A SURVEY OF MULTI-START METHODS FOR COMBINATORIAL OPTIMIZATION

RAFAEL MARTÍ, MAURICIO G.C. RESENDE, AND CELSO C. RIBEIRO

Abstract. Multi-start methods strategically sample the solution space of an optimization problem. The most successful of these methods have two phases that are alternated for a certain number of global iterations. The first phase generates a solution and the second seeks to improve the outcome. Each global iteration produces a solution that is typically a local optimum, and the best overall solution is the output of the algorithm. The interaction between the two phases creates a balance between search diversification (structural variation) and search intensification (improvement), to yield an effective means for generating high-quality solutions. This survey briefly sketches historical developments that have motivated the field, and then focuses on modern contributions that define the current state-of-the-art. We consider two categories of multi-start methods: memory-based and memoryless procedures. The former are based on identifying and recording specific types of information (attributes) to exploit in future constructions. The latter are based on order statistics of sampling and generate unconnected solutions. An interplay between the features of these two categories provides an inviting area for future exploration.

1. Introduction

The methods that provide the origins of what we now call *multi-start* procedures consist primarily of repeated applications of constructive methods. The best solution produced by these repeated applications is then normally selected for implementation. Early proposals can be found in the domains of heuristic scheduling (Muth and Thompson (1963) and Crowston et al. (1963)), the traveling salesman problem (Held and Karp (1970) and Lawler et al. (1985)), and knapsack problems with single and multiple constraints (Senju and Toyoda (1968), Wyman (1973), and Kochenberger et al. (1974)). It would be possible to go back even farther in time and identify various methods used in statistics and calculus as instances of utilizing repeated constructions to produce a preferred candidate, although such methods were not used to address problems in the realm of optimization as we view it today.

More recently, Glover (1977) makes several connections to multi-start search by means of a framework in which multi-start search includes local search to improve the starting solutions. Within this framework, procedures are given for generating starting values for variables and for generating values perturbed from other starting points. By varying the rules for the perturbation, these strategies include customary local search approaches for producing re-starts. A series of extensions of

Date: March 5, 2012.

Key words and phrases. Metaheuristics, multi-start methods, Adaptive memory programming, GRASP.

AT&T Labs Research Technical Report.

this framework are given in (Glover (1986), Glover (1989), and Glover (2000)) addressing controlled randomization, learning strategies, induced decomposition and adaptive memory processes (as introduced in tabu search). Emphasis is placed on the interaction between intensification and diversification as a means for creating a more effective search process. Several parts of the discussion bear on the area of multi-start methods. Controlled randomization classically takes two forms. The first is the well-known *random restart* approach, which injects a randomizing element into the generation of an initial starting point to which a heuristic is subsequently applied. The second classical version of this approach is the *random shakeup* procedure which, instead of restarting, periodically generates a randomized series of moves that leads the heuristic from its customary path into a region it would not otherwise reach.

1.1. **Characteristics of Early Methods.** Early multi-start methods from the optimization setting can be interpreted as using a binary representation of decision variables, starting from a null solution and selecting variables to set to 1, thus identifying assignments of jobs to machines, or edges to tours, or items to compose a knapsack, and so forth. This construction process continued until obtaining a complete or maximally feasible construction, at which point all remaining variables were implicitly assigned values of 0. We adopt this perspective of assigning values to zero-one variables as a basis for describing constructive processes within multi-start methods in general, allowing for the added provision of considering associated *destructive* processes that instead operate by successively assigning values of 0 to selected variables (where these variables would normally be assigned values of 1 in constructive processes). Different coding schemes can be used to encompass a vast array of problems within this framework, even in cases where a binary formulation that casts a problem as a mathematical program would be inappropriate or counterproductive (e.g., due to the complexity of describing the problem objective or constraints within a zero-one formulation). Later in this section, we make use of one such coding scheme. Aggregation and disaggregation methods can also be expressed within this framework by defining zero-one variables within hierarchies, but we will not focus on such approaches here.

The first multi-start methods were typically based on implementing the restarting step by randomly varying the choice of variables to receive a unit value, or at the other extreme by simply going through a pre-defined list of choice rules and applying a currently selected rule to build the current construction. In problem contexts where it was possible to modify a completed construction by moves that did not hopelessly destroy feasibility, the approach that is now commonly given the name of *local search* or *neighborhood search* was sometimes applied in conjunction with the constructive processes in an effort to improve the solutions generated. More recently, this marriage of constructive and local search procedures has become the customary way to apply multi-start methods, such as in the GRASP heuristics which we examine later in this survey. The motivation of enhancing constructed solutions is additionally joined by the motivation of using the varied re-constructions as a means of diversifying the solutions that launch the local search. In short, multi-start methods from the modern perspective embody a blend of intensification and diversification, and it is generally acknowledged that the nature of this blend is a primary determinant of the effectiveness of the overall method.

1.2. **Memory Structures.** In this survey we will chiefly focus on the features that have been found to characterize some of the best multi-start methods, rather than attempting to work our way through the back alleys of all the various methods that have been proposed over time. Among the substantial range of ways for classifying multi-start methods, we elect to employ a classification that divides multi-start methods into two main groups, consisting of memory-based versus memory-less procedures. A useful outcome of this classification is that it permits us to conveniently differentiate certain innovations that have provided important advances, according to our focus on the multi-start methods that currently rank among the leading algorithms of this genre.

Multi-start procedures were originally conceived as a way to exploit a local or neighborhood search procedure, by simply applying it from multiple random initial solutions. Modern multi-start methods usually incorporate a powerful form of diversification in the generation of solutions to help overcome local optimality. Without this diversification, such methods can become confined to a small region of the solution space, making it difficult, if not impossible, to find a global optimum.

The explicit use of memory structures constitutes the core of a large number of intelligent solvers, including tabu search (Glover, 1989), scatter search (Laguna and Martí, 2003), and path relinking (Ribeiro and Resende, 2011). These methods, generically referred to as *adaptive memory programming*, exploit a set of strategic memory designs. On the other hand, we can also find successful metaheuristics, such as simulated annealing (Kirkpatrick et al., 1983), noising methods (Charon and Hudry, 2002), and GRASP (Feo and Resende, 1995), with no memory structure in their original designs. To focus out attention memory-based and memory-less multi-start methods, we target adaptive memory programming and GRASP heuristics.

The re-start mechanism of multi-start methods can be superimposed on many different search methods. Once a new solution has been generated, a variety of options can be used to improve it, ranging from a simple greedy routine to a complex metaheuristic. An open question in order to design a good search procedure is whether it is better to implement a simple improving method that allows a great number of global iterations or, alternatively, to apply a complex routine that significantly improves a few generated solutions. A simple procedure depends heavily on the initial solution but a more elaborate method takes much more running time and therefore can only be applied a few times, thus reducing the sampling of the solution space.

2. Combinatorial Optimization

We consider in this survey a combinatorial optimization problem defined by a finite ground set $E = \{1, \ldots, n\}$, a set of feasible solutions $F \subseteq 2^E$, and an objective function $f : 2^E \to \mathbb{R}$. In its minimization version, we search an optimal solution $S^* \in F$ such that $f(S^*) \leq f(S)$, $\forall S \in F$. The ground set E, the cost function f, and the set of feasible solutions F are defined for each specific problem. For instance, in the case of the traveling salesman problem, the ground set E is that of all edges connecting the cities to be visited, F is formed by all edge subsets that determine a Hamiltonian cycle, and $f(S)$ is the sum of the costs of all edges in S. Another example is the maximum clique problem, where the ground set E is the set of all vertices of the graph, F is the set of all subsets of E for which all vertices are mutually adjacent, and $f(S)$ is the cardinality of the clique $S \in F$.

We consider next a simple algorithm to construct a feasible solution $S \in F \subseteq 2^E$ for a large class of combinatorial optimization problems. Infeasible solutions are those that contain certain subsets of 2^E and we restrict ourselves to problems for which we can easily identify infeasibilities by detecting the presence of any of these subsets in the solution under construction. This set encompasses many combinatorial optimization problems, including problems such as set covering, maximum clique, quadratic assignment, and traveling salesman. For example, in the case of the traveling salesman problem, a subset of the edges corresponding to a subtour indicates infeasibility. For the maximum clique problem, any setset of vertices whoses elements are not mutually adjacent indicates infeasibility.

```
procedure ConstructSolution
  Initialize solution: S ← ∅;
  Initialize candidate set:
      C ← {s ∈ E \ S | S ∪ {s} is not infeasible};
  while C ≠ ∅ do
  |   Select s ∈ C;
  |   Add s to solution: S ← S ∪ {s};
  |   Update candidate set:
  |       C ← {s ∈ E \ S | S ∪ {s} is not infeasible};
  end
  return S;
```

Algorithm 1: Pseudo-code for a solution construction procedure.

Algorithm 1 shows a procedure that constructs feasible solutions. The algorithm constructs the solution by adding elements of the ground set, one at a time, until a feasible solution is on hand. The procedure starts with an empty solution $S \leftarrow \emptyset$. At each step of the construction, let $\mathcal{C}$ be a subset of $E \setminus S$ such that, for all $s \in \mathcal{C}$, $S \cup \{s\}$ contains no infeasible subset of ground set elements. The construction algorithm selects some $s \in \mathcal{C}$ and adds it to S, i.e. $S \leftarrow S \cup \{s\}$, updates $\mathcal{C}$ taking into consideration the inclusion of s in S, and terminates when $\mathcal{C} = \emptyset$.

A basic multi-start procedure simply applies procedure `ConstructSolution` multiple times, returning the best solution found over all starts. This is illustrated in Algorithm 2.

In the random multi-start procedure, a random solution is built at each iteration. To accomplish this, an element is selected at random from the set of candidate elements $\mathcal{C}$ at each iteration of the construction, as can be seen in the pseudo-code `ConstructRandomSolution` shown in Algorithm 3. Algorithm 4 embeds this construction in a multi-start scheme.

In the following sections we trace some of the more salient contributions to multi-start methods, specially of the past decade. We have grouped them according to the two categories mentioned above: memory-based (Section 3) and memory-less designs (Section 4).

```
procedure MultiStart
  f* ← ∞;
  while stopping criterion not satisfied do
  |  Construct feasible solution:
  |      S ← ConstructSolution;
  |  if f(S) < f* then
  |  |  S* ← S;
  |  |  f* ← f(S);
  |  end
  end
  return S*;
```

Algorithm 2: Pseudo-code for multi-start algorithm.

```
procedure ConstructRandomSolution
  Initialize solution: S ← ∅;
  Initialize candidate set:
      C ← {s ∈ E \ S | S ∪ {s} is not infeasible};
  while C ≠ ∅ do
  |  Select s ∈ C at random;
  |  Add s to solution: S ← S ∪ {s};
  |  Update candidate set:
  |      C ← {s ∈ E \ S | S ∪ {s} is not infeasible};
  end
  return S;
```

Algorithm 3: Pseudo-code for a randomized solution construction procedure.

```
procedure RandomMultiStart
  f* ← ∞;
  while stopping criterion not satisfied do
  |  Construct random feasible solution:
  |      S ← ConstructRandomSolution;
  |  if f(S) < f* then
  |  |  S* ← S;
  |  |  f* ← f(S);
  |  end
  end
  return S*;
```

Algorithm 4: Pseudo-code for randomized multi-start algorithm.

3. Adaptive Memory Programming

From a naive standpoint, virtually all heuristics other than complete randomization induce a pattern whose present state depends on the sequence of past states, and therefore incorporate an implicit form of *memory*. However, such an implicit

memory, as indicated in Glover and Laguna (1997), does not take a form normally viewed to be a memory structure. By contrast, the explicit use of memory structures constitutes the core of a large number of intelligent solution methods. They include tabu search (Glover and Laguna, 1997), scatter search (Laguna and Martí, 2003), iterated-based methods (Lozano et al., 2012), and evolutionary path relinking (Resende and Werneck, 2004), among others. These methods focus on exploiting a set of strategic memory designs. Tabu search (TS), the metaheuristic that launched this perspective, is the source of the term *Adaptive Memory Programming* (AMP) to describe methods that use advanced memory strategies (and hence non-trivial learning) to guide a search. In linguistic terms, to define semantic hierarchies, we can say that AMP is the hypernym of tabu search such as mathematical programming is the hypernym of linear programming.

Taillard et al. (2001) characterized adaptive programming methods as those that exploit a memory structure to obtain a solution. Specifically they identified the following characteristics in these methods:

- A set of solutions or a special data structure that aggregates the particularities of the solutions produced by the search is memorized.
- A provisory solution is constructed using the data in memory.
- The provisory solution is improved using a greedy algorithm or a more sophisticated heuristic.
- The new solution is added to memory or is used to update the data structure that memorizes the search history.

Over time, various uses of memory strategies have also become incorporated into a variety of other metaheuristics. For example, a number of "hybrid" genetic and evolutionary methods have arisen that embed some form of these memory strategies within them, and more recently several have appeared that have dropped the hybrid nomenclature (and in some cases any reference to tabu search). Today it is not unusual for evolutionary methods to implement long term memory structures to record elite solutions found during the search for intensification or diversification purposes. In this survey, we examine the inclusion of memory structures in combined construction-improvement methods, comparing memory-based with memory-less designs.

To set the stage for discussing these strategies, it is useful to briefly sketch some of the features of tabu search. TS is a metaheuristic that guides a local search heuristic procedure to explore the solution space beyond local optimality. Its use of adaptive memory and associated strategies for exploiting such memory, creates a flexible search behavior and offers a means to learn improved trajectories through the solution space. The structure of a neighborhood in tabu search goes beyond that typically employed in local search by embracing the types of moves used in constructive and destructive processes (where the foundations for such moves are accordingly called *constructive* and *destructive neighborhoods*). As described in Glover and Laguna (1997), adaptive memory in these settings involves an attribute-based focus and closely depends on the elements of recency, frequency, and influence. We now review the most relevant adaptive memory programming contributions in the context of multi-start methods.

Boese et al. (1994) analyze relationships among local minima from the perspective of the best local minimum, finding convex structures in the cost surfaces.

Based on the results of that study, they propose a multi-start method where starting points for greedy descent are adaptively derived from the best previously found local minima. In the first step, Adaptive Multi-Start (AMS) heuristics generate r random starting solutions and run a greedy descent method from each one to determine a set of corresponding random local minima. In the second step, *adaptive starting solutions* are constructed based on the local minima obtained so far and improved with a greedy descent method. This improvement is applied several times from each adaptive starting solution to yield corresponding *adaptive local minima*. The authors test this method for the traveling salesman problem and obtain significant speedups over previous multi-start implementations. Hagen and Kahng (1997) apply this method for the iterative partitioning problem.

Moreno et al. (1995) propose a stopping rule for the multi-start method based on a statistical study of the number of iterations needed to find the global optimum. The authors introduce two random variables that together provide a way of estimating the number of global iterations needed to find the global optima: the number of initial solutions generated and the number of objective function evaluations performed to find the global optima. From these measures, the probability that the incumbent solution is the global optimum is evaluated via a normal approximation. Thus, at each global iteration, this value is computed and if it is greater than a fixed threshold, the algorithm stops, otherwise a new solution is generated. The authors illustrate the method using the median p-hub problem.

Hagen and Kahng (1997) implement an adaptive multi start method for a VLSI partitioning optimization problem where the objective is to minimize the number of signals sent between components. The method consists of two phases. It first generates a set of random starting points and performs the iterative (local search), thus determining a set of local minimum solutions. Then it constructs adaptive starting points derived from the best local minimum solutions found so far. The authors add a preprocessing cluster module to reduce the size of the problem. The resulting Clustering Adaptive Multi Start (CAMS) method is fast and stable, and improves upon previous partitioning results reported in the literature.

Fleurent and Glover (1999) propose some adaptive memory search principles to enhance multi-start approaches. The authors introduce a template of a constructive version of Tabu Search based on both, a set of elite solutions and the intensification strategies based on identifying *strongly determined and consistent variables* according to the following definitions:

- *Strongly determined variables* are those whose values cannot be changed without significantly eroding the objective function value or disrupting the values of other variables.
- A *consistent variable* is defined as one that receives a particular value in a significant portion of good solutions.

The authors propose the inclusion of memory structures within the multi-start framework as it is done with tabu search. Computational experiments for the quadratic assignment problem show that these methods improve significantly over previous multi-start methods like GRASP and random restart that do not incorporate memory-based strategies.

Patterson et al. (1999) introduce a multi-start framework called *Adaptive Reasoning Techniques* (ART), based on memory structures. The authors implement the short term and long term memory functions, proposed in the Tabu Search

framework, to solve the Capacitated Minimum Spanning Tree Problem. ART is an iterative, constructive solution procedure that implements learning methodologies on top of memory structures. ART derives its success from being able to learn about, and modify the behavior of a primary greedy heuristic. The greedy heuristic is executed repeatedly, and for each new execution, constraints that prohibit certain solution elements from being considered by the greedy heuristic are introduced in a probabilistic fashion. The active constraints are held in a short-term memory, while a long-term memory holds information regarding the constraints that were in the active memory for the best set of solutions.

Glover (2000) approaches the multi-start framework from a different perspective. The author views multi-start methods as an extreme version of the *strategic oscillation* approach. Strategic oscillation is a mechanism used in tabu search to allow the process to visit solutions around a "critical boundary," by approaching such a boundary from both sides. The most common application of strategic oscillation is in constrained problems, where the critical boundary is the feasibility boundary. The search process crosses the boundary from the feasible side to the infeasible side and also from the infeasible side to the feasible side. Two search principles, *persistent attractiveness* and *marginal conditional validity*, are proposed in Glover (2000) for creating improved forms of constructive multi-start and strategic oscillation methods.

- *Persistent attractiveness* states that good choices derive from making decisions that have often appeared attractive, but that have not previously been made within a particular region or phase of search. That is, persistent attractiveness also carries with it the connotation of persistently unselected within a specific domain or interval.
- *Marginal conditional validity* specifies that the more decisions are made, the consequences of imposing them cause the problem to be more restricted. Consequently, as the search progresses, future decisions face less complexity and less ambiguity about which choices are likely to be preferable. Therefore, early decisions are more likely to be bad ones or at least to look better than they should, once later decisions are made.

These concepts play a key role in deriving appropriate measures to capture information during prior search. Applied to constructive neighborhoods, strategic oscillation operates by alternating constructive and destructive phases, where each solution generated by a constructive phase is dismantled (to some degree) by the destructive phase, after which a new phase builds the solution anew. The conjunction of both phases and their associated memory structures provides the basis for an improved multi-start method. These strategies can also be joined with the target analysis (Glover and Laguna, 1997) to identify subsets of variables in 0-1 problems that are sufficient to generate optimal solutions. The creation of measures to capture information about recency, frequency, and attractiveness, makes Glover (2000) an important milestone in the multi-start literature based on memory structures.

Braysy et al. (2004) propose a multi-start local search heuristic for the vehicle routing problem with time windows. The objective in this problem is to design least cost routes for a fleet of identical capacitated vehicles to service geographically scattered customers within pre-specified service time windows. The suggested method uses a two-phase approach. In the first phase, a fast construction heuristic is used to generate several initial solutions. Then, injection trees, an extension of

the well-known ejection chain approach (Glover and Laguna, 1997) are used to reduce the number of routes. In the second phase, two new improvement heuristics, based on CROSS-exchanges (Taillard et al., 1997) are applied for distance minimization. The best solution identified by the algorithm is post-optimized using a threshold accepting post-processor with both intra-route and inter-route improvement heuristics. The resulting hybrid method is shown to be fast, cost-effective, and highly competitive.

Mezmaz et al. (2006) hybridize the multi-start framework with a model in which several evolutionary algorithms run simultaneously and cooperate to compute better solutions (called *island model*). They propose a solution method in the context of multi-objective optimization on a computational grid. The authors point out that although the combination of these two models usually provides very effective parallel algorithms, experiments on large-size problem instances are often stopped before convergence is achieved. The full exploitation of the cooperation model needs a large amount of computational resources and the management of the fault tolerance issue. In this paper, a grid-based fault-tolerant approach for these models and their implementation on the *XtremWeb grid middleware* is proposed. The approach has been experimented on the bi-objective Flow-Shop problem on a computational grid made of 321 heterogeneous Linux PCs within a multi-domain education network. The preliminary results, obtained after an execution time of several days, demonstrate that the use of grid computing effectively and efficiently exploits the two parallel models and their combination for solving challenging optimization problems. In particular, the effectiveness is improved by over 60 percent when compared with a serial meta-heuristic.

Under the template of a typical multi-start metaheuristic, Lan and DePuy (2006) propose Meta-RaPS (Meta-heuristic for Randomized Priority Search), in which several randomization methods and memory mechanisms are present. With the Set Covering Problem (SCP) as the application problem, it is found that these randomization and memory-based methods work well for Meta-RaPS. The memory methods consists of priority rules with fitness and partial construction in the construction phase to obtain good starting solutions. In the partial construction strategy, some basic elements are fixed according to the information gathered in previous iterations. As a conclusion, the authors state that the quality and efficiency of multi-start methods can be improved through the use of both memory mechanisms and randomization methods.

Beausoleil et al. (2008) consider a multi-objective combinatorial optimization problem called Extended Knapsack Problem. By applying multi-start search and path relinking their solving method rapidly guides the search toward the most balanced zone of the Pareto-optimal front (the zone in which all the objectives are equally important). The constructive process can be summarized as follows:

(1) Modify the problem by using a ghost image strategy (Glover and Laguna, 1997).
(2) Create a candidate list and a probability distribution.
(3) Construct a feasible solution for the modified problem by using the probability distribution

The Pareto relation is applied in order to designate a subset of the best generated solutions to be the current efficient set of solutions. A max-min criterion applied to the Hamming distance is used as a measure of dissimilarity in order to find

diverse solutions to be combined. The performance of this approach is compared with several state-of-the-art multi-objective evolutionary algorithms for a suite of test problems taken from the literature.

Essafi et al. (2010) propose a multi-start ant based heuristics for a machine line balancing problem. The proposed procedure is a constructive algorithm that assigns operations sequentially to stations. The algorithm builds a feasible solution step by step according to a greedy function that computes the contribution of each unassigned operation to the partial solution under construction based on operational time and weights. The selection of the operation to be added is performed with a roulette wheel mechanism based on the typical ant probability distribution (pheromones of previous assignments). The proposed heuristic is applied to solve a real industry problem.

Dhouib et al. (2010) propose a multi-start adaptive threshold accepting algorithm (MS-TA) to find multiple Pareto-optimal solutions for continuous optimization problems. Threshold accepting methods (TAs) are deterministic and faster variants of the well-known simulated annealing algorithms, in which every new move is accepted if it is not much worse than the old one. A multi-start technique is applied in this paper to the TA algorithm to allow more diversifications.

One of drawbacks of traditional multi-objective methods versus the evolutionary ones is the necessity of multiple single-objective runs to find multiple Pareto-optimal solutions. To overcome this drawback a multi-start technique is used and only one run is required to allow more diversification in the space search. This diversification consists in restarting with a new initial solution and different weight values (with respect to the objective functions) in order to diversify the search. Empirical experiments with non-linear mechanical design problems show the merit of the proposed procedure.

Villegas et al. (2010) propose two hybrid algorithms for the single truck and trailer routing problem. The first one is based on GRASP and variable neighborhood descent (VND), while the second one is and evolutionary local search (ELS). In the first one, large tours are constructed with a randomized nearest neighbor method with a restricted candidate list that ignores capacity constraints and trailer-point selection. VND is applied to improve these initial solutions obtained with GRASP. In the second one, a multi-start evolutionary search is applied starting from an initial solution (giant tour). The best solution found is strongly perturbed to obtain different solutions from which the search is re-started. The perturbation is managed by a mutation operator. The results of the computational experiments on a set of 32 randomly generated instances also unveil the robustness of the proposed metaheuristics, all of them achieving gaps to best known solutions of less than 1% even in the worst case. Among the proposed methods,the multi-start evolutionary local search is more accurate, faster, and scales better than the GRASP/VND.

4. Greedy randomized and GRASP multi-start methods

As was the case for the random multi-start procedure, an adaptive greedy algorithm constructs a feasible solution, one ground set element at a time. Instead of selecting an element at random from the set $\mathcal{C}$ of candidate elements, this algorithm computes the contribution to the cost of the solution of each candidate

element and then selects an element corresponding to the smallest cost contribution to add to the solution. A randomized version of the adaptive greedy algorithm, shown in procedure `ConstructSemiGreedySolution` of Algorithm 5, was proposed by Hart and Shogan (1987). In this variant a restricted candidate list (RCL) of good elements for selection is constructed. Let $g(s)$ be the contribution of ground set candidate element $s \in \mathcal{C}$ to the cost of the solution under construction and let $g_{\min} = \min\{g(s) \mid s \in \mathcal{C}\}$ and $g_{\max} = \max\{g(s) \mid s \in \mathcal{C}\}$. The restricted candidate list is defined as

$$\text{RCL} = \{s \in \mathcal{C} \mid g(s) \leq g_{\min} + \alpha(g_{\max} - g_{\min})\},$$

where α is a real-valued parameter such that $0 \leq \alpha \leq 1$. At each iteration of the greedy randomized construction procedure, an element is selected at random from the RCL and is added to the solution under construction. The RCL is then redefined to take into account the fact that the selected ground set element is now part of the solution under construction. Note that by varying the value of α we can make this construction procedure vary from a purely greedy procedure ($\alpha = 0$) to one that is purely random ($\alpha = 1$). Hart and Shogan (1987) called this procedure *semi-greedy*. The randomized greedy construction procedure is illustrated in Algorithm 5 and the corresponding multi-start procedure is shown in Algorithm 6.

procedure `ConstructSemiGreedySolution`(α)
  Initialize solution: $S \leftarrow \emptyset$;
  Initialize candidate set:
    $\mathcal{C} \leftarrow \{s \in E \setminus S \mid S \cup \{s\}$ is not infeasible$\}$;
  while $\mathcal{C} \neq \emptyset$ **do**
    $g_{\min} \leftarrow \min\{g(s) \mid s \in \mathcal{C}\}$;
    $g_{\max} \leftarrow \max\{g(s) \mid s \in \mathcal{C}\}$;
    RCL $\leftarrow \{s \in \mathcal{C} \mid g(s) \leq g_{\min} + \alpha(g_{\max} - g_{\min})\}$;
    Select at random $s \in$ RCL;
    Add s to solution: $S \leftarrow S \cup \{s\}$;
    Update candidate set:
      $\mathcal{C} \leftarrow \{s \in E \setminus S \mid S \cup \{s\}$ is not infeasible$\}$;
  end
  return S;

Algorithm 5: Pseudo-code for a randomized greedy solution construction procedure.

Given a feasible solution $S \in F$ of a combinatorial optimization problem, we define a *neighborhood* $N(S)$ of S to consist of all feasible solutions that can be obtained by making a predefined modification to S. For example, if S is represented as a binary vector, the k-flip neighborbood consists of all feasible solutions where the values of exactly k components of the binary vector are changed. In another example, if S is represented as a permutation n-vector (a vector on size n consisting of the entries $\{1, 2, \ldots, n\}$ in some order) the swap neighborhood is defined as all feasible solutions for which the i-th and j-th entries of the vector are interchanged,

```
procedure RandomizedGreedyMultiStart(α)
  f* ← ∞;
  while stopping criterion not satisfied do
  |  Construct a feasible randomized greedy solution:
  |      S ← ConstructSemiGreedySolution(α);
  |  if f(S) < f* then
  |  |  S* ← S;
  |  |  f* ← f(S);
  |  end
  end
  return S*;
```

Algorithm 6: Pseudo-code for randomized greedy multi-start algorithm.

for $i = 1, 2, \ldots, n-1$ and $j = i+1, \ldots, n$. We say a solution $S^* \in F$ is *locally optimal* if $f(S^*) \leq f(S)$ for all $S \in N(S^*)$. Given a feasible solution $S^0 \in F$, a local search procedure finds a locally optimal solution by exploring a sequence of neighborhoods, starting from $N(S^0)$. At the i-th iteration it explores the neighborhood of solution S^i. If there exists some solution $Y \in N(S^i)$ such that $f(Y) < f(S^i)$, it sets $S^{i+1} = Y$ and proceeds to iteration $i + 1$. Otherwise, $S^* = S^i$ is declared a locally optimal solution and the procedure stops.

Since there is no guarantee that a randomized greedy solution is locally optimal, local search can be applied after each semi-greedy construction step to attempt to find a locally optimal solution with smaller cost than that of the constructed solution. This was first proposed in a paper by Feo and Resende (1989) for set covering and was later referred to as GRASP, or greedy randomized adative search procedure, a metaheuristic for combinatorial optimization (Feo and Resende, 1995). The pseudo-code in Algorithm 7 shows a GRASP for minimization.

```
procedure GRASP(α)
  f* ← ∞;
  while stopping criterion not satisfied do
  |  Construct a feasible randomized greedy solution:
  |      S ← ConstructSemiGreedySolution(α);
  |  Find a locally optimal solution:
  |      S ← LocalSearch(S);
  |  if f(S) < f* then
  |  |  S* ← S;
  |  |  f* ← f(S);
  |  end
  end
  return S*;
```

Algorithm 7: Pseudo-code for GRASP.

We finish this survey with a brief mention to a successful hybridization of the two methodologies described in the previous sections. Specifically, Laguna and

Martí (1999) introduce Path Relinking within GRASP as a way to improve Multi-start methods. Path Relinking has been suggested as an approach to integrate intensification and diversification strategies in the context of tabu search (Glover and Laguna, 1997). This approach generates new solutions by exploring trajectories that connect high-quality solutions, by starting from one of these solutions and generating a path in the neighborhood space that leads toward the other solutions. This is accomplished by selecting moves that introduce attributes contained in the *guiding* solutions. Relinking in the context of GRASP consists of finding a path between a solution found after an improvement phase and a chosen elite solution. Therefore, the relinking concept has a different interpretation within GRASP, since the solutions found from one iteration to the next are not linked by a sequence of moves (as in the case of tabu search). The proposed strategy can be applied to any method that produces a sequence of solutions; specifically, it can be used in any multi-start procedure. Many different designs named *Evolutionary Path Relinking* have also been proposed in Resende et al. (2008).

5. Conclusions and Future Directions

The objective of this study has been to extend and advance the knowledge associated to implementing multi-start procedures. Unlike other well-known methods, it has not yet become widely implemented and tested as a metaheuristic itself for solving complex optimization problems. We described the two methodologies, adaptive memory programming and GRASP, within most of the multi-start implementations on combinatorial optimization can be found. We also described new ideas that have recently emerged within the multi-start area that add a clear potential to this framework which has yet to be fully explored.

Finally, we briefly make note of three rarely considered variations that can potentially enrich the field of multi-start methods by providing fertile offshoots worthy of investigation:

(1) Combining constructive and destructive processes via strategic oscillation (e.g., Glover (1977), Glover and Laguna (1997), Lozano et al. (2012)).
(2) Using probabilistic choice instead of pure randomization, accompanied by methods for learning good probabilities (Crowston et al., 1963).
(3) Systematically varying choice rules by changing weights attached to choice rule components (Crowston et al., 1963).

Acknowledgments

This research has been partially supported by the *Ministerio de Ciencia e Innovación* of Spain (Grant Ref. TIN2009-07516).

References

R.P. Beausoleil, G. Baldoquin, and R. Montejo. A Multi-start and path relinking methods to deal with multiobjective knapsack problems. *Annals of Operations Research*, 157:105–133, 2008.

K.D. Boese, A.B. Kahng, and S. Muddu. A new adaptive multi-start technique for combinatorial global optimisation. *Operations Research Letters*, 16:103–113, 1994.

O. Braysy, G. Hasle, and W. Dullaert. A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*, 159:586–605, 2004.

I. Charon and O. Hudry. The noising methods: A survey. In C.C. Ribeiro and P. Hansen, editors, *Essays and surveys in metaheuristics*, pages 245–261. Kluwer Academic Publishers, 2002.

W. B. Crowston, F. Glover, G. L. Thompson, and J. D. Trawick. Probabilistic and parametric learning combinations of local job shop scheduling rules. Technical Report 117, Carnegie-Mellon University, Pittsburgh, PA, 1963.

S. Dhouib, A. Kharrat, and H. Chabchoub. A multi-start threshold accepting algorithm for multiple objective continuous optimization problems. *International Journal for Numerical Methods in Engineering*, 83:1498–1517, 2010.

M. Essafi, X. Delorme, and Al Dolgui. Balancing lines with CNC machines: A multi-start and based heuristic. *CIRP Journal of Manufacturing Science and Technology*, 2:176–182, 2010.

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.

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

C. Fleurent and F. Glover. Improved constructive multi-start strategies for the Quadratic assignment problem using adaptive memory. *INFORMS Journal on Computing*, 11:198–204, 1999.

F. Glover. Heuristics for Integer Programming Using Surrogate Constraints. *Decision Sciences*, 8 (1):156–166, 1977.

F. Glover. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research*, 13 (5):533–549, 1986.

F. Glover. Tabu search. Part I. *ORSA Journal on Computing*, 1 (3):190–206, 1989.

F. Glover. Multi-start and strategic oscillation methods - Principles to exploit adaptive memory. In M. Laguna and J.L. Gonzalez-Velarde, editors, *Computing tools for modeling optimization and simulation*, pages 1–25. Kluwer Academic Publishers, 2000.

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

L.W. Hagen and A.B. Kahng. Combining problem reduction and adaptive multi-start: a new technique for superior iterative partitioning. *IEEE Trans. on CAD*, 16:709–717, 1997.

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

M. Held and R.M. Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.

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

G.A. Kochenberger, B. A. McCarl, and F. P. Wyman. A Heuristic for General Integer Programming. *Decision Sciences*, 5 (1):36–41, 1974.

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

M. Laguna and R. Martí. *Scatter search – Methodology and implementations in C*. Kluwer Academic Publishers, 2003.

G. Lan and G.W. DePuy. On the effectiveness of incorporating randomness and memory into a multi-start metaheuristic with application to the Set Covering Problem. *Computers and Industrial Engineering*, 51:362–374, 2006.

E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. *The Traveling Salesman Problem*. Wiley, 1985.

M. Lozano, F. Glover, C. García-Martínez, F.J. Rodrguez, and R. Martí. Tabu search with strategic oscillation for the quadratic minimum spanning tree. Technical Report 1A, University of Valencia, Valencia, Spain, 2012.

M. Mezmaz, N. Melab, and E.G. Talbi. Using the multi-start and island models for parallel multi-objective optimization on the computational grid. In *Second IEEE International Conference on e-Science and Grid Computing*, 2006.

J.A. Moreno, N. Mladenovic, and J.M. Moreno-Vega. An Statistical Analysis of Strategies for Multistart Heuristic Searches for p-Facility Location-Allocation Problems. In *Eighth Meeting of the EWG on Locational Analysis Lambrecht*, 1995.

J. F. Muth and G. L. Thompson. *Industrial Scheduling*. Prentice-Hall, 1963.

R. Patterson, H. Pirkul, and E. Rolland. Adaptive Reasoning Technique for the Capacitated Minimum Spanning Tree Problem. *Journal of Heuristics*, 5:159–180, 1999.

M.G. Resende, R. Martí, M. Gallego, and A. Duarte. GRASP and Path Relinking for the Max-Min Diversity Problem. *Computers and Operations Research*, 37: 498–508, 2008.

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

C.C. Ribeiro and M.G.C. Resende. Path-relinking intensification methods for stochastic local search algorithms. *J. of Heuristics*, 2011. doi: 10.1007/s10732-011-9167-1.

S. Senju and Y. Toyoda. An Approach to Linear Programming with 0-1 Variables. *Management Science*, 15:196–207, 1968.

E.D. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.Y. Potvin. A tabu search heuristic of the vehicle routing problem with time windows. *Transportation Science*, 31:170–186, 1997.

E.D. Taillard, L.M. Gambardella, M. Gendreau, and J.Y. Potvin. Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research*, 135 (1):1–16, 2001.

J.G. Villegas, C. Prins, C. Prodhon, A.L. Medaglia, and N. Velasco. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23:780–794, 2010.

F.P. Wyman. Binary Programming: A Occasion Rule for Selecting Optimal vs. Heuristic Techniques. *The Computer Journal*, 16:135–140, 1973.

(Rafael. Martí) Departamento de Estadística e Investigación Operativa, Facultad de Matematicas, Universidad de Valencia, Dr. Moliner 50, 46100 Burjassot (Valencia), Spain

E-mail address: rafael.marti@uv.es

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

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

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

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