.

R. Chen, Y. Kajitani, and S. Chan. A graph-theoretic via minimization algorithm for two-layer printed circuit boards. *IEEE Transactions on Circuits and Systems*, CAS-30:284–299, 1983.

H. Faria Jr., S. Binato, M.G.C. Resende, and D.J. Falcão. Transmission network design by a greedy randomized adaptive path relinking approach. *IEEE Transactions on Power Systems*, 20:43–49, 2005.

T.A. Feo, M.G.C. Resende, and S.H. Smith. A greedy randomized adaptive search procedure for maximum independent set. *Operations Research*, 42:860–878, 1994.

P. Festa and M.G.C. Resende. An annotated bibliography of GRASP, Part I: Algorithms. *International Transactions in Operational Research*, 16:1–24, 2009a.

P. Festa and M.G.C. Resende. An annotated bibliography of GRASP, Part II: Applications. *International Transactions in Operational Research*, 16:131–172, 2009b.

P. Festa, P.M. Pardalos, M.G.C. Resende, and C.C. Ribeiro. Randomized heutistics for the max-cut problem. *Optimization Methods and Sofware*, 6:1033–1058, 2002.

P. Festa, P.M. Pardalos, L.S. Pitsoulis, and M.G.C. Resende. GRASP with path-relinking for the weighted MAXSAT problem. *ACM Journal of Experimental Algorithmics*, 11:1–16, 2006.

F. Glover. Tabu search and adaptive memory programing – Advances, applications and challenges. In R.S. Barr, R.V. Helgason, and J.L. Kennington, editors, *Interfaces in Computer Science and Operations Research*, pages 1–75. Kluwer Academic Publishers, 1996.

F. Glover. Scatter search and path relinking. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 297–316. McGraw-Hill, 1999.

F. Glover. Multi-start and strategic oscillation methods – Principles to exploit adaptive memory. In M. Laguna and J.L. Gonzáles-Velarde, editors, *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, pages 1–24. Kluwer Academic Publishers, 2000.

F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.

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

F. Glover, M. Laguna, and R. Martí. Scatter search. In A. Ghosh and S. Tsutsui, editors, *Advances in Evolutionary Computation: Theory and Applications*, pages 519–537. Springer-Verlag, 2003.

F. Glover, M. Laguna, and R. Martí. Scatter search and path relinking: Foundations and advanced designs. In G.C. Onwubolu and B.V. Babu, editors, *New Optimization Techniques in Engineering*, volume 141 of *Studies in Fuzzyness and Soft Computing*, pages 87–100. Springer, 2004.

P. Hansen and N. Mladenović. Developments of variable neighborhood search. In C.C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, pages 415–439. Kluwer Academic Publishers, 2002.

J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

H.H. Hoos and T. Stützle. On the empirical evaluation of Las Vegas algorithms - Position paper. Technical report, Computer Science Department, University of British Columbia, 1998.

H.H. Hoos and T. Stützle. Evaluation of Las Vegas algorithms - Pitfalls and remedies. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 238–245, 1998.

R.M. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

M. Laguna and R. Martí. GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing*, 11:44–52, 1999.

M. Laguna and R. Martí. *Scatter search: Methodology and implementations in C.* Operations Research/Computer Science Interfaces Series. Kluwer Academic Publishers, Boston, 2003.

R. Martí, (Editor). Feature cluster on scatter search methods for optimization. *European Journal on Operational Research*, 169(2):351–698, 2006.

N. Mladenović and P. Hansen. Variable neighborhood search. *Computers and Operations Research*, 24:1097–1100, 1997.

C.A. Oliveira, P.M. Pardalos, and M.G.C. Resende. GRASP with path-relinking for the quadratic assignment problem. In C.C. Ribeiro and S.L. Martins, editors, *Proceedings of III Workshop on Efficient and Experimental Algorithms*, volume 3059, pages 356–368. Springer, 2004.

R.Y. Pinter. Optimal layer assignment for interconnect. *Journal of VLSI Computational Systems*, 1:123–137, 1984.

M. Ranjbar, F. Kianfar, and S. Shadrokh. Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm. *Applied Mathematics and Computation*, 196:879–888, 2008.

M. Reghioui, C. Prins, and N. Labadi. GRASP with path relinking for the capacitated arc routing problem with time windows. In M. Giacobini et al., editor, *Applications of Evolutinary Computing*, volume 4448 of *Lecture Notes in Computer Science*, pages 722–731. Springer, 2007.

M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219–242. Kluwer Academic Publishers, 2002.

M.G.C. Resende and C.C. Ribeiro. A GRASP with path-relinking for private virtual circuit routing. *Networks*, 41:104–114, 2003.

M.G.C. Resende and C.C. Ribeiro. GRASP with path-relinking: Recent advances and applications. In T. Ibaraki, K. Nonobe, and M. Yagiura, editors, *Metaheuristics: Progress as Real Problem Solvers*, pages 29–63. Springer, 2005.

M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In J.-Y. Potvin and M. Gendreau, editors, *Handbook of Metaheuristics*. Springer-Verlag, 2nd edition, 2010a.

M.G.C. Resende and C.C. Ribeiro. GRASP. In E.K. Burke and G. Kendall, editors, *Search Methodologies*. Springer, 2nd edition, 2010b.

M.G.C. Resende and R.F. Werneck. A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10:59–88, 2004.

M.G.C. Resende and R.F. Werneck. A hybrid multistart heuristic for the uncapacitated facility location problem. *European Journal of Operational Research*, 174: 54–68, 2006.

M.G.C. Resende, R. Martí, M. Gallego, and A. Duarte. GRASP and path relinking for the max-min diversity problem. *Computers and Operations Research*, 37: 498–508, 2010.

C.C. Ribeiro and I. Rosseti. Efficient parallel cooperative implementations of GRASP heuristics. *Parallel Computing*, 33:21–35, 2007.

C.C. Ribeiro and I. Rosseti. Exploiting run time distributions to compare sequential and parallel stochastic local search algorithms. In *Proceedings of the VIII Metaheuristics International Conference*, Hamburg, 2009.

C.C. Ribeiro and D.S. Vianna. A genetic algorithm for the phylogeny problem using an optimized crossover strategy based on path-relinking. In *Anais do II Workshop Brasileiro de Bioinformtica*, pages 97–102, Macaé, 2003.

C.C. Ribeiro and D.S. Vianna. A hybrid genetic algorithm for the phylogeny problem using path-relinking as a progressive crossover strategy. *International Transactions in Operational Research*, 16:641–657, 2009.

C.C. Ribeiro, E. Uchoa, and R.F. Werneck. A hybrid GRASP with perturbations for the Steiner problem in graphs. *INFORMS J. on Computing*, 14:228–246, 2002.

C.C. Ribeiro, I. Rosseti, and R. Vallejos. On the use of run time distributions to evaluate and compare stochastic local search algorithms. In T. Stützle, M. Biratari, and H.H. Hoos, editors, *Engineering Stochastic Local Search Algorithms*, volume 5752 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 2009.

R.Y. Rubinstein and D.P. Kroese. *The Cross Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning.* Springer-Verlag, 2004.

M. Scaparra and R. Church. A GRASP and path relinking heuristic for rural road network development. *J. of Heuristics*, 11:89–108, 2005.

E. Vallada and R. Ruiz. Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem. *Omega*, 38:57–67, 2010.

G.Q. Zhang and K.K. Lai. Combining path relinking and genetic algorithms for the multiple-level warehouse layout problem. *European Journal of Operational Research*, 169:413–425, 2006.

(Celso C. C. Ribeiro) Department of Computer Science, Universidade Federal Fluminense, Rua Passo da Pátria, 156, Niterói, RJ 24210-240 Brazil.

E-mail address: `celso@ic.uff.br`

(Mauricio G. C. Resende) Algorithms and Optimization Research Department, AT&T Labs Research, 180 Park Avenue, Room C241, Florham Park, NJ 07932 USA.

E-mail address: `mgcr@research.att.com`