GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURES (GRASP)

MAURICIO G.C. RESENDE

ABSTRACT. This paper is a survey of greedy randomized adaptive search procedures (GRASP). GRASP is a multi-start or iterative procedure where each GRASP iteration consists of a construction phase, where a feasible solution is constructed, followed by a local search procedure that finds a locally optimal solution. The construction phase of GRASP is essentially a randomized greedy algorithm. Repeated applications of the construction procedure yields diverse starting solutions for the local search. We review a basic GRASP, followed by enhancements to the basic procedure. We conclude by surveying operations research and industrial applications of GRASP.

1. INTRODUCTION

Optimization problems that involve a large finite number of alternatives often arise in industry, government and science. In these problems, one is given a finite solution set X and a real-valued function $f : X \to \mathbb{R}$, and one seeks a solution $x^* \in X$ with $f(x^*) \leq f(x)$, $\forall\, x \in X$. Common examples include designing efficient telecommunication networks and constructing cost effective airline crew schedules. To find the optimal solution in a combinatorial optimization problem it is theoretically possible to enumerate the solutions and evaluate each with respect to the stated objective. However, from a practical perspective, it is infeasible to follow such a strategy of complete enumeration because the number of combinations often grows exponentially with the size of problem.

Much work has been done over the last five decades to develop optimal seeking methods that do not explicitly require an examination of each alternative. This research has given rise to the field of *combinatorial optimization* (see Papadimitriou and Steiglitz [55]), and an increasing capability to solve ever larger real-world problems. Nevertheless, most problems found in industry and government are either computationally intractable by their nature, or sufficiently large so as to preclude the use of exact algorithms. In such cases, heuristic methods are usually employed to find good, but not necessarily guaranteed optimal solutions. The effectiveness of these methods depends upon their ability to adapt to a particular realization, avoid entrapment at local optima, and exploit the basic structure of the problem, such as a network or a natural ordering among its components. Furthermore, restart procedures, controlled randomization, efficient data structures, and preprocessing are also beneficial. Building on these notions, various heuristic search techniques have been developed that have demonstrably improved our ability to obtain good solutions to difficult combinatorial optimization problems. The most promising of such techniques include simulated annealing [35], tabu search [27, 28, 29], genetic algorithms [30] and GRASP (Greedy Randomized Adaptive Search Procedures) [21, 22].

In this article, we review GRASP. The components of a basic GRASP heuristic are addressed and enhancements proposed to the basic heuristic are discussed. The paper concludes with a brief literature review of applications of GRASP.

Date: December 22, 1998.
AT&T Labs Research Technical Report: 98.41.1.

2. A BASIC GRASP

A GRASP is a multi-start or iterative process, in which each GRASP iteration consists of two phases, a construction phase, in which a feasible solution is produced, and a local search phase, in which a local optimum in the neighborhood of the constructed solution is sought. The best overall solution is kept as the result. The pseudo-code below illustrates a GRASP procedure for minimization in which `maxitr` GRASP iterations are done.

```
procedure grasp(f(·),g(·),maxitr,x*)
1  x* = ∞;
2  for k = 1,2,...,maxitr do
3      construct(g(·),α,x);
4      local(f(·),x);
5      if f(x) < f(x*) do
6          x* = x;
7      end if;
8  end for;
end grasp;
```

In the construction phase, a feasible solution is iteratively constructed, one element at a time. The basic GRASP construction phase is similar to the semi-greedy heuristic proposed independently by Hart and Shogan [31]. At each construction iteration, the choice of the next element to be added is determined by ordering all candiate elements (i.e. those that can be added to the solution) in a candidate list C with respect to a greedy function $g : C \to \mathbb{R}$. This function measures the (myopic) benefit of selecting each element. The heuristic is adaptive because the benefits associated with every element are updated at each iteration of the construction phase to reflect the changes brought on by the selection of the previous element. The probabilistic component of a GRASP is characterized by randomly choosing one of the best candidates in the list, but not necessarily the top candidate. The list of best candidates is called the *restricted candidate list* (RCL). This choice technique allows for different solutions to be obtained at each GRASP iteration, but does not necessarily compromise the power of the adaptive greedy component of the method. Let $\alpha \in [0,1]$ be a given parameter. The pseudo code below describes a basic GRASP construction phase.

```
procedure construct(g(·),α,x)
1  x = ∅;
2  Initialize candidate set C;
3  while C ≠ ∅ do
4      s_ = min{g(t) | t ∈ C};
5      s¯ = max{g(t) | t ∈ C};
6      RCL = {s ∈ C | g(s) ≤ s_ + α(s¯ − s_)};
7      Select s, at random, from the RCL;
8      x = x ∪ {s};
7  end while;
end construct;
```

The pseudo-code shows that the parameter α controls the amounts of greediness and randomness in the algorithm. A value $\alpha = 0$ corresponds a greedy construction procedure, while $\alpha = 1$ produces random construction.

As is the case for many deterministic methods, the solutions generated by a GRASP construction are not guaranteed to be locally optimal with respect to simple neighborhood

definitions. Hence, it is almost always beneficial to apply a local search to attempt to improve each constructed solution. A local search algorithm works in an iterative fashion by successively replacing the current solution by a better solution in the neighborhood of the current solution. It terminates when no better solution is found in the neighborhood. The *neighborhood structure* N for a problem P relates a solution s of the problem to a subset of solutions $N(s)$. A solution s is said to be *locally optimal* if there is no better solution in $N(s)$. The key to success for a local search algorithm consists of the suitable choice of a neighborhood structure, efficient neighborhood search techniques, and the starting solution.

While such local optimization procedures can require exponential time from an arbitrary starting point, empirically their efficiency significantly improves as the initial solution improves. Through the use of customized data structures and careful implementation, an efficient construction phase can be created which produces good initial solutions for efficient local search. The result is that often many GRASP solutions are generated in the same amount of time required for the local optimization procedure to converge from a single random start. Furthermore, the best of these GRASP solutions is generally significantly better than the single solution obtained from a random starting point. The pseudo-code below describes a basic local search procedure.

procedure `local`($f(\cdot)$,$N(\cdot)$,x)
1 $H = \{y \in N(x) \mid f(y) < f(x)\}$;
2 **while** $|H| > 0$ **do**
3 Select $x \in H$;
4 $H = \{y \in N(x) \mid f(y) < f(x)\}$;
5 **end while**;
end `local`;

It is difficult to formally analyze the quality of solution values found by using the GRASP methodology. However, there is an intuitive justification that views GRASP as a repetitive sampling technique. Each GRASP iteration produces a sample solution from an unknown distribution of all obtainable results. The mean and variance of the distribution are functions of the restrictive nature of the candidate list. For example, if the cardinality of the restricted candidate list is limited to one, then only one solution will be produced and the variance of the distribution will be zero. Given an effective greedy function, the mean solution value in this case should be good, but probably suboptimal. If a less restrictive cardinality limit is imposed, many different solutions will be produced implying a larger variance. Since the greedy function is more compromised in this case, the mean solution value should degrade. Intuitively, however, by order statistics and the fact that the samples are randomly produced, the best value found should outperform the mean value. Indeed, often the best solutions sampled are optimal.

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 GRASP iterations. Finally, GRASP can be trivially implemented in parallel. Each processor can be initialized with its own copy of the procedure, the instance data, and an independent random number sequence. The GRASP iterations are then performed in parallel with only a single global variable required to store the best solution found over all processors.

3. Enhancements to the Basic GRASP

A number of enhancements to the basic GRASP, presented in the previous section, have been proposed in the literature. In this section we review the use path relinking, long-term memory, the proximate optimality principle, and bias functions in a GRASP. We discuss a parallelization scheme and the use of GRASP in hybrid meta-heuristics.

3.1. **Path relinking.** Laguna and Martí [42] adapted the concept of path relinking for use within a GRASP. To test their concept, they implemented a GRASP with path relinking for the 2-layer straight line crossing minimization problem. A small set of high-quality, or elite, solutions is stored to serve as guiding solutions for path relinking. Each GRASP iteration produces a locally optimal solution x^*. A solution y^* is chosen at random from the elite set and a path of solutions linking x^* to y^* is constructed by applying a series of changes to the original solution. For example, let $x^* = (1,0,0,0)$ and $y^* = (0,1,0,1)$. A path relinking of x^* and y^* is $x^* = (1,0,0,0) \rightarrow (0,0,0,0) \rightarrow (0,1,0,0) \rightarrow (0,1,0,1) = y^*$. Each of these path solutions is evaluated for solution quality. Laguna and Martí report that often improvements to the incumbent are found in this path relinking.

3.2. **Long-term memory.** Long-term memory is the basis for tabu search. Besides path relinking, which can thought of as a form of long-term memory, other uses of long term memory have been proposed for use in a GRASP. Fleurent and Glover [26] observe the fact that the basic GRASP does not make use of information gathered in previous iterations and propose a long term memory scheme to address this issue. Prais and Ribeiro [64] propose a scheme to learn an appropriate value for the RCL parameter α.

Fleurent and Glover introduced a way to use long-term memory in multi-start heuristics such as GRASP. Their scheme maintains a set S of elite solutions to be used in the construction phase. To become an elite solution a solution s must be either better than the best member of S, or better than the worst member of S and sufficiently different from the other elite solutions. For example, one can count identical solution vector components and set a threshold for rejection. A *strongly determined variable* is one that cannot be changed without eroding the objective or changing significantly other variables. A *consistent variable* is one that receives a particular value in a large portion of the elite solution set. Let $I(e)$ be a measure of the strongly determined and consistent features of choice e, i.e. $I(e)$ becomes larger as e resembles solutions in elite set S. The intensity function $I(e)$ is used in the construction phase as follows. Recall that $g(e)$ is the greedy function. Let $E(e) = F(g(e), I(e))$ be a function of the greedy and the intensification functions. For example, $E(e) = \lambda g(e) + I(e)$. The intensification scheme biases selection from the RCL to those elements e with a high value of $E(e)$ by setting the probability of selecting e to be $p(e) = E(e)/\sum_{s \in \mathrm{RCL}} E(s)$. The function $E(e)$ can vary with time by changing the value of λ, e.g. initially λ is set to a large value and when diversification is called for, λ is decreased. A procedure for changing the value of λ is given by Fleurent and Glover. See also [11] for an application of this long-term memory strategy.

3.3. **Reactive GRASP.** The term *Reactive GRASP* was introduced by Prais and Ribeiro [64] for a GRASP that reacts to solutions produced by different settings of the RCL parameter α and seeks to adjust α to give the GRASP an appropriate level of greediness and randomness. At each GRASP iteration, the value of α is chosen from a discrete set of values $\{\alpha_1, \alpha_2, \ldots, \alpha_m\}$. The probability of selecting the value α_k is $p(\alpha_k)$, for $k = 1, 2, \ldots, m$. Reactive GRASP adaptively changes the probabilities $\{p(\alpha_1), p(\alpha_2), \ldots, p(\alpha_m)\}$ to favor values that produce good solutions. Consider applying Reactive GRASP to a minimization problem. Initially the probabilities are set as $p(\alpha_k) = 1/m$, for $i = 1, 2, \ldots, m$ so that the

values are selected uniformly. To adaptively redefine the probabilities, define $F(S^*)$ to be the value of the best solution found so far and let A_i be the average value of the solutions obtained with α_i. Prais and Ribeiro propose a period of warm-up iterations to initialize the A_i values. Periodically (say every N_α iterations) the quantities $q_i = (F(S^*)/A_i)^\delta$ are computed for $i = 1,2,\ldots,m$ and the probabilities are updated to $p(\alpha_i) = q_i/\sum_{j=1}^{m} q_j$, for $i = 1,2,\ldots,m$. Observe that the more suitable a value α_i is, the larger the value of q_i is and, consequently, the higher the value of $p(\alpha_i)$, making α_i more likely to be selected. The parameter δ can be used as an attenuation parameter. See also Díaz and Fernández [16] for an application of Reactive GRASP.

3.4. **Proximate optimality principle.** The Proximate Optimality Principal (POP) is based on the idea that "Good solutions at one level are likely to be found 'close to' good solutions at an adjacent level" [29]. Fleurent and Glover [26] provide a GRASP interpretation of this principle. They suggest that imperfections introduced during steps of GRASP construction can be "ironed-out" by applying local search during (and not only at the end of) GRASP construction. Because of efficiency considerations, a practical implementation of POP to GRASP is to apply local search during a few points in the construction phase and not during each construction iteration. See also [11] for an application of the proximate optimality principle.

3.5. **Global convergence.** Mockus et al. [52] pointed out that GRASP with a fixed nonzero RCL parameter α is not asymptotically convergent to a global optimum. During construction, a fixed RCL parameter may rule out a candidate that is present in all optimal solutions. Several remedies have been proposed to get around this problem. The most straightforward is the use of a randomly selected α [72]. In this approach, the parameter is selected at random from the continuous interval $[0,1]$ at the start of each GRASP iteration. That value is used during the entire iteration. Since a subset of the iterations are random, the algorithm becomes asymptotically globally convergent. Reactive GRASP, as described above, can also be made asymptotically globally convergent by making $\alpha_m = 1$, i.e. allowing the choice of a value that produces a random GRASP iteration. Bresina [13] introduced the concept of a bias function to select a candidate element to be included in the solution. Bresina's method, which is directly applicable to GRASP construction, also allows for purely random construction and is therefore asymptotically globally convergent. At each construction step, the elements in the candidate set C are ranked by their greedy function values. A bias value $\texttt{bias}(r)$ is assigned to the r-th ranked element. Bresina proposes several bias functions. In logarithmic bias, $\texttt{bias}(r) = 1/\log r + 1$. In linear bias, $\texttt{bias}(r) = 1/r$. In polynomial bias of order n, $\texttt{bias}(r) = 1/r^n$. In exponential bias, $\texttt{bias}(r) = 1/e^r$. Finally, in random bias, $\texttt{bias}(r) = 1$. During construction, the probability of selecting the r-th ranked candidate is $\texttt{bias}(r)/\sum_{i=1}^{|C|} \texttt{bias}(i)$. See also [11] for an application of this bias function strategy.

3.6. **Parallel GRASP.** Parallel implementation of GRASP is straightforward. Two general strategies have been proposed. In search space decomposition, the search space is partitioned into several regions and GRASP is applied to each in parallel. An example of this is the GRASP for maximum independent set [23, 69] where the search space is decomposed by fixing two vertices to be in the independent set. In iteration parallelization, the GRASP iterations are partitioned and each partition is assigned to a processor. See [54, 57, 58, 59, 67] for examples of parallel implementations of GRASP. Some care is needed so that different random number generator seeds are assigned to the different iterations. This can be done by running the random number generator through an entire

cycle, recording all N_g seeds in a `seed` array. Iteration i is started with `seed`(i). GRASP has been implemented on distributed architectures. In [59] a PVM-based implementation is described. Two MPI-based implementations are given in [4, 50]. Alvim [4] proposes a general scheme for MPI implementations. A master process manages seeds for slave processors. It passes blocks of seeds to each slave processor and awaits the slaves to indicate that they have finished processing the block and need another block. Slaves also pass back to the master the best solution found for each block of iterations.

3.7. **GRASP in hybrid metaheuristics.** GRASP has been used in hybrid metaheuristic schemes. Laguna and González-Velarde [41] proposed a GRASP in which local search is done by tabu search. See also [16, 46] for implementations of GRASP using tabu search as the local search procedure. Simulated annealing can also be used as a GRASP local search procedure if the initial temperature is low so that it remains near the neighborhood of the constructed solution. Ahuja, Orlin, and Tiwari [3] use GRASP construction as a mechanism for generating the initial population in a genetic algorithm. GRASP is used by Lourenço, Paixão, and Portugal [45] in a genetic algorithm to implement a type of crossover called *perfect offspring*.

4. Applications of GRASP

We now turn our attention to a number of GRASP implementations that have appeared in the literature, covering a wide range of applications. An early tutorial on GRASP appears in Feo and Resende [22]. We group the work into two categories, applications to operations research problems and to industrial applications.

4.1. **Operations research problems.** Applications of GRASP to operations research problems can be classified into eight categories: scheduling problems, routing problems, logic, partitioning problems, location problems, graph theoretic problems, assignment problems, and nonconvex network flow problems.

GRASP has been applied to several scheduling problems, including operations sequencing in discrete parts manufacturing [7], flight scheduling [18], just-in-time scheduling in parallel machines [41], printed wire assembly scheduling [17, 9], single machine scheduling with sequence dependent setup costs and delay penalties [24], field technician scheduling [79], flow shop with setup costs [76, 77], and bus-driver scheduling [45].

Applications of GRASP to routing problems include vehicle routing with time windows [38], vehicle routing [32], aircraft routing [5], inventory routing problem with satellite facilities [10], and permanent virtual circuit (PVC) routing [66].

Problems in logic have been approached with GRASP. These include the satisfiability problem [68], maximum satisfiability [59, 71, 72], and inference of logical clauses from examples [15].

GRASP has been applied to partitioning problems, including graph two partition [40] and number partitioning [6].

Applications of GRASP to location problems include p-hub location [36], pure integer capacitated plant location [14], location with economies of scale [33], single source capacitated plant location [16], location of concentrators in network access design [74], and maximum covering [67].

GRASP has been used for finding approximate solutions to a number of graph theoretic problems, including set covering [21], maximum independent set [23, 69], maximum clique with weighted edges [47], graph planarization [73, 75], 2-layer straight line crossing minimization [42], sparse graph coloring [43], maximum weighted edge subgraph [48], the

Steiner tree problem in graphs [49, 50], feedback vertex set in directed graphs [56], maximum clique [1, 61], and the capacitated minimum spanning tree problem [2].

Several assignment problems have been approached with GRASP. A GRASP was introduced for the quadratic assignment problem (QAP) in [44]. A parallel version of this GRASP is described in [58]. Fortran subroutines for dense and sparse QAPs can be found respectively in [70] and [60]. A modified local search for the GRASP for QAP is proposed in [65]. GRASP has been used to generate the initial population of a genetic algorithm for the QAP [3]. Long term memory schemes have been adapted to a GRASP for the QAP in [26]. A GRASP for the biquadratic assignment problem is described in [51]. GRASP has been applied to two multidimensional assignment problems [53, 78] and to the radio link frequency assignment problem [62].

GRASP has been used for finding approximate solutions to a concave-cost network flow problem [34].

4.2. **Industrial applications.** Industrial applications of GRASP can be classified into seven categories: manufacturing, transportation, telecommunications, automatic drawing, electrical power systems, military, and biology.

GRASP has been applied to several manufacturing problems, including operations sequencing in discrete parts manufacturing [7], cutting path and tool selection in computer-aided process planning [19], manufacturing equipment selection [8], component grouping [37], and printed wire assembly scheduling [17, 9].

Applications of GRASP in transportation include flight scheduling and maintenance base planning [18], intermodal trailer assignment [20], and aircraft routing in response to groundings and delay [5].

In telecommunications, GRASP has been applied to the design of SDH mesh-restorable networks [63], the Steiner tree problem in graphs [49, 50], permanent virtual circuit (PVC) routing [66], location of concentrators in network access design [74], traffic scheduling in satellite switched time division multi-access (SS/TDMA) systems [64], location of points of presence (PoPs) [67], and to the multi-criteria radio link frequency assignment problem [62].

GRASP has been applied to automatic drawing problems, including seam drawing in mosaicking of aerial photographic maps [25], graph planarization [73, 75], and 2-layer straight line crossing minimization [42].

An application to electrical power systems is transmission expansion planning [12]. A military application of GRASP is in multi-target multi-sensor tracking [53]. GRASP has been applied in biology for protein structure prediction [39].

5. Conclusion

We have surveyed the literature on greedy randomized adaptive search procedures (GRASP) in the last ten years. In these years many enhancements to the basic GRASP introduced in 1988 have been proposed. The number and variety of applications has grown and continues to grow.

References

[1] 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*, DIMACS Series on Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1999. to appear.

[2] R. K. Ahuja, J. B. Orlin, and D. Sharma. New neighborhood search structures for the capacitated minimum spanning tree problem. Technical report, Department of ISE, University of Florida, Gainesville, FL 32611-6595, 1998.

[3] R.K. Ahuja, J.B. Orlin, and A. Tiwari. A greedy genetic algorithm for the quadratic assignment problem. Technical report, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA, 1997.

[4] A.C.F. Alvim. Parallelization strategies for the metaheuristic GRASP. Master's thesis, Department of Computer Science, Catholic University of Rio de Janeiro, Rio de Janeiro, RJ 22453-900 Brazil, April 1998. In Portuguese.

[5] 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.

[6] M.F. Argüello, T.A. Feo, and O. Goldschmidt. Randomized methods for the number partitioning problem. *Computers & Operations Research*, 23(2):103–111, 1996.

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

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

[9] J.F. Bard, T.A. Feo, and S. Holland. A GRASP for scheduling printed wiring board assembly. *I.I.E. Trans.*, 28:155–165, 1996.

[10] J.F. Bard, L. Huang, P. Jaillet, and M. Dror. A decomposition approach to the inventory routing problem with satellite facilities. *Transportation Science*, 32:189–203, 1998.

[11] S. Binato, W.J. Hery, D. Loewenstern, and M.G.C. Resende. Approximate solution of the job shop scheduling problem using GRASP. Technical report, AT&T Labs Research, Florham Park, NJ 07932 USA, 1999.

[12] S. Binato, G.C. Oliveira, and J.L. Araújo. A greedy randomized adaptive search procedure for transmission expansion planning. *IEEE Trans. on Power Systems*, 1998. To appear.

[13] J.L. Bresina. Heuristic-biased stochastic sampling. In *Proceedings of the AAAI-96*, pages 271–278, 1996.

[14] H. Delmaire, J.A. Díaz, E. Fernández, and M. Ortega. Comparing new heuristics for the pure integer capacitated plant location problem. Technical Report DR97/10, Department of Statistics and Operations Research, Universitat Politecnica de Catalunya, Barcelona, Spain, 1997.

[15] 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. *Mathl. Comput. Modelling*, 27:75–99, 1998.

[16] J.A. Díaz and E. Fernández. A hybrid GRASP-Tabu Search algorithm for the single source capacitated plant location problem. Technical report, Department of Statistics and Operations Research, Universitat Politecnica de Catalunya, Barcelona, Spain, 1998.

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

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

[19] T.A. Feo and J.F. Bard. The cutting path and tool selection problem in computer-aided process planning. *Journal of Manufacturing Systems*, 8:17–26, 1989.

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

[21] 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.

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

[23] 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.

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

[25] E. Fernández and R. Martí. GRASP and Tabu Search for seam drawing in mosaicking of aerial photographic maps. Technical Report DR97/13, Department of Statistics and Operations Research, Universitat Politecnica de Catalunya, Barcelona, Spain, 1997.

[26] C. Fleurent and F. Glover. Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. *INFORMS Journal on Computing*, 1998. To appear.

[27] F. Glover. Tabu search – Part I. *ORSA Journal on Computing*, 1:190–206, 1989.

[28] F. Glover. Tabu search – Part II. *ORSA Journal on Computing*, 2:4–32, 1990.

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

[30] D.E Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.

[31] J.P. Hart and A.W. Shogan. Semi-greedy heuristics: An empirical study. *Operations Research Letters*, 6:107–114, 1987.

[32] C.A. Hjorring. *The vehicle routing problem and local search metaheuristics*. PhD thesis, University of Auckland, Auckland, New Zealand, 1995.

[33] 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.

[34] 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.

[35] S. Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34:975–986, 1984.

[36] 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.

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

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

[39] N. Krasnogor, D.A. Pelta, W. Russo, and G. Terrazas. A grasp approach to the protein structure prediction problem. Technical report, LIFIA Lab, University of La Plata, La Plata, Argentina, 1998.

[40] 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.

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

[42] M. Laguna and R. Martí. GRASP and path relinking for 2-layer straight line crossing minimization. *INFOMRS J. on Computing*, 1998. To appear.

[43] M. Laguna and R. Martí. A GRASP for coloring sparse graphs. Technical report, Graduate School of Business, University of Colorado, Boulder, CO 80309-0419 USA, May 1998.

[44] 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.

[45] H. Ramalhinho Lourenço, J.P. Paix ao, and R. Portugal. Metaheuristics for the bus-driver scheduling problem. Technical report, Department of Economics and Management, Universitat Pompeu Fabra, Barcelona, Spain, 1998.

[46] H. Ramalhinho Lourenço and D. Serra. Adaptive approach heuristics for the generalized assignment problem. Technical report, Department of Economics and Management, Universitat Pompeu Fabra, Barcelona, Spain, May 1998.

[47] E.M. Macambira and C.C. de Souza. A GRASP for the maximum clique problem with weighted edges. In *Proceedings of the XXIX Brazilian Symposium on Operations Research*, page 70, October 1997. In Portuguese.

[48] E.M. Macambira and C.N. Meneses. A GRASP algorithm for the maximum weighted edge subgraph problem. Technical report, Department of Statistics and Computation, University of Ceará, Fortaleza, CE 60740-000 Brazil, 1998. In Portuguese.

[49] 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.

[50] S.L. Martins and C.C. Ribeiro. A parallel GRASP for the Steiner problem in graphs. In *Proceedings of the Irregular'98*, 1998. To appear.

[51] 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, 1997.

[52] J. Mockus, E. Eddy, A. Mockus, L. Mockus, and G.V. Reklaitis. *Bayesian discrete and global optimization*. Kluwer Academic Publishers, 1997.

[53] 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.

[54] 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-Verlag, 1998.

[55] C.H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: Algorithms and complexity*. Prentice-Hall, 1982.

[56] 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(3), 1998.

[57] P.M. Pardalos, L. Pitsoulis, , T. Mavridou, and M.G.C. Resende. Parallel search for combinatorial optimization: Genetic algorithms, simulated annealing and GRASP. In A. Ferreira and J. Rolim, editors, *Parallel Algorithms for Irregularly Structured Problems, Proceedings of the Second International Workshop –Irregular'95*, volume 980 of *Lecture Notes in Computer Science*, pages 317–331. Springer-Verlag, 1995.

[58] 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 111–130. Kluwer Academic Publishers, 1995.

[59] P.M. Pardalos, L.S. Pitsoulis, and M.G.C. Resende. A parallel GRASP for MAX-SAT problems. *Lecture Notes in Computer Science*, 1180:575–585, 1996.

[60] 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.

[61] P.M. Pardalos, M.G.C. Resende, and J. Rappe. An exact parallel algorithm for the maximum clique problem. In R. De Leone et al., editor, *High performance algorithms and software in nonlinear optimization*. Kluwer Academic Publishers, 1998.

[62] E.L. Pasiliao. A greedy randomized adaptive search procedure for the multi-criteria radio link frequency assignment problem. Technical report, Department of ISE, University of Florida, Gainesville, FL 32611-6595, 1998.

[63] F. Poppe, M. Pickavet, P. Arijs, and P. Demeester. Design techniques for SDH mesh-restorable networks. In *Proceedings of the European Conference on Networks and Optical Communications (NOC'97), Volume 2: Core and ATM Networks*, pages 94–101, 1997.

[64] M. Prais and C.C. Ribeiro. Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment. Technical report, Department of Computer Science, Catholic University of Rio de Janeiro, Rio de Janeiro, RJ 22453-900 Brazil, January 1998.

[65] M.C. Rangel, N.M.M. de Abreu, P.O. Boa ventura Netto, and M.C.S. Boeres. A modified local search for GRASP in the quadratic assignment problem. Technical report, Production Engineering Program, COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, RJ Brazil, 1998.

[66] L.I.P. Resende and M.G.C. Resende. A GRASP for frame relay PVC routing. Technical report, AT&T Labs Research, Florham Park, NJ 07932 USA, 1997.

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

[68] 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.

[69] M.G.C. Resende, T.A. Feo, and S.H. Smith. Fortran subroutines for approximate solution of maximum independent set problems using GRASP. *ACM Trans. Math. Software*, 1998. To appear.

[70] 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.

[71] 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.

[72] 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*, 1999. to appear.

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

[74] M.G.C. Resende and O. Ulular. SMART: A tool for AT&T Worldnet access design – Location of Cascade 9000 concentrators. Technical report, AT&T Labs Research, Florham Park, NJ 07932 USA, 1997.

[75] C.C. Ribeiro and M.G.C. Resende. Fortran subroutines for approximate solution of graph planarization problems using GRASP. Technical report, AT&T Labs Research, Florham Park, NJ 07932 USA, 1997.

[76] R.Z. Ríos-Mercado and J.F. Bard. An enhanced TSP-based heuristic for makespan minimization in a flow shop with setup costs. Technical report, Department of Industrial Engineering, Texas A&M University, College Station, TX 77843-3131 USA, 1997.

[77] R.Z. Ríos-Mercado and J.F. Bard. Heuristics for the flow line problem with setup costs. *European J. of Operational Research*, 1998. To appear.

[78] A.J. Robertson. A set of greedy randomized adaptive local search procedure (GRASP) implementations for the multidimensional assignment problem, 1998.

[79] J. Xu and S. Chiu. Solving a real-world field technician scheduling problem. In *Proceedings of the International Conference on Management Science and the Economic Development of China*, pages 240–248, July 1996.

INFORMATION SCIENCES RESEARCH, AT&T LABS RESEARCH, FLORHAM PARK, NJ 07932 USA.
E-mail address: mgcr@research.att.com