Intelligent Multi-Start Methods

R. Martí, R. Aceves, M.T. León, J.M. Moreno-Vega, and A. Duarte

Abstract Heuristic search procedures aimed at finding 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, which constitutes a multi-start procedure. In this chapter we describe the best known multi-start methods for solving optimization problems. We also describe their connections with other metaheuristic methodologies. 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 to illustrate the efficiency of the multi-start methodology in terms of solution quality and diversification power.

Rafael Martí
Departamento de Estadística e Investigación Operativa
Universidad de Valencia, Spain, e-mail: rafael.marti@uv.es

Ricardo Aceves
Departamento de Ingeniería de Sistemas
Universidad Nacional Autónoma de México, e-mail: aceves@unam.mx

Maria Teresa León
Departamento de Estadística e Investigación Operativa
Universidad de Valencia, Spain, e-mail: teresa.leon@uv.es

Jose Marcos Moreno-Vega
Departamento de Ingeniería Informática y de Sistemas
Universidad de La Laguna, Spain, e-mail: jmmoreno@ull.es

Abraham Duarte
Departamento de Ciencias de la Computación
Universidad Rey Juan Carlos, Spain, e-mail: abraham.duarte@urjc.es

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.

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 are aimed at finding 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.

For some problems, construction procedures are more effective than 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 construction process. 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 stepwise 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 focuses on the different strategies and methods for generating solutions to launch a succession of new searches for a global optimum. We illustrate the efficiency of the multi-start methodology with a computational comparison of different methods on solving the Maximum Diversity Problem. This chapter complements a recent survey [44] devoted to multi-start methods in the context of combinatorial optimization. In particular, the survey sketches historical developments that have motivated these methods and focuses on several contributions that defined the state-of-the-art of the field in 2013.

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 [16], scatter search [35], iterated-based methods [40], evolutionary path relinking [56], 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 2.5), constructive designs (Subsection 2.3) and hybrid designs (Subsection 2.4). 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 [57]. The convergence for random re-start methods is studied in [62], 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 [45]. More advanced probabilistic forms of re-starting based on memory functions were subsequently developed in [58, 38].

Fleurent and Glover [13] propose some adaptive memory search principles to enhance multi-start approaches. The authors introduce a template of a constructive version of Tabu Search using both a set of elite solutions and 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. [51] 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 [17] 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 of the search space. That is, persistent attractiveness also carries with it the connotation of persistently unselected (i.e., not selected in many trials) within a specific domain or interval. The principle of *marginal conditional validity* specifies that the problem becomes more restricted as more and more decisions are made. 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 [17].

Scatter Search and Path-Relinking [23] are effective methodologies to solve a great diversity of optimization problems. These methods differ from other evolutionary procedures, such as genetic algorithms, in their approach to combine solutions based on path construction. (both in Euclidean spaces and in neighborhood spaces). In the context of Scatter Search, Laguna and Martí [34] discuss the development and application of the OptQuest system. Using this library, Ugray et al. [67] develop the algorithm called OQNLP to find global optimal for pure and mixed integer non-linear problems, where all the functions are differentiable with respect to continuous variables. It uses OptQuest to generate candidate starting points for a local NLP solver as a kind of multi-start algorithm. Additionally, the authors show in [66] that OQNLP is a promising approach to NLP smooth nonconvex problems with continuous variables. Later, Lasdon and Plummer [37] describe modifications to OptQuest/NLP and Multistart-NLP for global optimization, which allow them to find feasible solutions to a system of nonlinear constraints more efficiently. Modifications include the replacement of the penalty function used to measure the goodness of an initial point by the sum of infeasibilities and ending the search when a feasible solution is found.

Beausoleil at al. [3] 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). Through the Pareto relation, a subset of the best generated solutions is designated as 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 on a suite of test problems taken from the literature.

Considering the problem of finding global optima for restricted multimodal functions, Lasdon et al. [36] present some multi-start methods based on the adaptive memory programming (AMP) structure, which involves memory structures that can be superimposed to a local optimizer, to guide the search for initial points when solving global optimization problems. The first approach is based on a tabu tunneling strategy and the second one on a pseudo-cut strategy. Both are designed to avoid being trapped in local optima.

Since we cannot refer here to all the previous developments in this area, and we limit ourselves to a few significant examples. For instance, there is a recent application in the context of mobile network design [64]. The problem of assigning network elements to controllers when defining network structure can be modeled as a graph partitioning problem. Accordingly, a comprehensive analysis of a sophisticated graph partitioning algorithm for grouping base stations into packet control units for a mobile network is presented. The proposed algorithm combines multilevel and adaptive multi-start schemes to obtain high quality solutions efficiently. Performance assessment is carried out on a set of problem instances built from measurements in a live network. The overall results confirm that the proposed algorithm finds solutions better than those obtained by classical multi-level approaches and much faster than classical multistart approaches. The analysis shows that the best local minima share strong similarities, which explains the superiority of adaptive multi-start approaches

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 [11]. It was first used to solve set covering problems [10]. 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, which becomes the output of a complete GRASP iteration. Some examples of successful applications are given in [32, 54, 33]. Recently, Festa and Resende [12] present an overview of GRASP, describing its basic components and enhancements to the basic procedure, including reactive GRASP and intensification strategies.

Laguna and Martí [33] 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 [22]. This approach generates new solutions by exploring trajectories that connect high-quality solutions. It starts from one of these solutions and generates 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 originally linked by a sequence of moves (as in tabu search), they are then linked for the first time when this process is applied. 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, [4] proposed the Greedy Randomized Adaptive Path Relinking. Many different designs named *Evolutionary Path Relinking* have also been studied in [55].

Prais and Ribeiro [52] 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.

Morillo et al. [48] propose a new design of the GRASP for solving the latency-aware partitioning problem in Distributed Virtual Environments (DVE systems) called M-GRASP or GRASP with memory. The idea is to start from scratch and to design a specific GRASP that can be implemented in parallel and can provide a feasible solution for the considered problem at any iteration, in such a way that it can be adapted to any time constraint. Since each iteration in GRASP consists of a constructive phase and a local search phase, they propose different alternatives for each phase, evaluating the performance obtained with each alternative. Additionally, they enhance this basic approach with some intensification strategies, selecting the option with the best performance as the proposed final implementation.

Ribeiro and Resende [56] compare the run time distributions of GRASP with and without path-relinking implementations for four different applications: three-index assignment, maximum satisfiability, bandwidth packing, and quadratic assignment. In all cases the plots show that GRASP with path relinking performs better (founding target solutions faster) than the memoryless basic algorithm.

Glover [19] introduces a new design for a framework that links iterated neighborhood search methods and iterated constructive methods by exploiting the notions of conditional influence within a strategic oscillation framework. These approaches, which are unified within a class of methods called multi-wave algorithms, exploit memory-based strategies that draw on the concept of persistent attractiveness. These algorithms provide new forms of both neighborhood search methods and multi-start methods and are readily embodied within evolutionary algorithms and memetic algorithms by solution combination mechanisms derived from path relinking.

In 2007, Hirsch [27] proposed an adaptation of GRASP for continuous global optimization called continuous GRASP (C-GRASP), which was shown to perform well on a set of multimodal test functions, as well as on real-world applications. C-GRASP is a stochastic local search metaheuristic for finding cost-efficient solutions to continuous global optimization problems subject to box constraints. Like GRASP, C-GRASP is a multi-start procedure where a starting solution for local improvement is constructed in a greedy randomized fashion. In 2010 Hirsch et al. [26] described several improvements to speed up the original C-GRASP and make it more robust. The authors compare the new C-GRASP with the original version as well as with other algorithms from the recent literature on a set of benchmark multimodal test functions whose global minima are known. A sequential stopping rule is implemented and C-GRASP is shown to converge.

De Santis et al. [8] recently propose a variant of the GRASP framework that uses a nonmonotone strategy to explore the neighborhood of the current solution. Inspired by an idea proposed for Newton's method, this approach controls uphill moves without using a tabu list but rather by maintaining a number of previously computed objective function values. A new solution is accepted if its function value improves the worst value in the set. The authors formally state the convergence of the nonmonotone local search to a locally optimal solution and illustrate the effectiveness of the resulting Nonmonotone GRASP on three classical hard combinatorial optimization problems: the maximum cut problem (MAX-CUT), the weighted maximum satisfiability problem (MAX-SAT), and the quadratic assignment problem (QAP).

2.3 Constructive designs

Multi-start procedures usually follow a global scheme in which generation and improvement alternate for a certain number of iterations. Note that 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 [59, 7] for successful applications of this approach in vehicle routing).

Hickernell and Yuan [25] 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. Then, the sample is reduced by 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 [24] 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 on each point, thus producing a set of local minima; and (2) construct adaptive starting points derived from the best local minima 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 [65] describe a multi-start approach with a clustering strategy for constrained optimization problems. It exploits the characteristics of non-linear constrained global optimization problems by extending 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 basins 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 proposed in 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. [6] present a multi-start local search heuristic for a typical ship scheduling problem. Their method generates a large number of initial solutions with a randomized insertion heuristic. The best initial solutions are improved with a quick local search heuristic coupled with an extended version. The quick local search is used to improve a given number of the best initial solutions. The extended local search heuristic then further improves 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.

In 2013, Glover [18] introduces advanced greedy algorithms and applies them on knapsack and covering problems with linear and quadratic objective functions. Beginning with single-constraint problems, he provides extensions for multiple knapsack and covering problems, where the elements should be assigned to different knapsacks and covers. For multi-constraint knapsack and covering problems, the constraints are exploited using surrogate constraint strategies. Also, he introduces a progressive probe strategy for improving the selection of variables that should be assigned a value. The author describes ways to utilize these algorithms with multi-start and strategic oscillation metaheuristics. He also identifies how surrogate constraints can be employed to produce inequalities that dominate those previously used in the best linear programming methods for multi-constraint knapsack problems. These algorithms are often embedded within constructive processes used in multi-start metaheuristics and also within linked constructive and destructive processes in strategic oscillation metaheuristics.

Talarico et al. [63] develop and combine four constructive heuristics, as well as a local search composed of six operators to solve a variant of the capacitated vehicle routing problem. The initial solution obtained with one of the four construction heuristics serves as input for the local search. The construction heuristics and the local search are embedded in two different global metaheuristic structures: a multi-start and a perturb-and-improve (or perturbation) structure. The multi-start structure repeats both the construction phase and the local search phase a number of times. The perturbation structure only uses the construction heuristic once, and restarts the local search block from a perturbed solution. The resulting metaheuristics are able to obtain solutions of excellent quality in very limited computing times.

Luis et al. [41] investigate a multi-start constructive heuristic algorithm based on the furthest distance rule and a concept of restricted regions is developed to tackle a variant of the classical multi-source location-allocation problem in the presence of capacity restrictions. The classical problem assumes that the number of facilities is known in advance, whereas in practice, determining the number of facilities is a decision factor. This new approach determines the number of facilities minimizing the total sum of fixed and variable costs in accordance with finding the best trade-off between customer demand and opening of new facilities. The proposed method is assessed using benchmark data sets from the literature.

2.4 Hybrid designs

Ulder et al. [68] 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 [17] or Memetic Algorithms [49].

Mezmaz et al. [46] 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 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 must often be stopped before convergence. The full exploitation of the cooperation model needs a large amount of computational resources and the management of fault tolerance issues. 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 tested 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 about the design of a good search procedure is whether it is better to implement a simple improving method that allows a large 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 restarting strategies within the Tabu Search framework has been well documented in several papers (see for example [15, 22]). In [43] 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. The authors 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 gained in this work and in previous research 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.

Based on Iterated Local Search (ILS), Prins [53] proposes heuristics for the Vehicle Routing Problem: an ILS with several offspring solutions per generation called Evolutionary Local Search (ELS), and two hybrid forms of GRASP. These variants share three main features: a simple structure, a mechanism to alternate between solutions encoded as giant tours and VRP solutions, and a fast local search based on a sequential decomposition of moves. Using this idea, Lacomme et al. [31] address an extension of the Capacitated Vehicle Routing Problem where the demand of a customer consists of three-dimensional weighted items(3L-CVRP), and the objective is to design a set of trips for a homogeneous fleet of vehicles based at a depot node so as to minimize the total transportation cost. The items in each vehicle trip must satisfy the three-dimensional orthogonal packing constraints. The proposed method is a multi-start algorithm where ELS is applied to the initial solutions generated by the greedy randomized heuristics.

Kaucic [29] presents a multi-start Particle Swarm Optimization (PSO) algorithm for the global optimization of a function subject to bound constraints. The procedure consists of three main steps. In the initialization phase, an opposition-based learning strategy is performed. Then, a variant of an adaptive differential evolution scheme is used to adjust the velocity of the particles. Finally, a re-initialization strategy based on two swarm diversity measures is applied to avoid premature convergence and stagnation. The overall idea is to increase the search abilities of PSO by employing an opposition-based selection for the initial swarm and an adaptive velocity update equation for the following iterations. The restart scheme is applied to the particles in the swarm whenever premature convergence and stagnation occur.

Pacheco et al. [50] propose a heuristic method for solving a problem of sequencing jobs on a machine with programmed preventive maintenance and sequence-dependent set-up times. The method hybridizes multi-start strategies with Tabu Search. Their algorithm, called Multi-start Tabu (MST), is an iterative algorithm that generates a solution in each iteration using a constructive algorithm (called Diversification Generator), and then, improves it using a Tabu Search procedure (called Basic Tabu). In this way, each iteration produces a local optimum and the best one is the algorithms output.To explore the whole space of feasible solutions, the designed constructive procedure takes into account the knowledge accumulated during previous executions, generating solutions in regions not visited previously.

The research work of Sharma and Glemmestad [60] focuses on the use of the Generalized Reduced Gradient (GRG) method [66] to solve a constraint multivariable lift gas allocation optimization problem. The GRG algorithm is a local solver i.e. the solution provided by GRG may only be a local optimum. To ensure that the final solution is as close as possible to a global optimum, a multi-start search routine is applied on top of the GRG algorithm. First, different feasible starting points are generated. Then, GRG is applied to each of these feasible starting points and the corresponding local optima are stored. Finally, when all points have been exploited, the solution which maximizes the objective function is returned as the final solution.

2.5 Theoretical analysis

From a theoretical point of view, Hu et al. [28] 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. [5] 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 [24] apply this method for the iterative partitioning problem.

Moreno et al. [47] 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 [2] 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 [1, 70].

Many heuristics used for global optimization can be described as population-based algorithms in which, at every iteration, the quality of a population of solutions is evaluated and a new population is randomly generated according to a given rule, designed to achieve an acceptable trade-off in the allocation of computational effort for ”exploration” versus ”exploitation”. Wang and García [69] propose an algorithmic design for global optimization with multiple interacting threads. It applies a multi-start method that makes use of a local search algorithm to guarantee the diversity of search spaces. In the proposed design, each thread implements a search with a relative emphasis on exploitation that does not vary over time. More efficient exploration is achieved by means of a simple acceptance-rejection rule preventing duplication of the search spaces.

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 possible choices for each element are not restricted to its presence or absence, but rather represent a whole continuum between these two extremes. We can identify these extremes 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 [22], such memory provides a foundation for incentive-based learning, where actions leading to good solutions are reinforced through incentives or actions leading to bad solutions are discouraged through deterrents. 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 information is identified to exploit the search 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 is 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, 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 alternating in various ways between randomized and deterministic processes. The GRASP method discussed later combines several of these intermediate possibilities.

The **Degree of Rebuild** measures 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 found in previously generated solutions. Such an approach was proposed in the context of identifying and then iteratively exploiting strongly determined and consistent variables [15]. This selective way of fixing elements, by reference to their impact and frequency of occurrence in previously visited solutions, 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 [21] 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 incentives 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.

This classification has already been used in a practical approach to solve a vehicle routing problem proposed by an international company operating in Spain. The work reported in [39] considered a variant of the Open Vehicle Routing Problem in which the makespan, i.e., the time spent on the vehicle by one person, must be minimized. A competitive multi-start algorithm producing high quality solutions within reasonable computing time is proposed. The effectiveness of the algorithm is analyzed through computational testing on a set of 19 school-bus routing benchmark problems from the literature, and on 9 hard real-world problem instances.

The multi-start algorithm in [39] is a classical two-phases iterative process. First, there is a construction phase in which a feasible solution is generated, followed by a local search phase in which an attempt to improve solution quality and (possibly) infeasibility is performed. As a consequence, each iteration produces a locally optimal solution, and the algorithm returns the best solution found during the iterative process. According to our classification, the authors classify their method as Memory-less, Randomized, and Build-from-scratch because those characteristics favor solutione diversity, thus providing a best overall result.

4 The Maximum Diversity Problem

In this section we consider a difficult optimization problem to illustrate how to implement a multi-start method. In particular, we describe different solution methods for the Maximum Diversity Problem. This gives also us the opportunity to evaluate the use of memory structures in the context of multi-start methods.

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 [20]. The MDP has been studied by numerous authors, the most prominent among them being Kuo et al. [30], who described four formulations of the problem, ranging from the most intuitive to the most efficient. These formulations also served to show that the MDP is NP-hard. In 1996, Ghosh [14] proposed a multi-start method and proved the completeness of the problem. Later, Glover et al. [20] proposed four deterministic heuristic methods, two of them constructive and the other two destructive. Silva et al. [61] 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. In this context, the solution of the MDP consists of 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 an optimization problem in the following terms:

$$\begin{aligned} \max \quad & z = \sum_{i<j} d_{ij} x_i x_j \\ \text{subject to} \quad & \\ & \sum_{i=1}^{n} x_i = m \\ & x_i \in \{0,1\} \; i = 1, \ldots, n \end{aligned}$$

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

Two constructive algorithms are proposed to solve the MDP using a multi-start scheme, one 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 comes from a heuristic method proposed in Glover et al. [20]. 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 elements may 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 the elements in Sel is then computed as follows:

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

To select the next element for inclusion in the set Sel, an ordered list L is constructed with all the elements $s_i \in S - Sel$ within 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 among the elements in L, thus ensuring a minimum quality as defined by the percentage α. So, it is not a purely greedy selection, but it combiness greedyness with randomization. This procedure is repeated until m elements have been chosen ($|Sel| = m$). At this point, Sel contains a solution to the problem. After $niter$ executions, 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 [14], 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 [33], 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 d_{i^*} to the current solution is selected and exchanged with the first element $s_j \in S - Sel$ (in lexicographical order) that leads to an increase in the objective value. 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 element 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 [9]. The method uses memory both in the solution construction and improvement phases. These strategies are integrated within the Tabu Search method [22].

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 $freq[i]$ over 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 $quality[i]$ over 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 metric 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 penalty 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 that 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 such 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 addition 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 replaced 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 (i.e., 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 s_j 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 [9], we present a comparison with two other previously reported algorithms. Specifically, the MSWoM and MSWM algorithms are compared with the D2 constructive algorithm [20], along with the improvement method described in [14], and the KLDv2 algorithm with its improvement procedure [61]. They are 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 $[0,10]$. Each point has 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 of deviation for each procedure with respect to the best solution produced 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* instances

	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* instances

	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* instances

	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

We can conclude from these tables 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 also show that the use of memory, at least for the instances tested, leads to better results. Note that in the case of *Glover* instances, 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* instances, where substantial improvements are obtained with the multi-start methods.

A thorough computational study to compare 10 heuristics and 20 metaheuristics for the maximum diversity problem (MDP) can be found in [42]. The authors present the benchmark library MDPLIB which contains 315 instances of the problem, and compare the 30 methods on MDPLIB making use of non-parametric statistical tests to draw significant conclusions. They conclude that even the simplest heuristics provide good solutions to this problem. However, to obtain high-quality solutions they recommend to apply multi-start metaheuristics.

5 Conclusion

The objective of this chapter is to extend and advance the knowledge on multi-start methods. Unlike other well-known methods, these procedures have not yet become widely implemented and tested as true metaheuristic for solving complex optimization problems. We have presented new ideas that have recently emerged in the field of multi-start methods. These ideas, which have yet to be fully explored, have great potential. We have also shown the connections between these methodologies and other metaheuristics.

Our findings indicate that memory appears to play an important role during both the constructive and the improvement phase of a multi-start procedure. One possible explanation may be that the repeated application of the constructive phase operates primarily as a diversification process, while 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 confirm these previous findings. The comparison between memory-based and memory-less designs is an interesting area for future research.

Acknowledgements

This research was partially supported by the *Ministerio de Economía y Competitividad* with codes TIN2015-65460-C2 (MINECO-FEDER) and TIN2015-70226-R.

References

1. ACKLEY, D. H. An empirical study of bit vector function optimization. *Genetic algorithms and simulated annealing 1* (1987), 170–204.
2. BALUJA, S. An empirical comparison of 7 iterative evolutionary function optimization heuristics. school of computer science, 1995.
3. BEAUSOLEIL, R. P., BALDOQUIN, G., AND MONTEJO, R. A. Multi-start and path relinking methods to deal with multiobjective knapsack problems. *Annals of Operations Research 157*, 1 (2008), 105–133.
4. BINATO, S., FARIA JR, H., AND RESENDE, M. G. Greedy randomized adaptive path relinking. In *Proceedings of the IV Metaheuristics International Conference* (2001), Citeseer, pp. 393–397.
5. BOESE, K., KAHNG, A., AND MUDDU, S. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters 16*, 2 (1994), 101–113.
6. BRNMO, G., CHRISTIANSEN, M., FAGERHOLT, K., AND NYGREEN, B. A multi-start local search heuristic for ship scheduling-a computational study. *Computers and Operations Research 34*, 3 (2007), 900–917.
7. CHIANG, W.-C., AND RUSSELL, R. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research 63* (1996), 3–27.

8. De Santis, M., Festa, P., Liuzzi, G., Lucidi, S., and Rinaldi, F. A nonmonotone grasp. *Mathematical Programming Computation 8*, 3 (2016), 271–309.
9. Duarte, A., and Martí, R. Tabu search and grasp for the maximum diversity problem. *European Journal of Operational Research 178*, 1 (2007), 71–84.
10. Feo, T., and Resende, M. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters 8*, 2 (1989), 67–71.
11. Feo, T. A., and Resende, M. G. Greedy randomized adaptive search procedures. *Journal of global optimization 6*, 2 (1995), 109–133.
12. Festa, P., and Resende, M. G. C. Grasp: basic components and enhancements. *Telecommunication Systems 46*, 3 (2011), 253–271.
13. Fleurent, C., and Glover, F. Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. *INFORMS Journal on Computing 11*, 2 (1999), 198–204.
14. Ghosh, J. Computational aspects of the maximum diversity problem. *Operations Research Letters 19*, 4 (1996), 175–181.
15. Glover, F. Heuristics for integer programming using surrogate constraints. *Decision Sciences 8*, 1 (1977), 156–166.
16. Glover, F. Tabu search. *ORSA Journal on Computing 1*, 3 (1989), 190–206.
17. Glover, F. *Multi-Start and Strategic Oscillation Methods — Principles to Exploit Adaptive Memory*. Springer US, Boston, MA, 2000, pp. 1–23.
18. Glover, F. Advanced greedy algorithms and surrogate constraint methods for linear and quadratic knapsack and covering problems. *European Journal of Operational Research 230*, 2 (2013), 212 – 225.
19. Glover, F. Multi-wave algorithms for metaheuristic optimization. *Journal of Heuristics 22*, 3 (2016), 331–358.
20. Glover, F., Kuo, C.-C., and Dhir, K. S. Heuristic algorithms for the maximum diversity problem. *Journal of information and Optimization Sciences 19*, 1 (1998), 109–132.
21. Glover, F., and Laguna, M. Tabu search, modern heuristic techniques for combinatorial problems, edited by cr reeves, 1993.
22. Glover, F., and Laguna, M. *Tabu Search*. Kluwer, Boston, MA, 1997.
23. Glover, F., Laguna, M., and Martí, R. Fundamentals of scatter search and path relinking. *Control and Cybernetics Vol. 29, no 3* (2000), 653–684.
24. Hagen, L. W., and Kahng, A. B. Combining problem reduction and adaptive multistart: A new technique for superior iterative partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 16*, 7 (1997), 709–717.
25. Hickernell, F. J., and Yuany, Y.-x. A simple multistart algorithm for global optimization. *OR Transactions 1*, 2 (1997), 2.
26. Hirsch, M., Pardalos, P., and Resende, M. Speeding up continuous {GRASP}. *"European Journal of Operational Research 205*, 3 (2010), 507 – 521.
27. Hirsch, M. J., Meneses, C. N., Pardalos, P. M., Ragle, M., and Resende, M. G. C. A continuous grasp to determine the relationship between drugs and adverse reactions. *AIP Conference Proceedings 953*, 1 (2007), 106–121.
28. Hu, X., Shonkwiler, R., and Spruill, M. C. Random restarts in global optimization.
29. Kaucic, M. A multi-start opposition-based particle swarm optimization algorithm with adaptive velocity for bound constrained global optimization. *Journal of Global Optimization 55*, 1 (2013), 165188.
30. Kuo, C.-C., Glover, F., and Dhir, K. S. Analyzing and modeling the maximum diversity problem by zero-one programming. *Decision Sciences 24*, 6 (1993), 1171–1185.
31. Lacomme, P., Toussaint, H., and Duhamel, C. A GRASP x els for the vehicle routing problem with basic three-dimensional loading constraints. *Engineering Applications of Artificial Intelligence 26*, 8 (2013), 1795 – 1810.
32. Laguna, M., Feo, T. A., and Elrod, H. C. A greedy randomized adaptive search procedure for the two-partition problem. *Operations Research 42*, 4 (1994), 677–687.
33. Laguna, M., and Martí, R. Grasp and path relinking for 2-layer straight line crossing minimization. *INFORMS Journal on Computing 11* (1999), 44–52.

34. LAGUNA, M., AND MARTÍ, R. *The OptQuest Callable Library*. Springer US, Boston, MA, 2002, pp. 193–218.
35. LAGUNA, M., AND MARTÍ, R. *Scatter search: methodology and implementations in C*, vol. 24. Springer Science & Business Media, 2012.
36. LASDON, L., DUARTE, A., GLOVER, F., LAGUNA, M., AND MARTÍ, R. Adaptive memory programming for constrained global optimization. *Computers & Operations Research 37*, 8 (2010), 1500 – 1509. Operations Research and Data Mining in Biological Systems.
37. LASDON, L., AND PLUMMER, J. C. Multistart algorithms for seeking feasibility. *Computers & Operations Research 35*, 5 (2008), 1379 – 1393. Part Special Issue: Algorithms and Computational Methods in Feasibility and Infeasibility.
38. LØKKETANGEN, A., AND GLOVER, F. Probabilistic move selection in tabu search for zero-one mixed integer programming problems. In *Meta-Heuristics*. Springer, 1996, pp. 467–487.
39. LÓPEZ-SÁNCHEZ, A., HERNÁNDEZ-DÍAZ, A., VIGO, D., CABALLERO, R., AND MOLINA, J. A multi-start algorithm for a balanced real-world open vehicle routing problem. *European Journal of Operational Research 238* (2014), 104–113.
40. LOZANO, M., GLOVER, F., GARCÍA-MARTÍNEZ, C., RODRÍGUEZ, F. J., AND MARTÍ, R. Tabu search with strategic oscillation for the quadratic minimum spanning tree. *IIE Transactions 46*, 4 (2014), 414–428.
41. LUIS, M., LAMSALI, H., IMRAN, A., AND LIN, A. A multi-start heuristic for the capacitated planar location-allocation problem with facility fixed costs. *International Information Institute (Tokyo). Information 19*, 7A (2016), 2441.
42. MARTÍ, R., GALLEGO, M., DUARTE, A., AND PARDO, E. Heuristics and metaheuristics for the maximum diversity problem. *Journal of Heuristics 19* (2013), 591–615.
43. MARTÍ, R., LAGUNA, M., GLOVER, F., AND CAMPOS, V. Reducing the bandwidth of a sparse matrix with tabu search. *European Journal of Operational Research 135*, 2 (2001), 450–459.
44. MARTÍ, R., RESENDE, M., AND RIBEIRO, C. Multi-start methods for combinatorial optimization. *European Journal of Operational Research (Invited Review) 226* (2013), 1–8.
45. MAYNE, D., AND MEEWELLA, C. A non-clustering multistart algorithm for global optimization. In *Analysis and optimization of systems*. Springer, 1988, pp. 334–345.
46. MEZMAZ, M., MELAB, N., AND TALBI, E.-G. Using the multi-start and island models for parallel multi-objective optimization on the computational grid.
47. MORENO, J., MLADENOVIC, N., AND MORENO-VEGA, J. 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, Germany* (1995).
48. MORILLO, P., ORDUNA, J. M., AND DUATO, J. M-grasp: A grasp with memory for latency-aware partitioning methods in dve systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 39*, 6 (2009), 1214–1223.
49. MOSCATO, P. Memetic algorithms: A short introduction. In *New ideas in optimization* (1999), McGraw-Hill Ltd., UK, pp. 219–234.
50. PACHECO, J., ÁNGEL-BELLO, F., AND ÁLVAREZ, A. A multi-start tabu search method for a single-machine scheduling problem with periodic maintenance and sequence-dependent set-up times. *Journal of Scheduling 16*, 6 (2013), 661673.
51. PATTERSON, R., PIRKUL, H., AND ROLLAND, E. A memory adaptive reasoning technique for solving the capacitated minimum spanning tree problem. *Journal of Heuristics 5*, 2 (1999), 159–180.
52. PRAIS, M., AND RIBEIRO, C. Reactive grasp: An application to a matrix decomposition problem in tdma traffic assignment. *INFORMS Journal on Computing 12*, 3 (2000), 164–176.
53. PRINS, C. *A GRASP Evolutionary Local Search Hybrid for the Vehicle Routing Problem*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 35–53.
54. RESENDE, M. G. Computing approximate solutions of the maximum covering problem with grasp. *Journal of Heuristics 4*, 2 (1998), 161–177.
55. RESENDE, M. G., MARTÍ, R., GALLEGO, M., AND DUARTE, A. Grasp and path relinking for the max–min diversity problem. *Computers & Operations Research 37*, 3 (2010), 498–508.

56. Ribeiro, C. C., and Resende, M. G. C. Path-relinking intensification methods for stochastic local search algorithms. *Journal of Heuristics 18*, 2 (2012), 193–214.
57. Rinnooy Kan, A., and Timmer, G. *Global optimization*, vol. 1. 1989, p. 631662.
58. Rochat, Y., and Taillard, E. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics 1*, 1 (1995), 147–167.
59. Russell, R. A. Hybrid heuristics for the vehicle routing problem with time windows. *Transportation Science 29*, 2 (1995), 156–166.
60. Sharma, R., and Glemmestad, B. On generalized reduced gradient method with multi-start and self-optimizing control structure for gas lift allocation optimization. *Journal of Process Control 23*, 8 (2013), 1129 – 1140.
61. Silva, G., Ochi, L., and Martins, S. Experimental comparison of greedy randomized adaptive search procedures for the maximum diversity problem. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 3059* (2004), 498–512.
62. Solis, F. J., and Wets, R. J. Minimization by random search techniques. *Mathematics of Operations Research 6*, 1 (1981), 19–30.
63. Talarico, L., Sörensen, K., and Springael, J. Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem. *European Journal of Operational Research 244*, 2 (2015), 457 – 470.
64. Toril, M., Wille, V., Molina-Fernández, I., and Walshaw, C. An adaptive multi-start graph partitioning algorithm for structuring cellular networks. *Journal of Heuristics 17* (2011), 615–635.
65. Tu, W., and Mayne, R. An approach to multi-start clustering for global optimization with non-linear constraints. *International Journal for Numerical Methods in Engineering 53*, 9 (2002), 2253–2269.
66. Ugray, Z., Lasdon, L., Plummer, J., Glover, F., Kelly, J., and Martí, R. Scatter search and local nlp solvers: A multistart framework for global optimization. *INFORMS Journal on Computing 19*, 3 (2007), 328–340.
67. Ugray, Z., Lasdon, L., Plummer, J. C., Glover, F., Kelly, J., and Martí, R. *A Multistart Scatter Search Heuristic for Smooth NLP and MINLP Problems*. Springer US, Boston, MA, 2005, pp. 25–57.
68. Ulder, N. L., Aarts, E. H., Bandelt, H.-J., Van Laarhoven, P. J., and Pesch, E. Genetic local search algorithms for the traveling salesman problem. In *International Conference on Parallel Problem Solving from Nature* (1990), Springer, pp. 109–116.
69. Wang, Y., and García, A. Interactive model-based search for global optimization. *Journal of Global Optimization 61*, 3 (2015), 479–495.
70. Wattenberg, M., and Juels, A. Stochastic hillclimbing as a baseline method for. Tech. rep., Berkeley, CA, USA, 1994.