Cite as: M.G.C. Resende and C.C. Ribeiro, "GRASP: Greedy randomized adaptive search procedures," in "Search Methodologies - Introductory tutorials in optimization and decision support systems," E.K. Burke and G. Kendall (Eds.), 2nd edition, pp. 287-312, 2014, Springer.

GRASP: GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURES

MAURICIO G. C. RESENDE AND CELSO C. RIBEIRO

Abstract. GRASP is a multi-start metaheuristic for combinatorial optimization problems, in which each iteration consists basically of two phases: construction and local search. The construction phase builds a feasible solution, whose neighborhood is investigated until a local minimum is found during the local search phase. The best overall solution is kept as the result. An intensification strategy based on path-relinking is frequently used to improve solution quality and to reduce computation times by exploring elite solutions previously found along the search. This chapter describes the basic components of GRASP, successful implementation strategies, and effective hybridizations with path-relinking and other metaheuristics. We also list some tricks to be used in the quest for good implementations. The bibliography is enriched by an account of relevant applications and by links to surveys, software, and additional sources of material.

1. Introduction

Metaheuristics are general high-level procedures that coordinate simple heuristics and rules to find good quality solutions to computationally difficult combinatorial optimization problems. Among them, we find simulated annealing (see Chapter 10), tabu search (see Chapter 9), genetic algorithms (see Chapter 4), scatter search (see Chapter 5), VNS (see Chapter 12), ant colonies (see Chapter 8), and others. The method described in this chapter represents another example of such a technique. Metaheuristics are based on distinct paradigms and offer different mechanisms to escape from locally optimal solutions. They are among the most effective solution strategies for solving combinatorial optimization problems in practice and they have been applied to a wide array of academic and real-world problems. The customization (or instantiation) of a metaheuristic to a given problem yields a *heuristic* for that problem.

In this chapter, we consider the combinatorial optimization problem of minimizing $f(S)$ over all solutions $S \in X$, which is defined by a finite set $E = \{e_1, \ldots, e_n\}$ (called the ground set), by a set of feasible solutions $X \subseteq 2^E$, and by an objective function $f : 2^E \to \mathbb{R}$. The ground set E, the objective function f, and the constraints defining the set of feasible solutions X are defined and specific for each problem. We seek an optimal solution $S^* \in X$ such that $f(S^*) \leq f(S)$, $\forall S \in X$.

GRASP, which stands for *Greedy Randomized Adaptive Search Procedures* [48, 49], is a multi-start, or iterative metaheuristic, in which each iteration consists of two phases: construction and local search. The construction phase builds a

Date: July 1, 2008; Revised December 2010.

Key words and phrases. GRASP, metaheuristics, stochastic seach procedure, local search, greedy algorithm.

AT&T Labs Research Technical Report.

solution. If this solution is not feasible, a repair procedure should be applied to attempt to achieve feasibility. If feasibility cannot be reached, it is discarded and a new solution is created. Once a feasible solution is obtained, its neighborhood is investigated until a local minimum is found during the local search phase. The best overall solution is kept as the result.

Principles and building blocks of GRASP, which are also common to other metaheuristics, are reviewed in Section 2. A template for the basic GRASP algorithm is described in Section 3. The GRASP with path-relinking heuristic is considered in Section 4, where different strategies for the efficient implementation of path-relinking are discussed. Hybridizations of GRASP with data mining and other metaheuristics are reviewed in Section 5. Recommendations and good problem-solving practices leading to more efficient implementations are presented in Section 6. Finally, the last section provides several sources of additional information, with references and links to literature surveys, annotated bibliographies and source codes, tools and software for algorithm evaluation and comparison, and accounts of applications and parallel implementations of GRASP.

2. Principles and building blocks

Several principles and building blocks appear as components common to GRASP and other metaheuristics. They are often blended using different strategies and additional features that distinguish one metaheuristic from another.

2.1. Greedy algorithms.

In a greedy algorithm, solutions are progressively built from scratch. At each iteration, a new element from the ground set E is incorporated into the partial solution under construction, until a complete feasible solution is obtained. The selection of the next element to be incorporated is determined by the evaluation of all candidate elements according to a *greedy evaluation function*. This greedy function usually represents the incremental increase in the cost function due to the incorporation of this element into the partial solution under construction. The greediness criterion establishes that an element with the smallest incremental increase is selected, with ties being arbitrarily broken. Figure 1 provides a template for a greedy algorithm for a minimization problem.

procedure `GreedyAlgorithm`
$S \leftarrow \emptyset$;
Initialize the candidate set: $C \leftarrow E$;
Evaluate the incremental cost $c(e)$ for all $e \in C$;
while $C \neq \emptyset$ **do**
    Select an element $s \in C$ with the smallest incremental cost $c(s)$;
    Incorporate s into the current solution: $S \leftarrow S \cup \{s\}$;
    Update the candidate set C;
    Reevaluate the incremental cost $c(e)$ for all $e \in C$;
end;
return S;
end.

Figure 1. Greedy algorithm for minimization.

The solutions obtained by greedy algorithms are not necessarily optimal. Greedy algorithms are often used to build initial solutions to be explored by local search or metaheuristics.

2.2. **Randomization and greedy randomized algorithms.** Randomization plays a very important role in algorithm design. Metaheuristics such as simulated annealing, GRASP, and genetic algorithms rely on randomization to sample the search space. Randomization can also be used to break ties, enabling different trajectories to be followed from the same initial solution in multistart methods, or sampling different parts of large neighborhoods. One particularly important use of randomization appears in the context of greedy algorithms.

Greedy randomized algorithms are based on the same principle guiding pure greedy algorithms. However, they make use of randomization to build different solutions at different runs. Figure 2 illustrates the pseudo-code of a greedy randomized algorithm for minimization. At each iteration, the set of candidate elements is formed by all elements that can be incorporated into the partial solution under construction without destroying feasibility. As before, the selection of the next element is determined by the evaluation of all candidate elements according to a greedy evaluation function. The evaluation of the elements by this function leads to the creation of a restricted candidate list (RCL) formed by the best elements, i.e. those whose incorporation into the current partial solution results in the smallest incremental costs. The element to be incorporated into the partial solution is randomly selected from those in the RCL. Once the selected element has been incorporated into the partial solution, the set of candidate elements is updated and the incremental costs are reevaluated.

procedure `GreedyRandomizedAlgorithm(Seed)`
$S \leftarrow \emptyset$;
Initialize the candidate set: $C \leftarrow E$;
Evaluate the incremental cost $c(e)$ for all $e \in C$;
while $C \neq \emptyset$ **do**
    Build a list with the candidate elements having the smallest incremental costs;
    Select an element s from the restricted candidate list at random;
    Incorporate s into the solution: $S \leftarrow S \cup \{s\}$;
    Update the candidate set C;
    Reevaluate the incremental cost $c(e)$ for all $e \in C$;
end;
return S;
end.

FIGURE 2. Greedy randomized algorithm for minimization.

Greedy randomized algorithms are used for a variety of purposes. For example, they are used in the construction phase of GRASP heuristics or to create initial solutions for population-based metaheuristics such as genetic algorithms or scatter search. Randomization is also a major component of metaheuristics, such as simulated annealing and VNS, in which a solution in the neighborhood of the current solution is randomly generated at each iteration.

2.3. **Neighborhoods.** A *neighborhood* of a solution S is a set $N(S) \subseteq X$. Each solution $S' \in N(S)$ is reached from S by an operation called a *move*. Normally, two neighbor solutions S and $S' \in N(S)$ differ by only a few elements. Neighborhoods may also eventually contain infeasible solutions not in X.

A solution S^* is a local optimum with respect to a given neighborhood N if $f(S^*) \leq f(S), \forall S \in N(S^*)$. Local search methods are based on the exploration of solution neighborhoods, searching for improving solutions until a local optimum is found.

The definition of a neighborhood is not unique. Some implementations of metaheuristics make use of multiple neighborhood structures. A metaheuristic may also modify the neighborhood, by excluding some of the possible moves and introducing others. Such modifications might also require changes in the nature of solution evaluation. The strategic oscillation approach [61, 62] illustrates this intimate relationship between changes in neighborhood and changes in evaluation.

2.4. **Local search.** Solutions generated by greedy algorithms are not necessarily optimal, even with respect to simple neighborhoods. A local search technique attempts to improve solutions in an iterative fashion, by successively replacing the current solution by a better solution in a neighborhood of the current solution. It terminates when no better solution is found in the neighborhood. The pseudo-code of a basic local search algorithm for a minimization problem is given in Figure 3. It starts from a solution S and makes use of a neighborhood structure N.

```
procedure LocalSearch(S)
    while S is not locally optimal do
        Find S' ∈ N(S) with f(S') < f(S);
        S ← S';
    end;
    return S;
end.
```

FIGURE 3. Local search algorithm for minimization.

The effectiveness of a local search procedure depends on several aspects, such as the neighborhood structure, the neighborhood search technique, the speed of evaluation of the cost function, and the starting solution. The neighborhood search may be implemented using either a best-improving or a first-improving strategy. In the case of a *best-improving* strategy, all neighbors are investigated and the current solution is replaced by the best neighbor. In the case of a *first-improving* strategy, the current solution moves to the first neighbor whose cost function value is smaller than that of the current solution.

2.5. **Restricted neighborhoods and candidate lists.** Glover and Laguna [63] point out that the use of strategies to restrict neighborhoods and to create candidate lists is essential to restrict the number of solutions examined in a given iteration in situations where the neighborhoods are very large or their elements are expensive to evaluate.

Their goal consists of attempting to isolate regions of the neighborhood containing desirable features and inserting them into a list of candidates for close examination. The efficiency of candidate list strategies can be enhanced by the use of memory structures for efficient updates of move evaluations from one iteration to another. The effectiveness of a candidate list strategy should be evaluated in terms of the quality of the best solution found in some specified amount of computation time. Strategies such as aspiration plus, elite candidate list, successive filtering, sequential fan candidate list, and bounded change candidate list are reviewed in [63].

Ribeiro and Souza [127] used a candidate list strategy, based on quickly computed estimates of move values, to significantly speedup the search for the best neighbor in their tabu search heuristic for the Steiner problem in graphs. Moves with bad estimates were discarded. Restricted neighborhoods based on filtering out unpromising solutions with high evaluations are discussed, e.g., in [84, 117].

2.6. **Intensification and diversification.** Two important components of metaheuristics are intensification and diversification. *Intensification* strategies encourage move combinations and solution features historically found to be good or to return to explore attractive regions of the solution space more thoroughly. The implementation of intensification strategies enforces the investigation of neighborhoods of elite solutions and makes use of explicit memory to do so. Intensification is often implemented in GRASP heuristics by using path-relinking, as described below.

Diversification strategies encourage the search to examine unvisited regions of the solution space or to generate solutions that significantly differ from those previously visited. Penalty and incentive functions are often used in this context. Diversification is often implemented by means of perturbations which destroy the structure of the current solution. In the context of GRASP, they are used, for example, within hybridizations with the iterated local search (ILS) metaheuristic, as described in Section 5.

2.7. **Path-relinking.** Path-relinking was originally proposed by Glover [61] as an intensification strategy exploring trajectories connecting elite solutions obtained by tabu search or scatter search [62, 64]. Starting from one or more elite solutions, paths in the solution space leading toward other elite solutions are generated and explored in the search for better solutions. To generate paths, moves are selected to introduce attributes in the current solution that are present in the elite guiding solution. Path-relinking may be viewed as a strategy that seeks to incorporate attributes of high quality solutions, by favoring these attributes in the selected moves.

The algorithm in Figure 4 illustrates the pseudo-code of the path-relinking procedure applied to a pair of solutions S_s (starting solution) and S_t (target solution). The procedure starts by computing the symmetric difference $\Delta(S_s, S_t)$ between the two solutions, i.e. the set of elements of the ground set E that appear in one of them but not in the other. The symmetric difference also defines the set of moves that have to be successively applied to S_s until S_t is reached. At each step, the procedure examines all moves $m \in \Delta(S, S_t)$ from the current solution S and selects the one which results in the least cost solution, i.e. the one which minimizes $f(S \oplus m)$, where $S \oplus m$ is the solution resulting from applying move m to solution S. The best move m^* is made, producing solution $S \oplus m^*$. The set of available moves is updated. If necessary, the best solution $\bar{S}$ is updated. The procedure terminates when S_t is reached, i.e. when $\Delta(S, S_t) = \emptyset$. A path of solutions is thus generated linking S_s to S_t and $\bar{S}$ is the best solution in this path. Since there is no guarantee

procedure `PathRelinking`(S_s, S_t)
Compute the symmetric difference $\Delta(S_s, S_t)$;
$\bar{f} \leftarrow \min\{f(S_s), f(S_t)\}$;
$\bar{S} \leftarrow \text{argmin}\{f(S_s), f(S_t)\}$;
$S \leftarrow S_s$;
while $\Delta(S, S_t) \neq \emptyset$ **do**
    $m^* \leftarrow \text{argmin}\{f(S \oplus m) : m \in \Delta(S, S_t)\}$;
    $\Delta(S \oplus m^*, S_t) \leftarrow \Delta(S, S_t) \setminus \{m^*\}$;
    $S \leftarrow S \oplus m^*$;
    if $f(S) < \bar{f}$ **then**
        $\bar{f} \leftarrow f(S)$;
        $\bar{S} \leftarrow S$;
    end_if;
end_while;
$\bar{S} \leftarrow$ `LocalSearch`($\bar{S}$);
return $\bar{S}$;
end.

FIGURE 4. Path-relinking procedure for minimization.

that $\bar{S}$ is a local minimum, local search can be applied to it and the resulting local minimum is returned by the algorithm.

Path-relinking may also be viewed as a constrained local search strategy applied to the initial solution S_s, in which only a limited set of moves can be performed and uphill moves are allowed. Several alternatives have been considered and combined in successful implementations of path-relinking in conjunction with GRASP and other metaheuristics. They are reviewed in Section 4.

3. A TEMPLATE FOR GRASP

Each iteration of the original GRASP metaheuristic proposed in [48] may be divided in two main phases: construction and local search (see also [49, 108, 116, 117, 118, 120] for other surveys on GRASP and its extensions). These steps are repeated many times, characterizing a multistart metaheuristic. The construction phase builds a solution. If this solution is not feasible, it is either discarded or a repair heuristic is applied to achieve feasibility (examples of repair procedures can be found in [41, 42, 89, 93]). Once a feasible solution is obtained, its neighborhood is investigated until a local minimum is found during the local search phase. The best solution found over all iterations is returned.

The pseudo-code in Figure 5 illustrates the main blocks of a GRASP procedure for minimization, in which `MaxIterations` iterations are performed and `Seed` is used as the initial seed for the pseudo-random number generator.

An especially appealing characteristic of GRASP is the ease with which it can be implemented. Few parameters need to be set and tuned, and therefore development can focus on implementing efficient data structures to assure quick iterations. Basic implementations of GRASP rely exclusively on two parameters: the number `MaxIterations` of iterations and the parameter used to limit the size of the

```
procedure GRASP(MaxIterations, Seed)
1.  Set f* ← ∞;
2.  for k = 1, ..., MaxIterations do
3.      S ← GreedyRandomizedAlgorithm(Seed);
4.      if S is not feasible then
5.          S ← RepairSolution(S);
6.      end;
7.      S ← LocalSearch(S);
8.      if f(S) < f* then
9.          S* ← S;
10.         f* ← f(S);
11.     end;
12. end;
13. return S*;
end.
```

FIGURE 5. Template of a GRASP heuristic for minimization.

restricted candidate list within the greedy randomized algorithm used by the construction phase. In spite of its simplicity and ease of implementation, GRASP is a very effective metaheuristic and produces the best known solutions for many problems, see [57, 58, 59] for extensive surveys of applications of GRASP.

For the construction of the RCL used in the first phase we consider, without loss of generality, a minimization problem such as the one formulated in Section 1. As before, we denote by $c(e)$ the incremental cost associated with the incorporation of element $e \in E$ into the solution under construction. At any GRASP iteration, let c^{min} and c^{max} be, respectively, the smallest and the largest incremental costs.

The restricted candidate list is made up of the elements $e \in E$ with the best (i.e., the smallest) incremental costs $c(e)$. This list can be limited either by the number of elements (cardinality-based) or by their quality (value-based). In the first case, it is made up of the p elements with the best incremental costs, where p is a parameter. In this chapter, the RCL is associated with a threshold parameter $\alpha \in [0, 1]$. The restricted candidate list is formed by all "feasible" elements $e \in E$ which can be inserted into the partial solution under construction without destroying feasibility and whose quality is superior to the threshold value, i.e., $c(e) \in [c^{min}, c^{min} + \alpha(c^{max} - c^{min})]$. The case $\alpha = 0$ corresponds to a pure greedy algorithm, while $\alpha = 1$ is equivalent to a random construction. The pseudo code in Figure 6 is a refinement of the greedy randomized construction algorithm, whose pseudo-code appears in Figure 2.

GRASP may be viewed as a repetitive sampling technique. Each iteration produces a sample solution from an unknown distribution, whose mean value and variance are functions of the restrictive nature of the RCL. The pseudo code in Figure 6 shows that the parameter α controls the amounts of greediness and randomness in the algorithm. Resende and Ribeiro [117, 120] have shown that what often leads to good solutions are relatively low average solution values (i.e., close to the value of the purely greedy solution obtained with $\alpha = 0$) in the presence of a

procedure `GreedyRandomizedConstruction`(α, `Seed`)
$S \leftarrow \emptyset$;
Initialize the candidate set: $C \leftarrow E$;
Evaluate the incremental cost $c(e)$ for all $e \in C$;
while $C \neq \emptyset$ **do**
  $c^{min} \leftarrow \min\{c(e) \mid e \in C\}$;
  $c^{max} \leftarrow \max\{c(e) \mid e \in C\}$;
  Build the restricted candidate list: $\text{RCL} \leftarrow \{e \in C \mid c(e) \leq c^{min} + \alpha(c^{max} - c^{min})\}$;
  Choose s at random from RCL;
  Incorporate s into solution: $S \leftarrow S \cup \{s\}$;
  Update the candidate set C;
  Reevaluate the incremental cost $c(e)$ for all $e \in C$;
end;
return S;
end.

FIGURE 6. Refined pseudo-code of the construction phase using parameter α for defining a quality threshold.

relatively large variance (i.e., solutions obtained with a larger degree of randomness as α increases), such as is often the case for $\alpha = 0.2$.

Prais and Ribeiro [103] showed that using a single fixed value for the value of the RCL parameter α often hinders finding a high-quality solution, which eventually could be found if another value was used. An alternative is to use a different value of α, chosen uniformly at random in the interval $[0, 1]$, at each GRASP iteration. Prais and Ribeiro [103] proposed another alternative, the *Reactive* GRASP extension of the basic procedure, in which the parameter α is self-tuned and its value is periodically modified according with the quality of the solutions previously obtained. Applications to other problems (see e.g. [58, 120]) have shown that Reactive GRASP outperforms the basic algorithm. These results motivated the study of the behavior of GRASP for different strategies for the variation of the value of the RCL parameter α. The experiments reported in [103] show that implementation strategies based on the variation of α are likely to be more affective than one using a single fixed value for this parameter.

Two other randomized greedy approaches, with smaller worst-case complexities than that depicted in the pseudo-code of Figure 6 were proposed in [121]. Instead of combining greediness and randomness at each step of the construction procedure, the *random plus greedy* scheme applies randomness during the first p construction steps to produce a random partial solution. Next, the algorithm completes the solution with one or more pure greedy construction steps. By changing the value of the parameter p, one can control the balance between greediness and randomness in the construction: larger values of p correspond to solutions that are more random, with smaller values corresponding to greedier solutions. The *sampled greedy* construction provides a different way to combine randomness and greediness. This procedure is also controlled by a parameter p. At each step of the construction process, the procedure builds a restricted candidate list by sampling $\min\{p, |C|\}$ elements of the candidate set C. Each of the sampled elements is evaluated by the greedy function and an element with the smallest greedy function value is added to the partial solution. These steps are repeated until there are no more candidate elements. As before, the balance between greediness and randomness can be controlled by changing the value of the parameter p, i.e. the number of candidate elements that are sampled. Small sample sizes lead to more random solutions, while large sample sizes lead to more greedy solutions.

4. GRASP with path-relinking

GRASP, as originally proposed, is a memoryless procedure in which each iteration does not make use of information gathered in previous iterations. Path-relinking is a major enhancement used for search intensification with GRASP. By adding memory structures to the basic procedure described above, path-relinking leads to significant improvements in solution time and quality.

The basic principles of path-relinking were described in Section 2.7. The use of path-relinking within a GRASP procedure was proposed in [78] and followed by extensions, improvements, and successful applications (see Section 7). Surveys of GRASP with path-relinking can be found in [116, 118, 120]. Different schemes have been proposed for the implementation of path-relinking. In essence, it has been applied as a post-optimization phase (between every pair of elite solutions in the pool of elite solutions) and as an intensification strategy (between every local optimum obtained after the local search phase and one or more elite solutions in the pool of elite solutions).

In this last context, path-relinking is applied to pairs of solutions, one of which is a locally optimal solution and the other is randomly chosen from a pool with a limited number `MaxElite` of elite solutions found along the search. A simple strategy is to assign equal probabilities of being selected to each elite solution. Another strategy assigns probabilities proportional to the cardinality of the symmetric difference between the elite solution and the locally optimal solution. This strategy favors elite solutions that result in longer paths. One of these solutions is called the *initial solution*, while the other is the *guiding solution*. One or more paths in the solution space graph connecting these solutions may be explored in the search for better solutions. The pool of elite solutions is originally empty. Since we wish to maintain a pool of good but diverse solutions, each locally optimal solution obtained by local search is considered as a candidate to be inserted into the pool if it is sufficiently different from every other solution currently in the pool. If the pool already has `MaxElite` solutions and the candidate is better than the worst of them, then a simple strategy is to have the candidate replace the worst elite solution. This strategy improves the quality of the elite set. Another strategy is to have the candidate replace an elite solution with worse objective function value that is most similar to it. This strategy improves the diversity of the elite set as well as its quality.

The pseudo-code in Figure 7 illustrates the main steps of a GRASP procedure using path-relinking to implement a memory-based intensification strategy.

Several alternatives for applying path-relinking to a pair of solutions S and S' have been considered and combined in the literature. These include forward, backward, back and forward, mixed, truncated, greedy randomized adaptive, and evolutionary path-relinking. All these alternatives involve trade-offs between computation time and solution quality.

```
procedure GRASPwithPathRelinking(MaxIterations, Seed)
    Set f* ← ∞;
    Set Pool ← ∅;
    for k = 1, ..., MaxIterations do
        S ← GreedyRandomizedAlgorithm(Seed);
        if S is infeasible then
            S ← RepairSolution(S);
        endif;
        S ← LocalSearch(S);
        if k > 1 then
            Randomly select a solution S' ∈ Pool;
            S ← PathRelinking(S', S);
        endif;
        if f(S) < f* then
            S* ← S;
            f* ← f(S);
        end_if;
        Update Pool with S if it satisfies the membership conditions;
    end_for;
    return S*;
end.
```

FIGURE 7. Template of a GRASP with path-relinking heuristic for minimization.

In *forward* path-relinking, the GRASP local optimum S is designated as the initial solution and the pool solution S' is made the guiding solution. The roles of S and S' are interchanged in *backward* path-relinking. This scheme was originally proposed in Aiex et al. [7], Ribeiro et al. [128], and Resende and Ribeiro [116]. The main advantage of this approach over forward path-relinking comes from the fact that, in general, there are more high-quality solutions near pool elements than near GRASP local optima. Backward path-relinking explores more thoroughly the neighborhood around the pool solution, whereas forward path-relinking explores more thoroughly the neighborhood around the GRASP local optimum. Experiments in [7, 116] have confirmed that backward path-relinking usually outperforms forward path-relinking. *Back and forward* path-relinking combines forward and backward path-relinking, exploring two different paths. It finds solutions at least as good as forward path-relinking or backward path-relinking, but at the expense of taking about twice as long to run. *Mixed* path-relinking shares the benefits of back and forward path-relinking, in about the same time as forward or backward path-relinking alone. This is achieved by interchanging the roles of the initial and guiding solutions at each step of the path-relinking procedure. Ribeiro and Rosseti [126] have shown experimentally that it outperforms forward, backward, and back and forward path-relinking (see also [120]).

Other strategies have been proposed more recently. *Truncated* path-relinking can be applied to either forward, backward, back and forward, or mixed path-relinking. Instead of exploring the entire path, it takes only a fraction of those steps and consequently takes a fraction of the time to run. Since high-quality solutions tend to be near the initial or guiding solutions, exploring part of the path near the extremities may produce solutions about as good as those found by exploring the entire path. Indeed, Resende et al. [111] showed experimentally that this is the case for instances of the max-min diversity problem. *Greedy randomized adaptive* path-relinking, introduced by Binato et al. [27], is a semi-greedy version of path-relinking. Instead of taking the best move in the symmetric difference still not performed, a restricted candidate list of good moves still not performed is set up and a randomly selected move from the RCL is applied. By applying this strategy several times between the initial and guiding solutions, several alternative paths can be explored. Resende and Werneck [121, 122] described an *evolutionary* path-relinking scheme applied to pairs of elite solutions and used as a post-optimization phase, in which the pool resulting from the GRASP with path-relinking iterations progressively evolves as a population. Similar schemes were also used in [7, 111].

5. Extensions

Hybridizations of GRASP with metaheuristics such as tabu search, simulated annealing, variable neighborhood search, iterated local search, and genetic algorithms have been reported in the literature.

Almost all the randomization effort in GRASP involves the construction phase, since the local search always stops at the first local optimum. VNS (Variable Neighborhood Search, see Chapter 12) relies almost entirely on the randomization of the local search to escape from local optima. Thus, GRASP and VNS may be considered as complementary and potentially capable of leading to effective hybrid methods. Festa et al. [56] studied different variants and combinations of GRASP and VNS for the MAX-CUT problem, finding and improving the best known solutions for some open instances in the literature. Other examples of hybrids of GRASP with VNS include [26, 32].

GRASP has also been used in conjunction with genetic algorithms. Basically, the greedy randomized strategy used in the construction phase of a GRASP is applied to generate the initial population for a genetic algorithm. We may cite, e.g., the genetic algorithm of Ahuja et al. [5] for the quadratic assignment problem, which makes use of the GRASP proposed by Li et al. [80] to create the initial population of solutions. A similar approach was used by Armony et al. [19], with the initial population made up of both randomly generated solutions and those built by a GRASP.

The hybridization of GRASP with tabu search was first studied by Laguna and González-Velarde [77]. Delmaire et al. [39] considered two approaches. In the first, GRASP is applied as a powerful diversification strategy in the context of a tabu search procedure. The second approach is an implementation of the Reactive GRASP algorithm, in which the local search phase is strengthened by tabu search. Two two-stage heuristics are proposed in [1] for solving the multi-floor facility layout problem. GRASP/TS applies a GRASP to find the initial layout and tabu search to refine it. Souza et al. [137] used a short-term tabu search procedure as a substitute for the standard local search in a GRASP heuristic for the capacitated minimum spanning tree problem.

Iterated local search (ILS) iteratively builds a sequence of solutions generated by the repeated application of local search and perturbation of the local optimum found by local search [25, 81, 83]. Ribeiro and Urrutia [129] presented a GRASP with ILS heuristic for the mirrored traveling tournament problem. In this case, the GRASP construction produces a solution which is passed on to the ILS procedure.

The hybridization of GRASP with data mining techniques was introduced in [130]. This scheme uses a data mining algorithm to search for solution patterns that occur in high-quality elite solutions produced by the basic GRASP algorithm. These mined patterns are used as initial building blocks that guide the construction of new solutions that are submitted to local search. A survey of applications of DM-GRASP can be found in [134].

6. Tricks of the trade

(1) An especially appealing characteristic of GRASP is the ease with which it can be implemented. Few parameters need to be set and tuned. Therefore, algorithm development and coding can focus on implementing efficient data structures to ensure quick GRASP iterations.

(2) Most metaheuristics benefit from good initial solutions. Clever low-complexity algorithms leading to good feasible solutions can often be devised by examination of the problem structure. Good initial solutions lead to better final solutions and significantly reduce the time taken by local search.

(3) Using a single, fixed value for the restricted candidate list parameter α very often hinders finding a high-quality solution, which eventually could be found if another value was used. The use of strategies such as Reactive GRASP which vary the value of α may lead to better and more diverse solutions. The reactive approach leads to improvements over the basic GRASP in terms of robustness and solution quality, due to greater diversification and less reliance on parameter tuning. In addition to the original applications reported in [103, 104], it has also been applied in [13, 28, 29, 31, 39, 135]. Another simple strategy is to uniformly select at random a value for α at each GRASP iteration from the interval $[0, 1]$.

(4) Local search procedures may be implemented using a best-improving or a first-improving strategy, as well as any combination of them. In the case of the best-improving strategy, all neighbors are investigated and the current solution is replaced by the best neighbor. In the case of a first-improving strategy, the current solution moves to the first neighbor whose cost function value is smaller than that of the current solution. Both strategies quite often lead to same quality solutions, but in smaller computation times when the first-improving strategy is used. Premature convergence to a non-global local minimum is more likely to occur with a best-improving strategy.

(5) The definition of a neighborhood is not unique. Some implementations of metaheuristics make use of multiple neighborhood structures to improve solution quality and to speed up the search. Variable neighborhood descent (VND) allows the systematic exploration of multiple neighborhoods [67]. It is based on the facts that a local minimum with respect to one neighborhood is not necessarily a local minimum with respect to another and that a global minimum is a local minimum with respect to all neighborhoods. Furthermore, VND is also based on the empirical observation that, for many problems, local minima with respect to one or more neighborhoods are relatively close to each other [68]. Since a global minimum is a local minimum with respect to all neighborhoods, it should be easier to find a

global minimum if more neighborhoods are explored. In the case of nested neighborhoods, the search is first confined to smaller neighborhoods. A larger neighborhood is explored only after a local minimum is found in the current, smaller neighborhood. Neighborhoods are not necessarily nested. Non-nested neighborhoods have been successfully used, e.g., by Aloise et al. [10]).

(6) Local search can be considerably accelerated with the use of appropriate data structures and efficient algorithms. All possible attempts should be made to improve the neighborhood search procedure. Algorithms should be coded to have minimum complexity. The use of circular lists to represent and search the neighborhood is very helpful. Candidate lists storing the move values may be easy to update or may be used as quick approximations to avoid their reevaluation at every iteration. We have seen several implementations in which the time taken by the first local search code dropped from several minutes to a few milliseconds in the final version.

(7) Path-relinking is a very effective strategy to improve solution quality and to reduce computation times, 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.

(8) Different metaheuristics make use of a number of common components, such as greedy constructions, local search, randomization, candidate lists, multiple neighborhoods, path-relinking, etc. Borrowing and incorporating principles from other metaheuristics lead to efficient hybridizations of GRASP, which often results in the best algorithm for some problem class.

(9) There is no universal, general purpose metaheuristic that gives the best results for every problem [141] (see Chapter 16). The structure of each problem should be explored to bring additional intelligence into the solution strategy. Knowledge, experience, and information available in the literature for similar problems are very helpful. However, one should not be obsessed with a fixed idea or bounded by strategies that worked for other problems but might not be appropriate for the one on hand. The best algorithm is always the one that most exploits the structure of your problem and gives the best results.

7. Sources of additional information

Surveys on GRASP [49, 117, 120], path-relinking [118], and its applications [57, 58, 59] can be found in the literature, to where the interested reader is referred for more details and references.

The web page at `http://www.research.att.com/~mgcr` contains an always-updated on-line version of the annotated bibliography on GRASP which appeared in [57, 58, 59]. Source codes for GRASP heuristics for several problems are also available at `http://www.research.att.com/~mgcr/src/index.html`. The Twitter page `http://twitter.com/graspheuristic` posts links to recently published papers on GRASP and its applications.

Time-to-target (TTT) 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. TTT plots were used by Feo, Resende, and Smith [50] and have been advocated also by Hoos and Stützle [72, 71] as a way to characterize the running times of stochastic algorithms for combinatorial optimization. Aiex et al. [8] advocate and largely explored the use of TTT plots to evaluate and compare different randomized algorithms running on the same problem. The use of TTT plots has been growing ever since and they have been extensively applied in computational studies of sequential and parallel implementations of randomized algorithms (see, e.g., [117, 120, 126]. The foundations of the construction of time-to-target plots, together with their interpretation and applications, were surveyed by Aiex et al. [9]. This reference also describes a Perl language program to create time-to-target plots for measured CPU times that can be downloaded from `http://www.research.att.com/~mgcr/tttplots`.

The first application of GRASP described in the literature concerned the set covering problem [48]. GRASP has been applied to many problems in different areas, such as routing [18, 21, 36, 75, 105]; logic [40, 54, 98, 109, 113, 114]; covering and partitioning [12, 17, 48, 65]; location [1, 33, 37, 66, 69, 39, 73, 96, 139, 140]; minimum Steiner tree [32, 84, 85, 88, 128]; optimization in graphs [2, 3, 4, 20, 50, 55, 56, 70, 76, 79, 82, 85, 100, 107, 110, 115, 124, 128, 137]; assignment [5, 7, 47, 60, 80, 90, 91, 92, 94, 97, 99, 102, 104, 112, 133]; timetabling, scheduling, and manufacturing [6, 13, 11, 15, 22, 23, 24, 28, 31, 35, 45, 46, 51, 52, 74, 77, 129, 131, 132, 142, 143]; transportation [18, 45, 47, 135, 136]; power systems [29, 30, 44]; telecommunications [2, 14, 15, 34, 73, 101, 104, 106, 107, 116, 125, 138]; graph and map drawing [37, 53, 78, 82, 95, 115, 124]; biology [16, 43]; and VLSI [17], among others.

GRASP is a metaheuristic very well suited for parallel implementation, due to the independence of its iterations. Parallel cooperative versions of GRASP with path-relinking may also be implemented in parallel if a centralized pool of elite solutions is kept by one of the processors. Surveys and accounts of parallel implementations of GRASP in networks of workstations, clusters, and grids may be found in [38, 86, 87, 119, 123, 125, 126].

References

[1] S. Abdinnour-Helm and S.W. Hadley. Tabu search based heuristics for multi-floor facility layout. *International J. of Production Research*, 38:365–383, 2000.

[2] J. Abello, P.M. Pardalos, and M.G.C. Resende. On maximum clique problems in very large graphs. In J. Abello and J. Vitter, editors, *External memory algorithms and visualization*, volume 50 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 199–130. American Mathematical Society, 1999.

[3] J. Abello, M.G.C. Resende, and S. Sudarsky. Massive quasi-clique detection. In S. Rajsbaum, editor, *LATIN 2002: Theoretical Informatics*, volume 2286 of *Lecture Notes in Computer Science*, pages 598–612. Springer, 2002.

[4] R.K. Ahuja, J.B. Orlin, and D. Sharma. Multi-exchange neighborhood structures for the capacitated minimum spanning tree problem. *Mathematical Programming*, 91:71–97, 2001.

[5] R.K. Ahuja, J.B. Orlin, and A. Tiwari. A greedy genetic algorithm for the quadratic assignment problem. *Computers and Operations Research*, 27:917–934, 2000.

[6] 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.

[7] 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.

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

[9] 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.

[10] D.J. Aloise, D. Aloise, C.T.M. Rocha, C.C. Ribeiro, J.C. Ribeiro Filho, and L.S.S. Moura. Scheduling workover rigs for onshore oil production. *Discrete Applied Mathematics*, 154:695–702, 2006.

[11] R. Álvarez-Valdés, E. Crespo, J.M. Tamarit, and F. Villa. GRASP and path relinking for project scheduling under partially renewable resources. *European J. of Operational Research*, 189:1153–1170, 2008.

[12] R. Álvarez-Valdés, F. Parreño, and J.M. Tamarit. A GRASP algorithm for constrained two-dimensional non-guillotine cutting problems. *J. of the Operational Research Society*, 56:414–425, 2005.

[13] R. Álvarez-Valdés, F. Parreño, and J.M. Tamarit. Reactive GRASP for the strip-packing problem. *Computers and Operations Research*, 35:1065–1083, 2008.

[14] E. Amaldi, A. Capone, and F. Malucelli. Planning UMTS base station location: Optimization models with power control and algorithms. *IEEE Transactions on Wireless Communications*, 2:939–952, 2003.

[15] D.V. Andrade and M.G.C. Resende. A GRASP for PBX telephone migration scheduling. In *Proceedings of The Eighth INFORMS Telecommunications Conference*, Dallas, 2006.

[16] A.A. Andreatta and C.C. Ribeiro. Heuristics for the phylogeny problem. *J. of Heuristics*, 8:429–447, 2002.

[17] S. Areibi and A. Vannelli. A GRASP clustering technique for circuit partitioning. In J. Gu and P.M. Pardalos, editors, *Satisfiability problems*, volume 35 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 711–724. American Mathematical Society, 1997.

[18] M.F. Argüello, J.F. Bard, and G. Yu. A GRASP for aircraft routing in response to groundings and delays. *J. of Combinatorial Optimization*, 1:211–228, 1997.

[19] M. Armony, J.C. Klincewicz, H. Luss, and M.B. Rosenwein. Design of stacked self-healing rings using a genetic algorithm. *J. of Heuristics*, 6:85–105, 2000.

[20] J.E.C. Arroyo, P.S. Vieira, and D.S. Vianna. A GRASP algorithm for the multi-criteria minimum spanning tree problem. *Annals of Operations Research*, 159:125–133, 2008.

[21] J.B. Atkinson. A greedy randomised search heuristic for time-constrained vehicle scheduling and the incorporation of a learning strategy. *J. of the Operational Research Society*, 49:700–708, 1998.

[22] J.F. Bard and T.A. Feo. Operations sequencing in discrete parts manufacturing. *Management Science*, 35:249–255, 1989.

[23] J.F. Bard and T.A. Feo. An algorithm for the manufacturing equipment selection problem. *IIE Transactions*, 23:83–92, 1991.

[24] J.F. Bard, T.A. Feo, and S. Holland. A GRASP for scheduling printed wiring board assembly. *IIE Transactions*, 28:155–165, 1996.

[25] J. Baxter. Local optima avoidance in depot location. *J. of the Operational Research Society*, 32:815–819, 1981.

[26] J.D. Beltrán, J.E. Calderón, R.J. Cabrera, J.A.M. Pérez, and J.M. Moreno-Vega. GRASP/VNS hybrid for the strip packing problem. In *Proceedings of Hybrid Metaheuristics*, pages 79–90, 2004.

[27] S. Binato, H. Faria Jr., and M.G.C. Resende. Greedy randomized adaptive path relinking. In J.P. Sousa, editor, *Proceedings of the IV Metaheuristics International Conference*, pages 393–397, Porto, 2001.

[28] S. Binato, W.J. Hery, D. Loewenstern, and M.G.C. Resende. A GRASP for job shop scheduling. In C.C. Ribeiro and P. Hansen, editors, *Essays and surveys in metaheuristics*, pages 59–79. Kluwer Academic Publishers, 2002.

[29] S. Binato and G.C. Oliveira. A reactive GRASP for transmission network expansion planning. In C.C. Ribeiro and P. Hansen, editors, *Essays and surveys in metaheuristics*, pages 81–100. Kluwer Academic Publishers, 2002.

[30] S. Binato, G.C. Oliveira, and J.L. Araújo. A greedy randomized adaptive search procedure for transmission expansion planning. *IEEE Transactions on Power Systems*, 16:247–253, 2001.

[31] M. Boudia, M.A.O. Louly, and C. Prins. A reactive GRASP and path relinking for a combined production-distribution problem. *Computers and Operations Research*, 34:3402–3419, 2007.

[32] 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.

[33] R. Colomé and D. Serra. Consumer choice in competitive location models: Formulations and heuristics. *Papers in Regional Science*, 80:439–464, 2001.

[34] C. Commander, C.A.S. Oliveira, P.M. Pardalos, and M.G.C. Resende. A GRASP heuristic for the cooperative communication problem in ad hoc networks. In *Proceedings of the VI Metaheuristics International Conference*, pages 225–330, Vienna, 2005.

[35] C.W. Commander, S.I. Butenko, P.M. Pardalos, and C.A.S. Oliveira. Reactive GRASP with path relinking for the broadcast scheduling problem. In *Proceedings of the 40th Annual International Telemetry Conference*, pages 792–800, 2004.

[36] A. Corberán, R. Martí, and J.M. Sanchís. A GRASP heuristic for the mixed Chinese postman problem. *European J. of Operational Research*, 142:70–80, 2002.

[37] G.L. Cravo, G.M. Ribeiro, and L.A. Nogueira Lorena. A greedy randomized adaptive search procedure for the point-feature cartographic label placement. *Computers and Geosciences*, 34:373–386, 2008.

[38] V.-D. Cung, S.L. Martins, C.C. Ribeiro, and C. Roucairol. Strategies for the parallel implementation of metaheuristics. In C.C. Ribeiro and C.C. Ribeiro, editors, *Essays and surveys in metaheuristics*, pages 263–308. Kluwer Academic Publishers, 2002.

[39] H. Delmaire, J.A. Díaz, E. Fernández, and M. Ortega. Reactive GRASP and tabu search based heuristics for the single source capacitated plant location problem. *INFOR*, 37:194–225, 1999.

[40] A.S. Deshpande and E. Triantaphyllou. A greedy randomized adaptive search procedure (GRASP) for inferring logical clauses from examples in polynomial time and some extensions. *Mathematical Computer Modelling*, 27:75–99, 1998.

[41] A.R. Duarte, C.C. Ribeiro, and S. Urrutia. A hybrid ILS heuristic to the referee assignment problem with an embedded MIP strategy. In *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 82–95. Springer, 2007.

[42] A.R. Duarte, C.C. Ribeiro, S. Urrutia, and E.H. Haeusler. Referee assignment in sports leagues. In *Practice and Theory of Automated Timetabling VI*, volume 3867 of *Lecture Notes in Computer Science*, pages 158–173. Springer, 2007.

[43] C.C. Ribeiro e D.S. Vianna. A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure. *International Transactions in Operational Research*, 12:325–338, 2005.

[44] 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.

[45] T.A. Feo and J.F. Bard. Flight scheduling and maintenance base planning. *Management Science*, 35:1415–1432, 1989.

[46] T.A. Feo, J.F. Bard, and S. Holland. Facility-wide planning and scheduling of printed wiring board assembly. *Operations Research*, 43:219–230, 1995.

[47] T.A. Feo and J.L. González-Velarde. The intermodal trailer assignment problem: Models, algorithms, and heuristics. *Transportation Science*, 29:330–341, 1995.

[48] T.A. Feo and M.G.C. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8:67–71, 1989.

[49] T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *J. of Global Optimization*, 6:109–133, 1995.

[50] 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.

[51] T.A. Feo, K. Sarathy, and J. McGahan. A GRASP for single machine scheduling with sequence dependent setup costs and linear delay penalties. *Computers and Operations Research*, 23:881–895, 1996.

[52] T.A. Feo, K. Venkatraman, and J.F. Bard. A GRASP for a difficult single machine scheduling problem. *Computers and Operations Research*, 18:635–643, 1991.

[53] E. Fernández and R. Martí. GRASP for seam drawing in mosaicking of aerial photographic maps. *J. of Heuristics*, 5:181–197, 1999.

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

[55] P. Festa, P.M. Pardalos, and M.G.C. Resende. Algorithm 815: FORTRAN subroutines for computing approximate solution to feedback set problems using GRASP. *ACM Transactions on Mathematical Software*, 27:456–464, 2001.

[56] P. Festa, P.M. Pardalos, M.G.C. Resende, and C.C. Ribeiro. Randomized heuristics for the MAX-CUT problem. *Optimization Methods and Software*, 7:1033–1058, 2002.

[57] P. Festa and M.G.C. Resende. GRASP: An annotated bibliography. In C.C. Ribeiro and P. Hansen, editors, *Essays and surveys in metaheuristics*, pages 325–367. Kluwer Academic Publishers, 2002.

[58] P. Festa and M.G.C. Resende. An annotated bibliography of GRASP, Part I: Algorithms. *International Transactions in Operational Research*, 16, 2009. To appear.

[59] P. Festa and M.G.C. Resende. An annotated bibliography of GRASP, Part II: Applications. *International Transactions in Operational Research*, 16, 2009. To appear.

[60] C. Fleurent and F. Glover. Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. *INFORMS J. on Computing*, 11:198–204, 1999.

[61] 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.

[62] 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.

[63] F. Glover and M. Laguna. *Tabu search*. Kluwer Academic Publishers, 1997.

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

[65] P.L. Hammer and D.J. Rader, Jr. Maximally disjoint solutions of the set covering problem. *J. of Heuristics*, 7:131–144, 2001.

[66] B.T. Han and V.T. Raja. A GRASP heuristic for solving an extended capacitated concentrator location problem. *International J. of Information Technology and Decision Making*, 2:597–617, 2003.

[67] P. Hansen and N. Mladenović. An introduction to variable neighbourhood search. In S Voss, S. Martello, I.H. Osman, and C. Roucairol, editors, *Metaheuristics: Advances and trends in local search procedures for optimization*, pages 433–458. Kluwer Academic Publishers, 1999.

[68] P. Hansen and N. Mladenović. Variable neighborhood search. In F. Glover and G. Kochenberger, editors, *Handbook of metaheuristics*, pages 145–184. Kluwer Academic Publishers, 2003.

[69] K. Holmqvist, A. Migdalas, and P.M. Pardalos. Greedy randomized adaptive search for a location problem with economies of scale. In I.M. Bomze et al., editor, *Developments in global optimization*, pages 301–313. Kluwer Academic Publishers, 1997.

[70] K. Holmqvist, A. Migdalas, and P.M. Pardalos. A GRASP algorithm for the single source uncapacitated minimum concave-cost network flow problem. In P.M. Pardalos and D.-Z. Du, editors, *Network design: Connectivity and facilities location*, volume 40 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 131–142. American Mathematical Society, 1998.

[71] H.H. Hoos and T. Stützle. Evaluating Las Vegas algorithms – Pitfalls and remedies. In *Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence*, pages 238–245, 1998.

[72] 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.

[73] J.G. Klincewicz. Avoiding local optima in the p-hub location problem using tabu search and GRASP. *Annals of Operations Research*, 40:283–302, 1992.

[74] J.G. Klincewicz and A. Rajan. Using GRASP to solve the component grouping problem. *Naval Research Logistics*, 41:893–912, 1994.

[75] G. Kontoravdis and J.F. Bard. A GRASP for the vehicle routing problem with time windows. *ORSA J. on Computing*, 7:10–23, 1995.

[76] M. Laguna, T.A. Feo, and H.C. Elrod. A greedy randomized adaptive search procedure for the two-partition problem. *Operations Research*, 42:677–687, 1994.

[77] M. Laguna and J.L. González-Velarde. A search heuristic for just-in-time scheduling in parallel machines. *J. of Intelligent Manufacturing*, 2:253–260, 1991.

[78] M. Laguna and R. Martí. GRASP and path relinking for 2-layer straight line crossing minimization. *INFORMS J. on Computing*, 11:44–52, 1999.
[79] M. Laguna and R. Martí. A GRASP for coloring sparse graphs. *Computational Optimization and Applications*, 19:165–178, 2001.
[80] Y. Li, P.M. Pardalos, and M.G.C. Resende. A greedy randomized adaptive search procedure for the quadratic assignment problem. In P.M. Pardalos and H. Wolkowicz, editors, *Quadratic assignment and related problems*, volume 16 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 237–261. American Mathematical Society, 1994.
[81] H.R. Lourenço, O.C. Martin, and T. Stützle. Iterated local search. In F. Glover and G. Kochenberger, editors, *Handbook of metaheuristics*, pages 321–353. Kluwer Academic Publishers, 2003.
[82] R. Martí. Arc crossing minimization in graphs with GRASP. *IIE Transactions*, 33:913–919, 2001.
[83] O. Martin, S.W. Otto, and E.W. Felten. Large-step Markov chains for the traveling salesman problem. *Complex Systems*, 5:299–326, 1991.
[84] S.L. Martins, P.M. Pardalos, M.G.C. Resende, and C.C. Ribeiro. Greedy randomized adaptive search procedures for the Steiner problem in graphs. In P.M. Pardalos, S. Rajasejaran, and J. Rolim, editors, *Randomization methods in algorithmic design*, volume 43 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 133–145. American Mathematical Society, 1999.
[85] S.L. Martins, M.G.C. Resende, C.C. Ribeiro, and P. Pardalos. A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy. *J. of Global Optimization*, 17:267–283, 2000.
[86] S.L. Martins, C.C. Ribeiro, and I. Rosseti. Applications and parallel implementations of metaheuristics in network design and routing. In *Applied Computing*, volume 3285 of *Lecture Notes in Computer Science*, pages 205–213. Springer, 2004.
[87] S.L. Martins, C.C. Ribeiro, and I. Rosseti. Applications of parallel metaheuristics to optimization problems in telecommunications and bioinformatics. In El.-G. Talbi, editor, *Parallel combinatorial optimization*, pages 301–325. Wiley, 2006.
[88] S.L. Martins, C.C. Ribeiro, and M.C. Souza. A parallel GRASP for the Steiner problem in graphs. In A. Ferreira and J. Rolim, editors, *Proceedings of IRREGULAR'98 – 5th International Symposium on Solving Irregularly Structured Problems in Parallel*, volume 1457 of *Lecture Notes in Computer Science*, pages 285–297. Springer, 1998.
[89] G.R. Mateus, M.G.C. Resende, and R.M.A. Silva. GRASP with path-relinking for the generalized quadratic assignment problem. *J. of Heuristics*, 2010. Published online 1 September 2010.
[90] T. Mavridou, P.M. Pardalos, L.S. Pitsoulis, and M.G.C. Resende. A GRASP for the biquadratic assignment problem. *European J. of Operational Research*, 105:613–621, 1998.
[91] R.A. Murphey, P.M. Pardalos, and L.S. Pitsoulis. A greedy randomized adaptive search procedure for the multitarget multisensor tracking problem. In P.M. Pardalos and D.-Z. Du, editors, *Network design: Connectivity and facilities location*, volume 40 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 277–301. American Mathematical Society, 1998.
[92] R.A. Murphey, P.M. Pardalos, and L.S. Pitsoulis. A parallel GRASP for the data association multidimensional assignment problem. In P.M. Pardalos, editor, *Parallel processing of discrete problems*, volume 106 of *The IMA Volumes in Mathematics and Its Applications*, pages 159–180. Springer, 1998.
[93] M.C.V. Nascimento, M.G.C. Resende, and F.M.B. Toledo. GRASP with path-relinking for the multi-plant capacitated plot sizing problem. *European J. of Operational Research*, 2008. To appear.
[94] 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.
[95] I.H. Osman, B. Al-Ayoubi, and M. Barake. A greedy random adaptive search procedure for the weighted maximal planar graph problem. *Computers and Industrial Engineering*, 45:635–651, 2003.

[96] J.A. Pacheco and S. Casado. Solving two location models with few facilities by using a hybrid heuristic: A real health resources case. *Computers and Operations Research*, 32:3075–3091, 2005.

[97] P.M. Pardalos, L.S. Pitsoulis, and M.G.C. Resende. A parallel GRASP implementation for the quadratic assignment problem. In A. Ferreira and J. Rolim, editors, *Parallel Algorithms for Irregularly Structured Problems – Irregular'94*, pages 115–133. Kluwer Academic Publishers, 1995.

[98] P.M. Pardalos, L.S. Pitsoulis, and M.G.C. Resende. A parallel GRASP for MAX-SAT problems. *Lecture Notes in Computer Science*, 1184:575–585, 1996.

[99] P.M. Pardalos, L.S. Pitsoulis, and M.G.C. Resende. Algorithm 769: Fortran subroutines for approximate solution of sparse quadratic assignment problems using GRASP. *ACM Transactions on Mathematical Software*, 23:196–208, 1997.

[100] P.M. Pardalos, T. Qian, and M.G.C. Resende. A greedy randomized adaptive search procedure for the feedback vertex set problem. *J. of Combinatorial Optimization*, 2:399–412, 1999.

[101] E. Piñana, I. Plana, V. Campos, and R. Martí. GRASP and path relinking for the matrix bandwidth minimization. *European J. of Operational Research*, 153:200–210, 2004.

[102] L.S. Pitsoulis, P.M. Pardalos, and D.W. Hearn. Approximate solutions to the turbine balancing problem. *European J. of Operational Research*, 130:147–155, 2001.

[103] M. Prais and C.C. Ribeiro. Parameter variation in GRASP procedures. *Investigación Operativa*, 9:1–20, 2000.

[104] M. Prais and C.C. Ribeiro. Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment. *INFORMS J. on Computing*, 12:164–176, 2000.

[105] 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.

[106] L.I.P. Resende and M.G.C. Resende. A GRASP for frame relay permanent virtual circuit routing. In C.C. Ribeiro and P. Hansen, editors, *Extended Abstracts of the III Metaheuristics International Conference*, pages 397–401, 1999.

[107] M.G.C. Resende. Computing approximate solutions of the maximum covering problem using GRASP. *J. of Heuristics*, 4:161–171, 1998.

[108] M.G.C. Resende. Metaheuristic hybridization with Greedy Randomized Adaptive Search Procedures. In Zhi-Long Chen and S. Raghavan, editors, *TutORials in Operations Research*, pages 295–319. INFORMS, 2008.

[109] M.G.C. Resende and T.A. Feo. A GRASP for satisfiability. In D.S. Johnson and M.A. Trick, editors, *Cliques, Coloring, and Satisfiability: The second DIMACS implementation challenge*, volume 26 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 499–520. American Mathematical Society, 1996.

[110] M.G.C. Resende, T.A. Feo, and S.H. Smith. Algorithm 787: Fortran subroutines for approximate solution of maximum independent set problems using GRASP. *ACM Trans. Math. Software*, 24:386–394, 1998.

[111] 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.

[112] M.G.C. Resende, P.M. Pardalos, and Y. Li. Algorithm 754: Fortran subroutines for approximate solution of dense quadratic assignment problems using GRASP. *ACM Transactions on Mathematical Software*, 22:104–118, 1996.

[113] M.G.C. Resende, L.S. Pitsoulis, and P.M. Pardalos. Approximate solution of weighted MAX-SAT problems using GRASP. In J. Gu and P.M. Pardalos, editors, *Satisfiability problems*, volume 35 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 393–405. American Mathematical Society, 1997.

[114] M.G.C. Resende, L.S. Pitsoulis, and P.M. Pardalos. Fortran subroutines for computing approximate solutions of MAX-SAT problems using GRASP. *Discrete Applied Mathematics*, 100:95–113, 2000.

[115] M.G.C. Resende and C.C. Ribeiro. A GRASP for graph planarization. *Networks*, 29:173–189, 1997.

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

[117] M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 219–249. Kluwer Academic Publishers, 2003.

[118] 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.

[119] M.G.C. Resende and C.C. Ribeiro. Parallel greedy randomized adaptive search procedures. In E. Alba, editor, *Parallel metaheuristics: A new class of algorithms*, pages 315–346. Wiley, 2005.

[120] M.G.C. Resende and C.C. Ribeiro. Greedy randomized adaptive search procedures: Advances and applications. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, pages 293–319. Springer, 2nd edition, 2010.

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

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

[123] C.C. Ribeiro, S.L. Martins, and I. Rosseti. Metaheuristics for optimization problems in computer communications. *Computer Communications*, 30:656–669, 2007.

[124] C.C. Ribeiro and M.G.C. Resende. Algorithm 797: Fortran subroutines for approximate solution of graph planarization problems using GRASP. *ACM Transactions on Mathematical Software*, 25:342–352, 1999.

[125] C.C. Ribeiro and I. Rosseti. A parallel GRASP heuristic for the 2-path network design problem. *Lecture Notes in Computer Science*, 2400:922–926, 2002.

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

[127] C.C. Ribeiro and M.C. Souza. Tabu search for the Steiner problem in graphs. *Networks*, 36:138–146, 2000.

[128] 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.

[129] C.C. Ribeiro and S. Urrutia. Heuristics for the mirrored traveling tournament problem. *European J. of Operational Research*, 179:775–787, 2007.

[130] M.H. Ribeiro, A. Plastino, and S.L. Martins. Hybridization of GRASP metaheuristic with data mining techniques. *J. of Mathematical Modelling and Algorithms*, 5:23–41, 2006.

[131] R.Z. Ríos-Mercado and J.F. Bard. Heuristics for the flow line problem with setup costs. *European J. of Operational Research*, pages 76–98, 1998.

[132] R.Z. Ríos-Mercado and J.F. Bard. An enhanced TSP-based heuristic for makespan minimization in a flow shop with setup costs. *J. of Heuristics*, 5:57–74, 1999.

[133] A.J. Robertson. A set of greedy randomized adaptive local search procedure (GRASP) implementations for the multidimensional assignment problem. *Computational Optimization and Applications*, 19:145–164, 2001.

[134] L.F. Santos, S.L. Martins, and A. Plastino. Applications of the DM-GRASP heuristic: A survey. *International Transactions on Operational Research*, 15:387–416, 2008.

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

[136] D. Sosnowska. Optimization of a simplified fleet assignment problem with metaheuristics: Simulated annealing and GRASP. In P.M. Pardalos, editor, *Approximation and complexity in numerical optimization*. Kluwer Academic Publishers, 2000.

[137] M.C. Souza, C. Duhamel, and C.C. Ribeiro. A GRASP heuristic for the capacitated minimum spanning tree problem using a memory-based local search strategy. In M.G.C. Resende and J. Souza, editors, *Metaheuristics: Computer decision-making*, pages 627–658. Kluwer Academic Publishers, 2004.

[138] A. Srinivasan, K.G. Ramakrishnan, K. Kumaram, M. Aravamudam, and S. Naqvi. Optimal design of signaling networks for Internet telephony. In *IEEE INFOCOM 2000*, volume 2, pages 707–716, Tel-Aviv, 2000.

[139] T.L. Urban. Solution procedures for the dynamic facility layout problem. *Annals of Operations Research*, 76:323–342, 1998.

[140] T.L. Urban, W.-C. Chiang, and R.A. Russel. The integrated machine allocation and layout problem. *International J. of Production Research*, 38:2913–2930, 2000.

[141] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67–82, 1997.

[142] J.Y. Xu and S.Y. Chiu. Effective heuristic procedure for a field technician scheduling problem. *J. of Heuristics*, 7:495–509, 2001.

[143] J. Yen, M. Carlsson, M. Chang, J.M. Garcia, and H. Nguyen. Constraint solving for inkjet print mask design. *J. of Imaging Science and Technology*, 44:391–397, 2000.

(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

(Celso 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