Chapter 11
GRASP: Greedy Randomized Adaptive Search Procedures

Mauricio G.C. Resende and Celso C. Ribeiro

11.1 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 Chap. 10), tabu search (see Chap. 9), genetic algorithms (Chap. 4), scatter search (Chap. 5), variable neighborhood search (Chap. 12), ant colonies (Chap. 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 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 $f : 2^E \rightarrow \mathbb{R}$. The ground set E, the objective function f, and the constraints defining the set of feasible solutions X are 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 (Feo and Resende 1989, 1995), is a multistart, or iterative, metaheuristic in which

M.G.C. Resende (✉)
Algorithms and Optimization Research Department, AT&T Labs Research,
Florham Park, NJ, USA
e-mail: mgcr@research.att.com

C.C. Ribeiro
Department of Computer Science, Universidade Federal Fluminense, Niterói, RJ, Brazil
e-mail: celso@ic.uff.br

E.K. Burke and G. Kendall (eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, DOI 10.1007/978-1-4614-6940-7_11,

each iteration consists of two phases: construction and local search. The construction phase builds a 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.

The principles and building blocks of GRASP, which are also common to other metaheuristics, are reviewed in Sect. 11.2. A template for the basic GRASP algorithm is described in Sect. 11.3. The GRASP with path-relinking heuristic is considered in Sect. 11.4, where different strategies for the efficient implementation of path-relinking are discussed. Hybridizations of GRASP with data mining and other metaheuristics are reviewed in Sect. 11.5. Recommendations and good problem-solving practices leading to more efficient implementations are presented in Sect. 11.6. Finally, we suggest 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.

11.2 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.

11.2.1 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 11.1 provides a template for a greedy algorithm for a minimization problem.

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.

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.

Fig. 11.1 Greedy algorithm for minimization

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.

Fig. 11.2 Greedy randomized algorithm for minimization

11.2.2 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 11.2 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 re-evaluated.

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.

11.2.3 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 in an iterative fashion by successively 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 (Glover 1996) illustrates this intimate relationship between changes in neighborhood and changes in evaluation.

11.2.4 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 Fig. 11.3. It starts from a solution S and makes use of a neighborhood structure N.

procedure `LocalSearch`(S)
1. **while** S is not locally optimal **do**
2. Find $S' \in N(S)$ with $f(S') < f(S)$;
3. $S \leftarrow S'$;
4. **end**;
5. **return** S;

end.

Fig. 11.3 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.

11.2.5 Restricted Neighborhoods and Candidate Lists

Glover and Laguna (1997) 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 in 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 Glover and Laguna (1997).

Ribeiro and Souza (2000) used a candidate list strategy, based on quickly computed estimates of move values, to significantly speed up 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, for example, in Martins et al. (1999) and Resende and Ribeiro (2003b).

11.2.6 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 Sect. 11.5.

11.2.7 Path-Relinking

Path-relinking was originally proposed by Glover (1996) as an intensification strategy exploring trajectories connecting elite solutions obtained by tabu search or scatter search (Glover et al. 2000). 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 Fig. 11.4 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 that $\bar{S}$ is a local minimum, local search can be applied to it and the resulting local minimum is returned by the algorithm.

```
procedure PathRelinking(S_s, S_t)
    Compute the symmetric difference Δ(S_s, S_t);
    f̄ ← min{f(S_s), f(S_t)};
    S̄ ← argmin{f(S_s), f(S_t)};
    S ← S_s;
    while Δ(S, S_t) ≠ ∅ do
        m* ← argmin{f(S ⊕ m) : m ∈ Δ(S, S_t)};
        Δ(S ⊕ m*, S_t) ← Δ(S, S_t) \ {m*};
        S ← S ⊕ m*;
        if f(S) < f̄ then
            f̄ ← f(S);
            S̄ ← S;
        end_if;
    end_while;
    S̄ ← LocalSearch(S̄);
    return S̄;
end.
```

Fig. 11.4 Path-relinking procedure for minimization

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 Sect. 11.4.

11.3 A Template for GRASP

Each iteration of the original GRASP metaheuristic proposed in Feo and Resende (1989) may be divided in two main phases: construction and local search (see also Feo and Resende 1995; Resende 2008; Resende and Ribeiro 2003a,b, 2005a, 2010 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 Duarte et al. 2007a,b; Mateus et al. 2011; Nascimento et al. 2010). 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 Fig. 11.5 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

```
procedure GRASP(MaxIterations,Seed)
    Set f* ← ∞;
    for k = 1,...,MaxIterations do
        S ← GreedyRandomizedAlgorithm(Seed);
        if S is not feasible then
            S ← RepairSolution(S);
        end;
        S ← LocalSearch(S);
        if f(S) < f* then
            S* ← S;
            f* ← f(S);
        end;
    end;
    return S*;
end.
```

Fig. 11.5 Template of a GRASP heuristic for minimization

MaxIterations of iterations and the parameter used to limit the size of the RCL 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 Festa and Resende (2002, 2009a,b) 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 Sect. 11.1. 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 RCL 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 RCL 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, that is $c(e) \in [c^{\min}, c^{\min} + \alpha(c^{\max} - c^{\min})]$. The case $\alpha = 0$ corresponds to a pure greedy algorithm, while $\alpha = 1$ is equivalent to a random construction. The pseudo-code in Fig. 11.6 is a refinement of the greedy randomized construction algorithm, whose pseudo-code appears in Fig. 11.2.

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 Fig. 11.6 shows that the parameter α controls the amounts of greediness and randomness in the algorithm. Resende and Ribeiro (2003b, 2010) have shown that what often leads to good solutions are relatively low average solution values (i.e. close to the value of

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: 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.

Fig. 11.6 Refined pseudo-code of the construction phase using parameter α for defining a quality threshold

the purely greedy solution obtained with $\alpha = 0$) in the presence of a 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 (2000a) 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 (2000a) 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. Festa and Resende 2009a; Resende and Ribeiro 2010) 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 Prais and Ribeiro (2000a) 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 Fig. 11.6, were proposed by Resende and Werneck (2004). 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 RCL 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.

11.4 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 Sect. 11.2.7. The use of path-relinking within a GRASP procedure was proposed in Laguna and Martí (1999) and followed by extensions, improvements, and successful applications (see Sect. 11.7.2). Surveys of GRASP with path-relinking can be found in Resende and Ribeiro (2003a, 2005a, 2010). 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.

```
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.
```

Fig. 11.7 Template of a GRASP with path-relinking heuristic for minimization

The pseudo-code in Fig. 11.7 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.

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. (2005), Ribeiro et al. (2002), and Resende and Ribeiro (2003a). 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 Aiex et al. (2005) and Resende and Ribeiro (2003a) 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 (2007) have shown experimentally that it outperforms forward, backward, and back and forward path-relinking (see also Resende and Ribeiro 2010).

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. (2010) showed experimentally that this is the case for instances of the max–min diversity problem. *Greedy randomized adaptive* path-relinking, introduced by Faria et al. (2005), is a semi-greedy version of path-relinking. Instead of taking the best move in the symmetric difference still not performed, a RCL 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 (2004, 2006) 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 by Aiex et al. (2005) and Resende et al. (2010).

11.5 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. Variable neighborhood search (VNS), see Chap. 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. (2002) 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 Beltrán et al. (2004) and Canuto et al. (2001).

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, for example, the genetic algorithm of Ahuja et al. (2000) for the quadratic assignment problem, which makes use of the GRASP proposed by Li et al. (1994) to create the initial population of solutions. A similar approach was used by Armony et al. (2000), 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 (1991). Delmaire et al. (1999) 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 Abdinnour-Helm and Hadley (2000) 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. (2004) 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 (Lourenço et al. 2003; Martin et al. 1991). Ribeiro and Urrutia (2007) 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 by Ribeiro et al. (2006). 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 Santos et al. (2008).

11.6 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 RCL 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 by Prais and Ribeiro (2000a,b), it has also

been applied by Álvarez-Valdés et al. (2008b), Binato et al. (2002), Binato and Oliveira (2002), Boudia et al. (2007), Delmaire et al. (1999) and Scaparra and Church (2005). 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 (Hansen and Mladenović 2003). 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 (Hansen and Mladenović 2003). 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 by, for example, Aloise et al. (2006).
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 re-evaluation 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 and path-relinking. Borrowing and incorporating principles from other metaheuristics leads 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 (Wolpert and Macready 1997)—see Chap. 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.

11.7 Some Promising Areas for Future Application

We conclude this chapter with two promising areas for future applications of GRASP.

11.7.1 Continuous GRASP

Hirsch et al. (2007b) (see also Hirsch 2006) proposed an adaptation of GRASP for derivative-free continuous global optimization. Continuous GRASP (or simply C-GRASP) was shown to perform well on a set of multimodal test functions, as well as on difficult real-world applications (Hirsch et al. 2007b). It was applied to the registration of sensors in a sensor network (Hirsch et al. 2006), to compute solutions for systems of nonlinear equations (Hirsch et al. 2009), to determine which drugs are responsible for adverse reactions in patients (Hirsch et al. 2007a), and for dynamic, decentralized path planning of unmanned aerial vehicles (Hirsch and Ortiz-Pena 2009; Hirsch et al. 2007c). Improvements to the original C-GRASP (Hirsch et al. 2007b) are presented in Hirsch et al. (2010). These improvements are aimed at making implementations of the algorithm more efficient and increasing robustness, while at the same time keeping the overall algorithm simple to implement.

The local improvement procedures in the derivative-free C-GRASP sample points around the solution produced by the global greedy randomized procedure. Since they only make function evaluations and do not use gradient information, they can be used for local optimization of any type of function, including ones that are not smooth. Birgin et al. (2010) adapt C-GRASP for global optimization of functions for which gradients can be computed. This is accomplished by using GENCAN (Birgin and Martínez 2002), an active-set method for bound-constrained local minimization.

11.7.2 Probabilistic-Based Stopping Rules

The absence of effective stopping criteria is one of the main drawbacks of most metaheuristics. Implementations of such algorithms usually stop after performing a given maximum number of iterations or a given maximum number of consecutive iterations without improvement in the best known solution value, or after the stabilization of a set of elite solutions found along the search. Ribeiro et al. (2011) proposed effective probabilistic stopping rules for randomized metaheuristics such as GRASP, VNS, simulated annealing and genetic algorithms, based on the estimation of the probability of finding better solutions than the incumbent. Such probabilities may be computed and used online to estimate the trade-off between solution improvement and the time needed to achieve it. The results described in Ribeiro et al. (2011) are being extended to encompass memory-based methods such as GRASP with path-relinking and tabu search.

Sources of Additional Information

Surveys on GRASP (Feo and Resende 1995; Resende and Ribeiro 2003b, 2010), path-relinking (Resende and Ribeiro 2005a; Ribeiro and Resende 2012) and its applications (Festa and Resende 2002, 2009a,b) can be found in the literature, to which the interested reader is referred for more details.

The web page www.research.att.com/~mgcr contains an always-updated version of the annotated bibliography on GRASP which appeared in Festa and Resende (2002, 2009a,b). Source codes for GRASP heuristics for several problems are also available at http://www.research.att.com/~mgcr/src/index.html. The Twitter web 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 et al. (1994) and have been advocated also by Hoos and Stützle (1998) as a way to characterize the running times of stochastic algorithms for combinatorial optimization. Aiex et al. (2002) 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. Resende and Ribeiro 2003b, 2010; Ribeiro and Rosseti 2007). The foundations of the construction of TTT plots, together with their interpretation and applications, were surveyed by Aiex et al. (2007). This reference also describes a Perl language program to create TTT 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 (Feo and Resende 1989). GRASP has been applied to many problems in different areas, such as routing (Argüello et al. 1997; Corberán et al. 2002; Kontoravdis and Bard 1995; Reghioui et al. 2007), logic (Deshpande and Triantaphyllou 1998; Festa et al. 2006; Pardalos et al. 1996; Resende and Feo 1996; Resende et al. 1997, 2000), covering and partitioning (Álvarez-Valdés et al. 2005; Areibi and Vannelli 1997; Feo and Resende 1989; Hammer and Rader Jr 2001), location (Abdinnour-Helm and Hadley 2000; Colomé and Serra 2001; Cravo et al. 2008; Han and Raja 2003; Holmqvist et al. 1997; Delmaire et al. 1999; Urban 1998; Klincewicz 1992), minimum Steiner tree (Canuto et al. 2001; Martins et al. 1999, 2000, 1998; Ribeiro et al. 2002), optimization in graphs (Abello et al. 1999, 2002; Ahuja et al. 2001; Arroyo et al. 2008; Feo et al. 1994; Festa et al. 2001, 2002; Holmqvist et al. 1998; Laguna et al. 1994; Laguna and Martí 2001; Martí 2001; Martins et al. 2000; Pardalos et al. 1999; Resende 1998; Resende et al. 1998; Resende and Ribeiro 1997; Ribeiro and Resende 1999; Ribeiro et al. 2002; Souza et al. 2004), assignment (Ahuja et al. 2000; Aiex et al. 2005; Feo and González-Velarde 1995; Fleurent and Glover 1999; Li et al. 1994; Mavridou et al. 1998; Murphey et al. 1998a,b; Oliveira et al. 2004; Pardalos et al. 1995, 1997; Pitsoulis et al. 2001; Prais and Ribeiro 2000b; Resende et al. 1996; Robertson 2001), timetabling, scheduling, and manufacturing (Aiex et al. 2003; Álvarez-Valdés et al. 2008b,a; Andrade and Resende 2006; Bard and Feo 1989, 1991; Bard et al. 1996; Binato et al. 2002; Boudia et al. 2007; Commander et al. 2004; Feo and Bard 1989; Feo et al. 1995, 1996, 1991; Klincewicz and Rajan 1994; Laguna and González-Velarde 1991; Ribeiro and Urrutia 2007; Ríos-Mercado and Bard 1998, 1999; Xu and Chiu 2001; Yen et al. 2000), transportation (Argüello et al. 1997; Feo and Bard 1989; Feo and González-Velarde 1995; Scaparra and Church 2005), power systems (Binato and Oliveira 2002; Binato et al. 2001; Faria et al. 2005), telecommunications (Abello et al. 1999; Amaldi et al. 2003; Andrade and Resende 2006; Klincewicz 1992; Piñana et al. 2004; Prais and Ribeiro 2000b; Resende and Resende 1999; Resende 1998; Resende and Ribeiro 2003a; Ribeiro and Rosseti 2002; Srinivasan et al. 2000), graph and map drawing (Cravo et al. 2008; Fernández and Martí 1999; Laguna and Martí 1999; Martí 2001; Osman et al. 2003; Resende and Ribeiro 1997; Ribeiro and Resende 1999), biology (Andreatta and Ribeiro 2002; Ribeiro and Vianna 2005) and VLSI (Areibi and Vannelli 1997), 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 Cung et al. (2002), Martins et al. (2004, 2006), Resende and Ribeiro (2005b), Ribeiro et al. (2007) and Ribeiro and Rosseti (2002, 2007).

References

Abdinnour-Helm S, Hadley SW (2000) Tabu search based heuristics for multi-floor facility layout. Int J Prod Res 38:365–383

Abello J, Pardalos PM, Resende MGC (1999) On maximum clique problems in very large graphs. In: Abello J, Vitter J (eds) External memory algorithms and visualization, DIMACS 50. AMS, Providence, pp 199–130

Abello J, Resende MGC, Sudarsky S (2002) Massive quasi-clique detection. In: Rajsbaum S (ed) LATIN 2002: theoretical informatics. LNCS 2286. Springer, Berlin, pp 598–612

Ahuja RK, Orlin JB, Tiwari A (2000) A greedy genetic algorithm for the quadratic assignment problem. Comput Oper Res 27:917–934

Ahuja RK, Orlin JB, Sharma D (2001) Multi-exchange neighborhood structures for the capacitated minimum spanning tree problem. Math Program 91:71–97

Aiex RM, Resende MGC, Ribeiro CC (2002) Probability distribution of solution time in GRASP: an experimental investigation. J Heuristics 8:343–373

Aiex RM, Binato S, Resende MGC (2003) Parallel GRASP with path-relinking for job shop scheduling. Parallel Comput 29:393–430

Aiex RM, Pardalos PM, Resende MGC, Toraldo G (2005) GRASP with path-relinking for three-index assignment. INFORMS J Comput 17:224–247

Aiex RM, Resende MGC, Ribeiro CC (2007) TTTPLOTS: a Perl program to create time-to-target plots. Optim Lett 1:355–366

Aloise DJ, Aloise D, Rocha CTM, Ribeiro CC, Ribeiro Filho JC, Moura LSS (2006) Scheduling workover rigs for onshore oil production. Discret Appl Math 154:695–702

Álvarez-Valdés R, Parreño F, Tamarit JM (2005) A GRASP algorithm for constrained two-dimensional non-guillotine cutting problems. J Oper Res Soc 56:414–425

Álvarez-Valdés R, Crespo E, Tamarit JM, Villa F (2008a) GRASP and path relinking for project scheduling under partially renewable resources. Eur J Oper Res 189:1153–1170

Álvarez-Valdés R, Parreño F, Tamarit JM (2008b) Reactive GRASP for the strip-packing problem. Comput Oper Res 35:1065–1083

Amaldi E, Capone A, Malucelli F (2003) Planning UMTS base station location: optimization models with power control and algorithms. IEEE Trans Wirel Commun 2:939–952

Andrade DV, Resende MGC (2006) A GRASP for PBX telephone migration scheduling. In: Proc 8th INFORMS telecoms, Dallas, TX

Andreatta AA, Ribeiro CC (2002) Heuristics for the phylogeny problem. J Heuristics 8:429–447

Areibi S, Vannelli A (1997) A GRASP clustering technique for circuit partitioning. In: Gu J, Pardalos PM (eds) Satisfiability problems, DIMACS 35. AMS, Providence, pp 711–724

Argüello MF, Bard JF, Yu G (1997) A GRASP for aircraft routing in response to groundings and delays. J Comb Optim 1:211–228

Armony M, Klincewicz JC, Luss H, Rosenwein MB (2000) Design of stacked self-healing rings using a genetic algorithm. J Heuristics 6:85–105

Arroyo JEC, Vieira PS, Vianna DS (2008) A GRASP algorithm for the multi-criteria minimum spanning tree problem. Ann Oper Res 159:125–133

Bard JF, Feo TA (1989) Operations sequencing in discrete parts manufacturing. Manage Sci 35:249–255

Bard JF, Feo TA (1991) An algorithm for the manufacturing equipment selection problem. IIE Trans 23:83–92

Bard JF, Feo TA, Holland S (1996) A GRASP for scheduling printed wiring board assembly. IIE Trans 28:155–165

Beltrán JD, Calderón JE, Cabrera RJ, Pérez JAM, Moreno-Vega JM (2004) GRASP/VNS hybrid for the strip packing problem. In: Proc hybrid metaheuristics, Valencia, Spain, pp 79–90

Binato S, Oliveira GC (2002) A reactive GRASP for transmission network expansion planning. In: Ribeiro CC, Hansen P (eds) Essays and surveys in metaheuristics. Kluwer, Dordrecht, pp 81–100

Binato S, Oliveira GC, Araújo JL (2001) A greedy randomized adaptive search procedure for transmission expansion planning. IEEE Trans Power Syst 16:247–253

Binato S, Hery WJ, Loewenstern D, Resende MGC (2002) A GRASP for job shop scheduling. In: Ribeiro CC, Hansen P (eds) Essays and surveys in metaheuristics. Kluwer, Dordrecht, pp 59–79

Birgin EG, Martínez JM (2002) Large-scale active-set box-constrained optimization method with spectral projected gradients. Comput Optim Appl 23:101–125

Birgin EG, Gozzi EM, Resende MGC, Silva RMA (2010) Continuous GRASP with a local active-set method for bound-constrained global optimization. J Glob Optim 48:289–310

Boudia M, Louly MAO, Prins C (2007) A reactive GRASP and path relinking for a combined production-distribution problem. Comput Oper Res 34:3402–3419

Canuto SA, Resende MGC, Ribeiro CC (2001) Local search with perturbations for the prize-collecting Steiner tree problem in graphs. Networks 38:50–58

Colomé R, Serra D (2001) Consumer choice in competitive location models: formulations and heuristics. Pap Reg Sci 80:439–464

Commander CW, Butenko SI, Pardalos PM, Oliveira CAS (2004) Reactive GRASP with path relinking for the broadcast scheduling problem. In: Proceedings of the 40th Annual International telemetry conference, San Diego, CA, pp 792–800

Corberán A, Martí R, Sanchís JM (2002) A GRASP heuristic for the mixed Chinese postman problem. Eur J Oper Res 142:70–80

Cravo GL, Ribeiro GM, Nogueira Lorena LA (2008) A greedy randomized adaptive search procedure for the point-feature cartographic label placement. Comput Geosci 34:373–386

Cung V-D, Martins SL, Ribeiro CC, Roucairol C (2002) Strategies for the parallel implementation of metaheuristics. In: Ribeiro CC, Ribeiro CC (eds) Essays and surveys in metaheuristics. Kluwer, Dordrecht, pp 263–308

Delmaire H, Díaz JA, Fernández E, Ortega M (1999) Reactive GRASP and tabu search based heuristics for the single source capacitated plant location problem. INFOR 37:194–225

Deshpande AS, Triantaphyllou E (1998) A greedy randomized adaptive search procedure (GRASP) for inferring logical clauses from examples in polynomial time and some extensions. Math Comput Model 27:75–99

Duarte AR, Ribeiro CC, Urrutia S (2007a) A hybrid ILS heuristic to the referee assignment problem with an embedded MIP strategy. In: Hybrid metaheuristics. LNCS 4771. Springer, Berlin, pp 82–95

Duarte AR, Ribeiro CC, Urrutia S, Haeusler EH (2007b) Referee assignment in sports leagues. In: PATAT VI. LNCS 3867. Springer, Berlin, pp 158–173

Faria H Jr, Binato S, Resende MGC, Falcão DJ (2005) Transmission network design by a greedy randomized adaptive path relinking approach. IEEE Trans Power Syst 20:43–49

Feo TA, Bard JF (1989) Flight scheduling and maintenance base planning. Manage Sci 35:1415–1432

Feo TA, González-Velarde JL (1995) The intermodal trailer assignment problem: models, algorithms, and heuristics. Transp Sci 29:330–341

Feo TA, Resende MGC (1989) A probabilistic heuristic for a computationally difficult set covering problem. Oper Res Lett 8:67–71

Feo TA, Resende MGC (1995) Greedy randomized adaptive search procedures. J Glob Optim 6:109–133

Feo TA, Venkatraman K, Bard JF (1991) A GRASP for a difficult single machine scheduling problem. Comput Oper Res 18:635–643

Feo TA, Resende MGC, Smith SH (1994) A greedy randomized adaptive search procedure for maximum independent set. Oper Res 42:860–878

Feo TA, Bard JF, Holland S (1995) Facility-wide planning and scheduling of printed wiring board assembly. Oper Res 43:219–230

Feo TA, Sarathy K, McGahan J (1996) A GRASP for single machine scheduling with sequence dependent setup costs and linear delay penalties. Comput Oper Res 23:881–895

Fernández E, Martí R (1999) GRASP for seam drawing in mosaicking of aerial photographic maps. J Heuristics 5:181–197

Festa P, Resende MGC (2002) GRASP: an annotated bibliography. In: Ribeiro CC, Hansen P (eds) Essays and surveys in metaheuristics. Kluwer, Dordrecht, pp 325–367

Festa P, Resende MGC (2009a) An annotated bibliography of GRASP, part I: algorithms. Int Trans Oper Res 16:1–24

Festa P, Resende MGC (2009b) An annotated bibliography of GRASP, part II: applications. Int Trans Oper Res 16:131–172

Festa P, Pardalos PM, Resende MGC (2001) Algorithm 815: FORTRAN subroutines for computing approximate solution to feedback set problems using GRASP. ACM Trans Math Softw 27:456–464

Festa P, Pardalos PM, Resende MGC, Ribeiro CC (2002) Randomized heuristics for the MAX-CUT problem. Optim Methods Softw 7:1033–1058

Festa P, Pardalos PM, Pitsoulis LS, Resende MGC (2006) GRASP with path-relinking for the weighted MAXSAT problem. ACM J Exp Algorithm 11:1–16
Fleurent C, Glover F (1999) Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. INFORMS J Comput 11:198–204
Glover F (1996) Tabu search and adaptive memory programing—advances, applications and challenges. In: Barr RS, Helgasonm RV, Kennington JL (eds) Interfaces in computer science and operations research. Kluwer, Dordrecht, pp 1–75
Glover F, Laguna M (1997) Tabu search. Kluwer, Dordrecht
Glover F, Laguna M, Martí R (2000) Fundamentals of scatter search and path relinking. Control Cybern 39:653–684
Hammer PL, Rader DJ Jr (2001) Maximally disjoint solutions of the set covering problem. J Heuristics 7:131–144
Han BT, Raja VT (2003) A GRASP heuristic for solving an extended capacitated concentrator location problem. Int J Inf Technol Decis Making 2:597–617
Hansen P, Mladenović N (2003) Variable neighborhood search. In: Glover F, Kochenberger G (eds) Handbook of metaheuristics. Kluwer, Dordrecht, pp 145–184
Hirsch MJ (2006) GRASP-based heuristics for continuous global optimization problems. PhD thesis, University of Florida
Hirsch MJ, Ortiz-Pena H (2009) UAV cooperative control for multiple target tracking. In: Du D-Z, Pardalos PM (eds) DIMACS/DyDAn workshop on approximation algorithms in wireless ad hoc and sensor networks, Piscataway, NJ
Hirsch MJ, Pardalos PM, Resende MGC (2006) Sensor registration in a sensor network by continuous GRASP. In: Proceeding of the MILCOM 2006, Washington, DC
Hirsch MJ, Meneses CN, Pardalos PM, Ragle MA, Resende MGC (2007a) A Continuous GRASP to determine the relationship between drugs and adverse reactions. In: Seref O, Kundakcioglu OE, Pardalos PM (eds) Data Mining, systems analysis, and optimization in biomedicine. American Institute of Physics, New York, pp 106–121
Hirsch MJ, Meneses CN, Pardalos PM, Resende MGC (2007b) Global optimization by continuous GRASP. Optim Lett 1:201–212
Hirsch MJ, Ortiz-Pena H, Sapankevych N, Neese R (2007c) Efficient flight formation for tracking of a ground target. In: Proceeding of the National Fire Control Symposium, San Diego, CA
Hirsch MJ, Pardalos PM, Resende MGC (2009) Solving systems of nonlinear equations using continuous GRASP. Nonlinear Anal Real World Appl 10:2000–2006
Hirsch MJ, Pardalos PM, Resende MGC (2010) Speeding up continuous GRASP. Eur J Oper Res 205:507–521
Holmqvist K, Migdalas A, Pardalos PM (1997) Greedy randomized adaptive search for a location problem with economies of scale. In: Bomze IM et al (eds) Developments in global optimization. Kluwer, Dordrecht, pp 301–313
Holmqvist K, Migdalas A, Pardalos PM (1998) A GRASP algorithm for the single source uncapacitated minimum concave-cost network flow problem.

In: Pardalos PM, Du D-Z (eds) Network design: connectivity and facilities location, DIMACS 40. AMS, Providence, pp 131–142

Hoos HH, Stützle T (1998) Evaluating Las Vegas algorithms—pitfalls and remedies. In: Proceedings of the 14th Conference on uncertainty in artificial intelligence, Madison, WI, pp 238–245

Klincewicz JG (1992) Avoiding local optima in the *p*-hub location problem using tabu search and GRASP. Ann Oper Res 40:283–302

Klincewicz JG, Rajan A (1994) Using GRASP to solve the component grouping problem. Nav Res Logist 41:893–912

Kontoravdis G, Bard JF (1995) A GRASP for the vehicle routing problem with time windows. ORSA J Comput 7:10–23

Laguna M, González-Velarde JL (1991) A search heuristic for just-in-time scheduling in parallel machines. J Intell Manuf 2:253–260

Laguna M, Martí R (1999) GRASP and path relinking for 2-layer straight line crossing minimization. INFORMS J Comput 11:44–52

Laguna M, Martí R (2001) A GRASP for coloring sparse graphs. Comput Optim Appl 19:165–178

Laguna M, Feo TA, Elrod HC (1994) A greedy randomized adaptive search procedure for the two-partition problem. Oper Res 42:677–687

Li Y, Pardalos PM, Resende MGC (1994) A greedy randomized adaptive search procedure for the quadratic assignment problem. In: Pardalos PM, Wolkowicz H (eds) Quadratic assignment and related problems, DIMACS 16. AMS, Providence, pp 237–261

Lourenço HR, Martin OC, Stützle T (2003) Iterated local search. In: Glover F, Kochenberger G (eds) Handbook of metaheuristics. Kluwer, Dordrecht, pp 321–353

Martí R (2001) Arc crossing minimization in graphs with GRASP. IIE Trans 33:913–919

Martin O, Otto SW, Felten EW (1991) Large-step Markov chains for the traveling salesman problem. Complex Syst 5:299–326

Martins SL, Ribeiro CC, Souza MC (1998) A parallel GRASP for the Steiner problem in graphs. In: Ferreira A, Rolim J (eds) Proceedings of IRREGULAR'98. LNCS 1457. Springer, Berlin, pp 285–297

Martins SL, Pardalos PM, Resende MGC, Ribeiro CC (1999) Greedy randomized adaptive search procedures for the Steiner problem in graphs. In: Pardalos PM, Rajasejaran S, Rolim J (eds) Randomization methods in algorithmic design, DIMACS 43. AMS, Providence, pp 133–145

Martins SL, Resende MGC, Ribeiro CC, Pardalos P (2000) A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy. J Glob Optim 17:267–283

Martins SL, Ribeiro CC, Rosseti I (2004) Applications and parallel implementations of metaheuristics in network design and routing. In: Applied computing. LNCS 3285. Springer, Berlin, pp 205–213

Martins SL, Ribeiro CC, Rosseti I (2006) Applications of parallel metaheuristics to optimization problems in telecommunications and bioinformatics. In: Talbi E-G (ed) Parallel combinatorial optimization. Wiley, New York, pp 301–325

Mateus GR, Resende MGC, Silva RMA (2011) GRASP with path-relinking for the generalized quadratic assignment problem. J Heuristics 17:527–565

Mavridou T, Pardalos PM, Pitsoulis LS, Resende MGC (1998) A GRASP for the biquadratic assignment problem. Eur J Oper Res 105:613–621

Murphey RA, Pardalos PM, Pitsoulis LS (1998a) A greedy randomized adaptive search procedure for the multitarget multisensor tracking problem. In: Pardalos PM and Du D-Z (eds) Network design: connectivity and facilities location, DIMACS 40. AMS, Providence, pp 277–301

Murphey RA, Pardalos PM, Pitsoulis LS (1998b) A parallel GRASP for the data association multidimensional assignment problem. In: Pardalos PM (ed) Parallel processing of discrete problems. The IMA volumes in mathematics and its applications 106. Springer, Berlin, pp 159–180

Nascimento MCV, Resende MGC, Toledo FMB (2010) GRASP with path-relinking for the multi-plant capacitated plot sizing problem. Eur J Oper Res 200:747–754

Oliveira CA, Pardalos PM, Resende MGC (2004) GRASP with path-relinking for the quadratic assignment problem. In: Ribeiro CC, Martins SL (eds) Proceedings of III workshop on efficient and experimental algorithms. LNCS 3059. Springer, Berlin, pp 356–368

Osman IH, Al-Ayoubi B, Barake M (2003) A greedy random adaptive search procedure for the weighted maximal planar graph problem. Comput Ind Eng 45:635–651

Pardalos PM, Pitsoulis LS, Resende MGC (1995) A parallel GRASP implementation for the quadratic assignment problem. In: Ferreira A, Rolim J (eds) Parallel algorithms for irregularly structured problems—IRREGULAR'94. Kluwer, Dordrecht, pp 115–133

Pardalos PM, Pitsoulis LS, Resende MGC (1996) A parallel GRASP for MAX-SAT problems. LNCS 1184. Springer, Berlin/New York, pp 575–585

Pardalos PM, Pitsoulis LS, Resende MGC (1997) Algorithm 769: Fortran subroutines for approximate solution of sparse quadratic assignment problems using GRASP. ACM Trans Math Softw 23:196–208

Pardalos PM, Qian T, Resende MGC (1999) A greedy randomized adaptive search procedure for the feedback vertex set problem. J Comb Optim 2:399–412

Piñana E, Plana I, Campos V, Martí R (2004) GRASP and path relinking for the matrix bandwidth minimization. Eur J Oper Res 153:200–210

Pitsoulis LS, Pardalos PM, Hearn DW (2001) Approximate solutions to the turbine balancing problem. Eur J Oper Res 130:147–155

Prais M, Ribeiro CC (2000a) Parameter variation in GRASP procedures. Investigación Operativa 9:1–20

Prais M, Ribeiro CC (2000b) Reactive GRASP: an application to a matrix decomposition problem in TDMA traffic assignment. INFORMS J Comput 12:164–176

Reghioui M, Prins C, Labadi N (2007) GRASP with path relinking for the capacitated arc routing problem with time windows. In: Giacobini M et al (eds) Applications of evolutinary computing. LNCS 4448. Springer, Berlin, pp 722–731
Resende MGC (1998) Computing approximate solutions of the maximum covering problem using GRASP. J Heuristics 4:161–171
Resende MGC (2008) Metaheuristic hybridization with greedy randomized adaptive search procedures. In: Chen Z-L, Raghavan S (eds) TutORials in operations research. INFORMS, Hanover, pp 295–319
Resende MGC, Feo TA (1996) A GRASP for satisfiability. In: Johnson DS, Trick MA (eds) Cliques, coloring, and satisfiability: the second DIMACS implementation challenge, DIMACS 26. AMS, Providence, pp 499–520
Resende LIP, Resende MGC (1999) A GRASP for frame relay permanent virtual circuit routing. In: Ribeiro CC, Hansen P (eds) Extended abstracts of the III metaheuristics International Conference, Angra dos Reis, Brazil, pp 397–401
Resende MGC, Ribeiro CC (1997) A GRASP for graph planarization. Networks 29:173–189
Resende MGC, Ribeiro CC (2003a) A GRASP with path-relinking for private virtual circuit routing. Networks 41:104–114
Resende MGC, Ribeiro CC (2003b) Greedy randomized adaptive search procedures. In: Glover F, Kochenberger G (eds) Handbook of metaheuristics. Kluwer, Dordrecht, pp 219–249
Resende MGC, Ribeiro CC (2005a) GRASP with path-relinking: recent advances and applications. In: Ibaraki T, Nonobe K, Yagiura M (eds) Metaheuristics: progress as real problem solvers. Springer, Berlin, pp 29–63
Resende MGC, Ribeiro CC (2005b) Parallel greedy randomized adaptive search procedures. In: Alba E (ed) Parallel metaheuristics: a new class of algorithms. Wiley, New York, pp 315–346
Resende MGC, Ribeiro CC (2010) Greedy randomized adaptive search procedures: advances and applications. In: Gendreau M, Potvin J-Y (eds) Handbook of metaheuristics, 2nd edn. Springer, Berlin, pp 293–319
Resende MGC, Werneck RF (2004) A hybrid heuristic for the p-median problem. J Heuristics 10:59–88
Resende MGC, Werneck RF (2006) A hybrid multistart heuristic for the uncapacitated facility location problem. Eur J Oper Res 174:54–68
Resende MGC, Pardalos PM, Li Y (1996) Algorithm 754: Fortran subroutines for approximate solution of dense quadratic assignment problems using GRASP. ACM Trans Math Softw 22:104–118
Resende MGC, Pitsoulis LS, Pardalos PM (1997) Approximate solution of weighted MAX-SAT problems using GRASP. In: Gu J, Pardalos PM (eds) Satisfiability problems, DIMACS 35. AMS, Providence, pp 393–405
Resende MGC, Feo TA, Smith SH (1998) Algorithm 787: Fortran subroutines for approximate solution of maximum independent set problems using GRASP. ACM Trans Math Softw 24:386–394

Resende MGC, Pitsoulis LS, Pardalos PM (2000) Fortran subroutines for computing approximate solutions of MAX-SAT problems using GRASP. Discret Appl Math 100:95–113

Resende MGC, Martí R, Gallego, M, Duarte A (2010) GRASP and path relinking for the max–min diversity problem. Comput Oper Res 37:498–508

Ribeiro CC, Resende MGC (1999) Algorithm 797: Fortran subroutines for approximate solution of graph planarization problems using GRASP. ACM Trans Math Softw 25:342–352

Ribeiro CC, Resende MGC (2012) Path-relinking intensification methods for stochastic local search algorithms. J Heuristics 18:193–214

Ribeiro CC, Rosseti I (2002) A parallel GRASP heuristic for the 2-path network design problem. LNCS 2400. Springer, Berlin, pp 922–926

Ribeiro CC, Rosseti I (2007) Efficient parallel cooperative implementations of GRASP heuristics. Parallel Comput 33:21–35

Ribeiro CC, Souza MC (2000) Tabu search for the Steiner problem in graphs. Networks 36:138–146

Ribeiro CC, Urrutia S (2007) Heuristics for the mirrored traveling tournament problem. Eur J Oper Res 179:775–787

Ribeiro CC, Vianna DS (2005) A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure. Int Trans Oper Res 12:325–338

Ribeiro CC, Uchoa E, Werneck RF (2002) A hybrid GRASP with perturbations for the Steiner problem in graphs. INFORMS J Comput 14:228–246

Ribeiro MH, Plastino A, Martins SL (2006) Hybridization of GRASP metaheuristic with data mining techniques. J Math Model Algorithm 5:23–41

Ribeiro CC, Martins SL, Rosseti I (2007) Metaheuristics for optimization problems in computer communications. Comput Commun 30:656–669

Ribeiro CC, Rosseti I, Souza RC (2011) Effective probabilistic stopping rules for randomized metaheuristics: GRASP implementations. LNCS 6683. Springer, Berlin, pp 146–160

Ríos-Mercado RZ, Bard JF (1998) Heuristics for the flow line problem with setup costs. Eur J Oper Res 110:76–98

Ríos-Mercado RZ, Bard JF (1999) An enhanced TSP-based heuristic for makespan minimization in a flow shop with setup costs. J Heuristics 5:57–74

Robertson AJ (2001) A set of greedy randomized adaptive local search procedure (GRASP) implementations for the multidimensional assignment problem. Comput Optim Appl 19:145–164

Santos LF, Martins SL, Plastino A (2008) Applications of the DM-GRASP heuristic: a survey. Int Trans Oper Res 15:387–416

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

Souza MC, Duhamel C, Ribeiro CC (2004) A GRASP heuristic for the capacitated minimum spanning tree problem using a memory-based local search strategy. In: Resende MGC, de Sousa JP (eds) Metaheuristics: computer decision-making. Kluwer, Dordrecht, pp 627–658

Srinivasan A, Ramakrishnan KG, Kumaram K, Aravamudam M, Naqvi S (2000) Optimal design of signaling networks for Internet telephony. In: IEEE INFOCOM 2000, Tel Aviv, Israel, vol 2, pp 707–716

Urban TL (1998) Solution procedures for the dynamic facility layout problem. Ann Oper Res 76:323–342

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82

Xu JY, Chiu SY (2001) Effective heuristic procedure for a field technician scheduling problem. J Heuristics 7:495–509

Yen J, Carlsson M, Chang M, Garcia JM, Nguyen H (2000) Constraint solving for inkjet print mask design. J Imaging Sci Technol 44:391–397