Advanced Multi-Start Methods

R. Martí, J.M. Moreno-Vega, A. Duarte

Departamento de Estadística e Investigación Operativa
Universidad de Valencia, Spain

Departamento de Estadística, IO y Computación
Universidad de La Laguna, Spain

Departamento de Ciencias de la Computación
Universidad Rey Juan Carlos, Spain

Abstract

Heuristic search procedures that aspire to find globally optimal solutions to hard combinatorial optimization problems usually require some type of diversification to overcome local optimality. One way to achieve diversification is to re-start the procedure from a new solution once a region has been explored. In this chapter we describe the best known multi-start methods for solving optimization problems. We propose classifying these methods in terms of their use of randomization, memory and degree of rebuild. We also present a computational comparison of these methods on solving the Maximum Diversity Problem in terms of solution quality and diversification power.

Key words: Optimization, Heuristic Search, Re-Starting.

1 Introduction

Metaheuristics are high level solution methods that provide guidelines to design and integrate subordinate heuristics to solve optimization problems. These high level methods characteristically focus on strategies to escape from local optima and perform a robust search of a solution space. Most of them are based, at least partially, on a neighborhood search, and the degree to which neighborhoods are exploited varies according to the type of method.

Email addresses: rafael.marti@uv.es (R. Martí,), jmmoreno@ull.es (J.M. Moreno-Vega,), abraham.duarte@urjc.es (A. Duarte).

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. It is well known that search methods based on local optimization that aspire to find global optima usually require some type of diversification to overcome local optimality. Without this diversification, such methods can become reduced to tracing paths that are confined to a small area of the solution space, making it impossible to find a global optimum. Multi-start methods, appropriately designed, incorporate a powerful form of diversification.

There are some problems in which it turns out to be much simpler and more effective to construct solutions than to apply neighborhood based procedures. For example, in constrained scheduling problems it is difficult to define neighborhoods (i.e., structures that allow transitions from a given solution to so-called adjacent solutions) that maintain feasibility, whereas solutions can be created relatively easily by an appropriate process of construction. Something similar happens in simulation-optimization where the model treats the objective-function evaluation as a black box, making the search algorithm context-independent. In these problems the generation of solutions by step-wise constructions, according to information recorded during the search process, is more efficient than the exploration of solutions in the neighborhood of a given solution since the evaluation requires a simulation process that is usually very time-consuming. Therefore, Multi-start methods provide an appropriate framework within which to develop algorithms to solve combinatorial optimization problems.

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. This chapter is focused on studying the different strategies and methods for generating solutions to launch a succession of new searches for a global optimum.

2 An Overview

Multi-start methods have two phases: the first one in which the solution is generated and the second one in which the solution is typically (but not necessarily) improved. Then, each global iteration produces a solution (usually a local optima) and the best overall is the algorithm's output.

In recent years, many heuristic algorithms have been proposed to solve some combinatorial optimization problems. Some of them are problem-dependent and the ideas and strategies implemented are difficult to apply to different problems, while others are based on a framework that can be used directly to design solving methods for other problems. In this section we describe the most relevant procedures in terms of applying them to a wide variety of problems. We pay special attention to the adaptation of *memory structures* to multi-start methods.

The explicit use of memory structures constitutes the core of a large number of intelligent solving methods. They include tabu search, scatter search, evolutionary path relinking and some hybridizations of multi-start procedures. 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 learning, in a non-trivial sense) to guide a search.

In the following subsections we trace some of the more salient contributions to multi-start methods of the past two decades (though the origins of the methods go back somewhat farther). We have grouped them according to four categories: memory based designs (subsection 2.1), GRASP (subsection 2.2), theoretical analysis (subsection 2.3), constructive designs (subsection 2.4) and hybrid designs (subsection 2.5). Based on the analysis of these methods, we propose a classification of multi-start procedures (Section 3) in which the use of memory plays a central role.

2.1 Memory based designs

Many papers on multi-start methods that appeared before the mid-90s do not use explicit memory, as notably exemplified by the Monte Carlo random re-start approach in the context of nonlinear unconstrained optimization. Here, the method simply evaluates the objective function at randomly generated points. The probability of success approaches one as the sample size tends to infinity under very mild assumptions about the objective function. Many algorithms have been proposed that combine the Monte Carlo method with local search procedures (Rinnooy et al. (1989)). The convergence for random re-start methods is studied in Solis and Wets (1981), where the probability distribution used to choose the next starting point can depend on how the search evolves. Some extensions of these methods seek to reduce the number of complete local searches that are performed and increase the probability that they start from points close to the global optimum (Mayne and Meewella (1988)). More advanced probabilistic forms of re-starting based on memory functions were subsequently developed in Rochat and Taillard (1995) and Lokkentangen and Glover (1996).

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*. 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 (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 probabilistically introduced. The active constraints are held in a short term memory. A long term memory holds information regarding the constraints that were in the active memory for the best set of solutions.

Glover (2000) proposes approaches for creating improved forms of constructive multi-start and *strategic oscillation* methods, based on new search principles: *persistent attractiveness* and *marginal conditional validity*. 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 a variable 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.

The principle of *persistent attractiveness* says 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. The principle of *marginal conditional validity* specifies that as more are 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. Specific strategies for exploiting these concepts and their underlying principles are given in Glover (2000).

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 guide the search toward the most balanced zone of the Pareto-optimal front (the zone in which all the objectives are equally important). 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.

2.2 GRASP

One of the most well known Multi-start methods is the Greedy Adaptive Search Procedures (GRASP), which was introduced by Feo and Resende (1995). It was first used to solve set covering problems (Feo and Resende (1989)). Each GRASP iteration consists of constructing a trial solution and then applying a local search procedure to find a local optimum (i.e., the final solution for that iteration). The construction step is an adaptive and iterative process guided by a greedy evaluation function. It is iterative because the initial solution is built considering one element at a time. It is greedy because the addition of each element is guided by a greedy function. It is adaptive because the element chosen at any iteration in a construction is a function of previously chosen elements. (That is, the method is adaptive in the sense of updating relevant information from one construction step to the next.). At each stage, the next element to be added to the solution is randomly selected from a candidate list of high quality elements according to the evaluation function. Once a solution has been obtained, it is typically improved by a local search procedure. The improvement phase performs a sequence of moves towards a local optimum solution, which becomes the output of a complete GRASP iteration. Some examples of successful applications are given in Laguna et al. (1994), Resende (1998) and Laguna and Martí (1999).

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 (Glover and Laguna (1997)) in the context of tabu search. 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. Based on these ideas, Binato et al. (2001) proposed the Greedy Randomized Adaptive Path Relinking. Many different designs named *Evolutionary Path Relinking* have also been studied in Resende et al. (2008).

Prais and Ribeiro (2000) propose an improved GRASP implementation, called reactive GRASP, for a matrix decomposition problem arising in the context of traffic assignment in communication satellites. The method incorporates a memory structure to record information about previously found solutions. In Reactive GRASP, the basic parameter which restricts the candidate list during the construction phase is self-adjusted, according to the quality of the previously found solutions. The proposed method matches most of the best solutions known.

2.3 Theoretical analysis

From a theoretical point of view, Hu et al. (1994) study the combination of the *gradient algorithm* with random initializations to find a global optimum. Efficacy of parallel processing, choice of the restart probability distribution and number of restarts are studied for both discrete and continuous models. The authors show that the uniform probability is a good choice for restarting procedures.

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 heuristics (AMS) 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.

Simple forms of multi-start methods are often used to compare other methods and measure their relative contribution. Baluja (1995) compares different genetic algorithms for six sets of benchmark problems commonly found in the GA literature: Traveling Salesman Problem, Job-Shop Scheduling, Knapsack, Bin Packing, Neural Network Weight Optimization, and Numerical Function Optimization. The author uses the multi-start method (Multiple Restart Stochastic Hill-climbing, MRSH) as a baseline in the computational testing. Since solutions are represented with strings, the improvement step consists of a local search based on random flip of bits. The results indicate that using Genetic Algorithms for the optimization of static functions does not yield a benefit, in terms of the final result obtained, over simpler optimization heuristics. Other comparisons between MRSH and GAs can be found, for example, in Ackley (1987) or Wattenberg and Juels (1994).

2.4 Constructive designs

Multi-start procedures usually follow a global scheme in which generation and improvement are alternated for a certain number of iterations; but there are some applications in which the improvement can be applied several times within a global iteration. In the *incomplete construction methods*, the improvement phase is periodically invoked during the construction process of the partial solution rather than after the complete construction, as it is usually done. See Russell (1995) and Chiang and Russell (1995) for successful applications of this approach to vehicle routing.

Hickernell and Yuan (1997) present a multi-start algorithm for unconstrained global optimization based on *quasirandom samples*. Quasirandom samples are sets of deterministic points, as opposed to random points, that are evenly distributed over a set. The algorithm applies an inexpensive local search (steepest descent) on a set of quasirandom points to concentrate the sample. The sample is reduced replacing worse points with new quasirandom points. Any point that is retained for a certain number of iterations is used to start an efficient complete local search. The algorithm terminates when no new local minimum is found after several iterations. An experimental comparison shows that the method performs favorably with respect to other global optimization procedures.

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: (1) generate a set of random starting points and perform the iterative (local search) algorithm, thus determining a set of local minimum solutions; and (2) construct 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 method (CAMS) is fast and stable and improves upon previous partitioning results reported in the literature.

Tu and Mayne (2002) describe a multi-start with a clustering strategy for constrained optimization problems. It is based on the characteristics of non-linear constrained global optimization problems and extends a strategy previously tested on unconstrained problems. In this study, variations of multi-start with clustering are considered including a simulated annealing procedure for sampling the design domain and a quadratic programming (QP) sub-problem for cluster formation. The strategies are evaluated by solving 18 non-linear mathematical problems and six engineering design problems. Numerical results show that the solution of a one-step QP sub-problem helps predict possible regions of attraction of local minima and can enhance robustness and effectiveness in identifying local minima without sacrificing efficiency. In comparison with other multi-start techniques, the strategies of this study are superior in terms of the number of local searches performed, the number of minima found and the number of function evaluations required.

Bronmo et al. (2007) present a multi-start local search heuristic for a typical ship scheduling problem. Their method generates a large number of initial solutions with an insertion heuristic partially based on random elements. The best initial solutions are improved by a local search heuristic that is split into a quick and an extended version. The quick local search is used to improve a given number of the best initial solutions. The extended local search heuristic is then used to further improve some of the best solutions found.

The multi-start local search heuristic is compared with an optimization-based solution approach with respect to computation time and solution quality. The computational study shows that the multi-start local search method consistently returns optimal or near-optimal solutions to real-life instances of the ship scheduling problem within a reasonable amount of CPU time.

2.5 Hybrid designs

Ulder et al. (1990) combine genetic algorithms with local search strategies to improve previous genetic approaches for the travelling salesman problem. They apply an iterative algorithm to improve each individual, either before or while being combined with other individuals to form a new solution (offspring). The combination of these three elements: *Generation*, *Combination* and *Local Search*, extends the paradigm of Re-Start and establishes links with other metaheuristics such as Scatter Search (Glover (2000)) or Memetic Algorithms (Moscato (1999)).

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 solving method in the context of multi-objective optimization on the 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..

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. Some metaheuristics, such as GRASP, launch limited local searches from numerous constructions (i.e., starting points). In most tabu search implementations, the search starts from one initial point and if a restarting procedure is also part of the method, it is invoked only a limited number of times. However, the inclusion of re-starting strategies within the Tabu Search framework has been well documented in several papers (see for example Glover (1977) and Glover and Laguna (1997)). In Martí et al. (2001) the balance between restarting and search-depth (i.e., the time spent searching from a single starting point) is studied in the context of the Bandwidth Matrix Problem. They tested both alternatives and concluded that it was better to invest the CPU time to search from a few starting points than re-starting the search more often. Although we cannot draw a general conclusion from these experiments, the experience in the current context and in previous projects indicates that some metaheuristics, like Tabu Search, need to reach a critical search depth to be effective. If this search depth is not reached, the effectiveness of the method is severely compromised.

3 A Classification

We have found three key elements in multi-start methods that can be used for classification purposes: memory, randomization and degree of rebuild. The choices for each one of these elements are not restricted to the extreme cases where the element is simply present or absent, but represent a whole continuum between the extremes that can be labeled as:

- *Memory/Memory-less*
- *Systematic/Randomized*
- *Rebuild/Build-from-scratch*

The **Memory** classification refers to elements that are common to certain previously generated solutions. As in the Tabu Search framework (Glover and Laguna, 1997), such memory provides a foundation for incentive-based learning, by means of incentives that reinforce actions leading to good solutions or deterrents that discourage actions leading to bad ones. Thus, instead of simply resorting to randomized re-starting processes, in which the current decisions do not get any benefit from the knowledge accumulated during prior search, specific types of information are identified to exploit history. On the other hand, memory avoidance (via the *Memory-less* classification) is employed in a variety of methods where the construction of unconnected solutions is viewed as a means of strategically sampling the solution space. It should be noted that memory is not restricted to recording good solutions (or attributes of these solutions) but also includes recording solutions that exhibit diversity.

Starting solutions can be randomly generated or, on the contrary, they can be generated in a systematic way. ***Randomization*** is a very simple way of achieving diversification, but with no control over the diversity achieved since in some cases randomization can obtain very similar solutions. Moreover, there are a variety of forms of diversity that can be more important for conducting an effective search process than the haphazard outcomes of randomization. More systematic mechanisms are available to control the similarities among solutions produced, as a way to yield outcomes exhibiting a useful range of structural differences. Between the extremes of *Randomized* and *Systematic* (or deterministic) generation of solutions lie a significant number of possibilities. These can range from imposing deterministic controls on a randomized process to joining randomized and deterministic processes in various forms of alternation. The GRASP method discussed later combines several of these intermediate possibilities.

The **Degree of Rebuild** is a measure of the number or proportion of elements that remain fixed from one generation to another. Most applications *build* the solution at each generation *from scratch*, but some strategies fix (or lock-in) some elements during the construction process that have appeared in previously generated solutions. Such an approach was proposed in the context of identifying and then iteratively exploiting strongly determined and consistent variables in Glover (1977). This selective way of fixing elements, by reference to their previous impact and frequency of occurrence in various solution classes, is a memory-based strategy of the type commonly used in tabu search. This type of approach is also implicit in the operation of Path Relinking (Glover and Laguna, 1993) which generates new solutions by exploring trajectories that connect high-quality solutions. In this case the process seeks to incorporate the attributes of previously generated elite solutions by creating inducements to favor these attributes in currently generated solutions. In an extreme case all the elements in the new solution will be determined (and fixed) by the information generated from the set of elite solutions considered. This is labeled as (complete) Rebuild.

4 The Maximum Diversity Problem

The problem of choosing a subset of elements with maximum diversity from a given set is known as the Maximum Diversity Problem (MDP). This problem has a wide range of practical applications involving fields such as medical treatments, environmental balance, immigration policies and genetic engineering, among others (Glover 1998). The MDP has been studied by numerous authors, most prominent among them being Kuo et al. (1993), who described four formulations of the problem, ranging from the most intuitive to the most efficient, and which also served to show that the MDP is NP-hard. In

1996, Ghosh proposed a multi-start method and proved the completeness of the problem. Later, Glover et al. (1998) proposed four deterministic heuristic methods, two of them constructive and the other two destructive. Silva et al. (2004) presented a multi-start algorithm based on the GRASP methodology. Specifically, they described three constructive methods, called KLD, KLDv2 and MDI, and two improvement methods: LS, which is an adaptation of the one proposed by Ghosh, and SOMA, based on a VNS implementation.

The MDP can be formally described as a combinational optimization problem which can be stated as follows: let $S = \{s_i : i \in N\}$ be a set of elements where $N = \{1, 2, \ldots, n\}$ is the set of indexes. Each element of the set $s_i \in S$ may be represented by a vector $s_i = (s_{i_1}, s_{i_2}, .., s_{i_r})$. Let d_{ij} be the distance between two elements s_i and s_j and let m (with $m < n$) be the desired size of the maximum diversity set. Within this context, the solution of the MDP consists in finding a subset Sel of m elements of S ($Sel \subset S$ and $|Sel| = m$) in order to maximize the sum of the distances between the selected elements. Mathematically, the MDP may be rewritten as a decision problem in the following terms:

$$\max\ z = \sum_{i<j} d_{ij} x_i x_j$$

subject to

$$\sum_{i=1}^{n} x_i = m$$

$$x_i \in \{0, 1\}\ i = 1, \ldots, n$$

where $x_i = 1$ indicates that element s_i has been selected.

Two constructive algorithms are proposed in order to use a multi-start scheme to solve the MDP, one of them with memory and the other without. Each algorithm is described in turn in the following sections.

4.1 Multi-Start Without Memory (MSWoM)

The Multi-Start Without Memory (MSWoM) algorithm consists of a GRASP based constructive procedure and a first improvement local search. This approach was inspired by a heuristic method proposed in Glover et al. (1998). In each step, the constructive procedure adds a high quality element (given by a greedy function) to the set Sel. The non-selected elements are contained in the set $S - Sel$. The set Sel is initially empty, meaning that all the elements might be selected. The algorithm starts by selecting an element from S at random and placing it in the set Sel. The distance from all the non-selected elements $s_i \in S - Sel$ to Sel is then computed as follows:

$$d(s_i, Sel) = \sum_{s_j \in Sel} d(s_i, s_j) \tag{1}$$

which serves to arrange all the non-selected elements. To select the next element for inclusion in the set Sel, an ordered list L is constructed with all the elements $s_i \in S - Sel$ at a percent α of the maximum distance. Mathematically, L is defined as:

$$L = \{s_i \in S - Sel / d(s_i, Sel) \geq d_{min} + \alpha(d_{max} - d_{min})\} \tag{2}$$

where

$$d_{max} = \max_{s_i \in S - Sel} d(s_i, Sel) \qquad d_{min} = \min_{s_i \in S - Sel} d(s_i, Sel)$$

The next element introduced in set Sel is chosen at random from among the elements in L so as to ensure it has a minimum quality percentage, set by α. So, it is not a purely greedy selection which would lead to a local optimum. This procedure is repeated until m elements have been chosen ($|Sel| = m$) such that Sel contains the solution to the problem. After $niter$ executions of the process, the arithmetic mean of the $niter$ solutions will typically be worse than if the solution had been constructed by taking the element with a maximum distance over those already selected, although some of the $niter$ solutions will probably improve on this value.

For the algorithm to have a reactive behavior, the parameter α is initially set at 0.5 and then adjusted dynamically depending on the quality of the solutions obtained; that is, if after $niter/5$ consecutive iterations, the best solution has not improved, then α is increased by 0.1 (up to a maximum of 0.9).

The improvement method is based on a simplification of the local search described in Ghosh (1996), which seeks to increase the efficiency of the local search. The proposed method is classified as a first improvement local search which, as described in Laguna et al. (1999), not only tends to yield better results than the best improvement strategies, but also requires much less time. It does so by factoring the contribution from each element s_i in Sel; that is, for each element $s_i \in Sel$, its contribution d_i to the objective function is:

$$d_i = \sum_{s_j \in Sel} d_{ij} = d(s_i, Sel) \tag{3}$$

with the objective function defined as:

$$z = \frac{1}{2} \sum_{s_i \in Sel} d_i \tag{4}$$

Subsequently, the element $s_{i^*} \in Sel$ with the lowest contribution to the current solution is selected; that is, the element $s_{i^*} \in Sel$ with the lowest value of d_{i^*}, such that $s_{i^*} \in Sel$ is exchanged with the first element $s_j \in S - Sel$ that increases the value of the objective function (where the elements in $S - Sel$ are examined in lexicographical order). The search procedure continues for as long as the objective function improves by extracting the element from the set Sel which contributes the least and inserting another from $S - Sel$ which improves the value of the objective function. When there is no improvement, the second least-contributing element is used, and so on. This procedure is continued until no further improvement is obtained.

4.2 Multi-Start With Memory (MSWM)

Multistart with Memory (MSWM) is the second multistart algorithm described in Duarte and Martí (2007). The method uses memory both in the solution construction and improvement phases. These strategies are integrated within the Tabu Search methodology (Glover and Laguna, 1997).

In each iteration, the constructive algorithm penalizes the frequency of use of those elements which appeared in previous solutions. The procedure also rewards those elements which previously appeared in high quality solutions. To implement this algorithm, the number of times element s_i was selected in previous constructions is stored in $freq[i]$. The maximum value of $freq[i]$ for all i is stored in $maxfreq$. The average value of the solutions in which element s_i has appeared is stored in $quality[i]$. In addition, max_q stores the maximum value of $quality[i]$ for all i. The evaluation of each non-selected element in the current construction is modified depending on these values, thus favoring the selection of low-frequency, high-quality elements. This is achieved by using the following expression instead of the distance described in Eq. (3) between an element and the set of selected elements:

$$d'(s_i, Sel) = d(s_i, Sel) - \beta range(Sel) \frac{freq[i]}{max\_freq} + \delta range(Sel) \frac{quality[i]}{max\_q}$$

with

$$range(Sel) = \max_{s_j \in S - Sel} d(s_j, Sel) - \min_{s_j \in S - Sel} d(s_j, Sel)$$

where β and δ are parameters that quantify the contributions of the frequency penalization and the reward for quality. Both are adjusted experimentally. The purpose of the $range(Sel)$ parameter is to smooth the changes in the penalty function.

The set Sel is initially empty, meaning any element can be selected. The algorithm starts by selecting an element from S at random and inserting it in the set Sel. It then computes the distance $d'(s_i, Sel)$ for each element $s_i \in S - Sel$, which in the first construction would correspond with $d(s_i, Sel)$, since $freq[i] = quality[i] = 0$. The chosen element i^* is the one that:

$$d'(s_{i^*}, Sel) = \max_{s_i \in S}\{d'(s_i, Sel)\}$$

It is then inserted in Sel, after which the frequency vector is updated. This procedure is repeated until m elements have been chosen. Once a solution is constructed, the quality vector is updated. The tabu multi-start method executes this procedure $niter$ times, in such a way that with each construction the distances between an element and the set of those already selected is updated depending on its past history.

The improvement method is a modification of the one described above with the added feature of a short-term memory based on the exchange of an element between Sel and $S - Sel$. One iteration of this algorithm consists of randomly selecting an element $s_i \in Sel$. The probability of selecting this element is inversely proportional to its associated d_i value. That element of Sel is substituted by the first element $s_j \in S - Sel$ which improves the value of the objective function. If this element does not exist, then the one which degrades the least the objective function is chosen (in such a way that an exchange is always performed). When this exchange is carried out, both s_i, and s_j take on a tabu status for $TabuTenure$ iterations. Consequently, it is forbidden to remove element ij from set Sel (respectively, element s_i from set $S - Sel$) for that number of iterations. The tabu search process continues until $MaxIter$ consecutive iterations are executed without improving the best value obtained thus far.

4.3 Experimental results

To illustrate the behavior of the two multi-start algorithms summarized in this paper and proposed in Duarte and Martí (2007), we present a comparison with two other previously reported algorithms. Specifically, the MSWoM and MSWM algorithms are compared with the D2 constructive algorithm, proposed in Glover et al. (1998) along with the improvement method described in Ghosh (1996), and the KLDv2 algorithm and its respective improvement procedure, presented in Silva et al. (2004), which represent the best methods for this problem. All the algorithms were coded in C and compiled with Borland Builder 5.0, optimized for maximum speed. The experiments were carried out on a 3-GHz Pentium IV with 1 GB RAM.

The algorithms were executed on three sets of instances:

(1) **Silva:** 20 $n \times n$ matrices with random integer values generated from a $[0, 9]$ uniform distribution with $n \in [100, 500]$ and $m \in [0.1n, 0.4n]$.
(2) **Glover:** 20 $n \times n$ matrices in which the values are the distances between each pair of points with Euclidean coordinates randomly generated in $[0, 10]$. These n points have r coordinates, with $r \in [2, 21]$.
(3) **Random:** 20 $n \times n$ matrices with real weights generated from a $(0, 10)$ uniform distribution with $n = 2000$ and $m = 200$. It should be noted that these were the largest problem instances solved in the references consulted.

Tables 1, 2 and 3 compare MSWoM, MSWM, D2 + LS and KLDv2+LS. These tables show the average percentage deviation for each procedure with respect to the best known solution (in each experiment, since the optimal values are unknown), the number of best solutions and the number of constructions and improvements made by the algorithm in 10 seconds (stopping criterion).

Table 1. Constructive methods - *Silva*- type examples

	D2 + LS	KLDv2 + LS	MSWoM	MSWM
Dev.	1.722%	1.079%	0.0377%	0.0130%
♯ Best	2	5	12	13
♯ Const.	5140.5	5	12	13

Table 2. Constructive methods - *Glover*- type examples

	D2 + LS	KLDv2 + LS	MSWoM	MSWM
Dev.	0.018%	0.006%	0.000%	0.000%
♯ Best	16	18	20	20
♯ Const.	2149.6	971.0	790.4	397.5

Table 3. Constructive methods - *Random*- type examples

	D2 + LS	KLDv2 + LS	MSWoM	MSWM
Dev.	1.270%	1.219%	0.204%	0.099%
♯ Best	0	0	7	15
♯ Const.	128.1	3.5	12	14.8

The conclusion that may be drawn from these tables is that the proposed multi-start methods substantially improve on previous algorithms, with regard to both the deviation from the best known values and the number of times that value is found. Moreover, the experiments conducted also show that the use of memory, at least for the instances tested, leads to better results. Note that in the case of *Glover*-type examples, the algorithms studied yield very similar values. This fact indicates that these are the simplest problem instances, and consequently say little about the quality of each algorithm. At the other extreme are the *Random*-type examples, where substantial improvements are obtained with the multi-start methods.

5 Conclusion

The objective of this study was to extend and advance the knowledge multi-start methods. Unlike other well-known methods, these procedures have not yet become widely implemented and tested as a metaheuristic themselves for solving complex optimization problems. We have shown 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.

Our findings disclose the fact that memory appears to play an important role during both the constructive and the improvement phase of a multi-start procedure. This effect may be due to the fact that the repeated application of the constructive phase operates primarily as a diversification process, and the introduction of memory structures guides the diversification in an efficient way. On the other hand, the benefits associated with the inclusion of memory structures in the local search (improvement phase) has been extensively documented in the Tabu Search literature. Our results with the Maximum Diversity Problem are in line with these previous references. The comparison between memory-based and memory-less designs provides an interesting area for future research.

Acknowledgements

The authors want to thank Prof. Fred Glover for his valuable comments to improve this chapter in both presentation and content.

This research was partially supported by the *Ministerio de Educación* y Ciencia (TIN2006-02696) and by the Comunidad de Madrid-Universidad Rey Juan Carlos project (CCG08-URJC/TIC-3731).

References

Ackley, D.H. (1987) An empirical study of bit vector function optimization Genetic Algorithms and simulated annealing, Ed. Davis, Morgan Kaufmann Publishers.

Baluja, S. (1995) An Empirical Comparison of 7 iterative evolutionary function optimization heuristics, School of computer science, Carnegie Mellon University.

Beausoleil, R.P., Baldoquin, G., Montejo, R.A Multi-start and path relinking methods to deal with multiobjective knapsack problems, Annals of Operations Research, Vol. 157 (1), pp. 105-133.

Binato, S., Faria Jr. Haroldo and Resende, M.G.C. (2001), Greedy Randomized Adaptive Path Relinking, MIC2001 conference proceedings.

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

Bronmo, G., Christiansen, M., Fagerholt, K., Nygreen, B. A multi-start local search heuristic for ship scheduling-a computational study, Computers and Operations Research, Vol. 34 (3), pp. 900-917.

Campos, V., Laguna, M. and Martí R. (1999) Scatter Search for the Linear Ordering Problem New Ideas in Optimization, Eds Corne, Dorigo and Glover, Mc Graw Hill, pp 331-341.

Chiang, W.C. and Russell, R.A. (1995) Simulated annealing metaheuristics for the vehicle routing problems with time windows, Annals of Operations Research, Vol. 60.

Duarte, A. and R. Martí (2007) Tabu Search and GRASP for the Maximum Diversity Problem, European Journal of Operational Research, Vol. 178, pp. 71-84.

Feo, T. and M. G. C. Resende (1989) A probabilistic Heuristic for a Computationally Difficult Set Covering Problem, Operations Research Letters, Vol. 8, pp. 6771

Feo, T. and M. G. C. Resende (1995) Greedy Randomized Adaptive Search Procedures, Journal of Global Optimization, vol. 2, pp. 1-27.

Fleurent C. and Glover F. (1999) Improved constructive multi-start strategies for the Quadratic assignment problem using adaptive memory INFORMS Journal on Computing.

Glover, F. (1977) Heuristics for Integer Programming Using Surrogate Constraints, Decision Sciences, Vol. 8, pp. 156-166.

Glover, F. (1990) Tabu Search: A tutorial, Interfaces, Vol. 20, pp. 74-94.

Glover, F. (2000) Multi-start and strategic oscillation methods - Principles to exploit adaptive memory, Computing tools for modeling optimization and simulation, Eds Laguna and Gonzalez-Velarde, Kluwer, pp 1-25.

Glover, F. and M. Laguna (1993) Tabu Search, Modern Heuristic Techniques for Combinatorial Optimization Problems, C. Reeves (Ed.), Blackwell Scientific Publishing, Oxford, pp. 70-141.

Glover, F. and M. Laguna (1997) Tabu Search, Kluwer Academic Publishers, Boston.

Glover, F. , M . Laguna and R. Martí (2000) Scatter Seach to appear in Theory and Applications of Evolutionary Computation: Recent Trends, A. Ghosh and S. Tsutsui (Eds.), Springer-Verlag

Grotschel, M., M. Junger and G. Reinelt (1984), A Cutting Plane Algorithm for the Linear Ordering Problem, Operations Research, vol. 32, no. 6, pp. 1195-1220.

Hagen, L.W. and Kahng A.B. (1997) Combining problem reduction and adaptive multi-start: a new technique for superior iterative partitioning, IEEE Trans. on CAD, Vol. 16 (7), pp. 709-717.

Hickernell, F.J. and Yuan, Y. (1997) A simple multistart algorithm for global optimization, OR Transactions, Vol. 1 (2).

Hu, X., Shonkwiler, R. and Spruill, M.C. (1994) Random restarts in global optimization Georgia Institute of technology, Atlanta.

Laguna, M., T. Feo and H. Elrod (1994) A Greedy Randomized Adaptive Search Procedure for the 2-Partition Problem, Operations Research, vol. 42, no. 4, pp. 677-687.

Laguna, M. and Martí, R. (1999) GRASP and Path relinking for 2-layer straight line crossing minimization INFORMS journal on computing, vol. 11 (1), pp. 44-52.

Lokketangen, A. and Glover, F. (1996) Probabilistic move selection in tabu search for 0/1 mixed integer programming problems Meta-Heuristics: Theory and Practice, Kluwer, pp. 467-488.

LOLIB (1997): The Linear Ordering Problem Library. http://www.iwr.uni-heildelberg.de/iwr/comopt/soft/LOLIB/LOLIB.html.

Martí, R., Laguna, M., Glover, F. and Campos, V. (2001) Reducing the bandwidth of a sparse matrix with tabu search, European Journal of Operational Research, vol. 135, pp. 450-459.

Mayne, D.Q. and Meewella, C.C. (1988) A non-clustering multistart algorithm for global optimization, Analysis and Optimization of Systems, Ed. Bensoussan and Lions, Lecture Notes in control and information sciences, vol. 111, springer verlag.

Mezmaz, M., Melab, N., Talbi, E.-G. Using the multi-start and island models for parallel multi-objective optimization on the computational grid, Second IEEE International Conference on e-Science and Grid Computing, art. no. 4031085.

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

Moscato, P. (1999) Memetic Algorithms New ideas in optimization Eds. Corne, Dorigo and Glover, Mc Graw Hill, pp. 219-235.

Osman, I.H. and Kelly, J.P. (1996) Meta-Heuristics: An Overview, Meta-Heuristics: Theory and Practice, Kluwer, pp 1-23.

Prais, M. and Ribeiro C.C. (2000) Reactive GRASP: An application to a matrix decomposition problem in TDMA traffic assignment vol 12, 164-176.

Rochat and Taillard (1995) Probabilistic diversification and intensification in local search for vehicle routing Journal of heuristics, Vol. 1, 1, pp 147-167.

Resende M. G. C. (1998) Computing approximate solutions for the maximum covering problem using GRASP Journal of Heuristics, Vol. 4, pp. 161 171.

Resende M. G. C., Mart, R., Gallego, M. and Duarte A. (2008) GRASP and Path Relinking for the Max-Min Diversity Problem, Computers and Operations Research, to appear.

Patterson R., Pirkul H. and Rolland E (1999) Adaptive Reasoning Technique for the Capacitated Minimum Spanning Tree Problem, Journal of Heuris-

tics, 5, pp. 159-180.
Rinnooy Kan, A.H.G. and Timmer, G.T. (1989) Global Optimization Handbooks in operations research and management science, Vol. 1, Ed. Rinnoy Kan and Todds, North Holland, pp. 631-662.
Russell, R.A. (1995) Hybrid heuristics for the vehicle routing problem with time windows, Transportation Science.
Solis, F. and Wets, R. (1981) Minimization by random search techniques, Math. of Operations Research, Vol. 6, pp. 19-30.
Tu, W., Mayne, R.W (2002) An approach to multi-start clustering for global optimization with non-linear constraints, International Journal for Numerical Methods in Engineering, Vol. 53 (9), pp. 2253-2269.
Ulder, N.L.J., Aarts, E.H.L., Bandelt, H.J., Van Laarhoven, P.J.M. and Pesch, E. (1990) Genetic Local Search algorithms for the traveling salesman problem, Parallel problem solving from nature, Eds. Schwefel and Mnner, Springer Verlag, pp. 109-116.
Wattenberg, M. and Juels, A. (1994) Stochastic Hillclimbing as a baseline method for evaluationg genetic algorithms, University of California at Berkeley, CSD94-834.