MIRHA: Multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problem

Juan, A. A., Faulin, J., Ferrer, A., **Lourenço, H.R.**, and Barros, B. (2013), MIRHA: Multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problem. TOP: Volume 21, Issue 1, Pages 109-132.

DOI 10.1007/s11750-011-0245-1.
ISSN: 1134-5764 (print version) ISSN: 1863-8279 (electronic version).
Link to publication

MIRHA: multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems

Angel A. Juan, Javier Faulin, Albert Ferrer, Helena R. Lourenço and Barry Barrios

Abstract This paper discusses the use of probabilistic or randomized algorithms for solving vehicle routing problems with non-smooth objective functions. Our approach employs non-uniform probability distributions to add a biased random behavior to the well-known savings heuristic. By doing so, a large set of alternative good solutions can be quickly obtained in a natural way and without complex configuration processes. Since the solution-generation process is based on the criterion of maximizing the savings, it does not need to assume any particular property of the objective function. Therefore, the procedure can be especially useful in problems where properties such as non-smoothness or non-convexity lead to a highly irregular solution space, for which the traditional optimization methods -both of exact and approximate nature- may fail to reach their full potential. The results obtained so far are promising enough to suggest that the idea of using biased probability distributions to randomize classical heuristics is a powerful one that can be successfully applied in a variety of cases.

Keywords: Randomized algorithms, combinatorial optimization, vehicle routing problem, biased random search, heuristics.

Mathematical Subject Classification (2000): 65K05, 90C26, 90C27, 90C59.

Angel A. Juan, Computer Science Dept., Open University of Catalonia, Spain.
ajuanp@uoc.edu

Javier Faulin, Statistics and Operations Research Dept., Public University of Navarre.
javier.faulin@unavarra.es

Albert Ferrer, Applied Mathematics 1 Dept., Technical University of Catalonia, Spain.
alberto.ferrer@upc.edu

Helena R. Lourenço Economics and Business Dept., Universitat Pompeu Fabra, Spain.
helena.ramalhinho@upf.edu

Barry Barrios, Industrial Engineering and Management Science Dept.,Northwestern University, USA.
barrybarrios2015@u.northwestern.edu

1 Introduction

Combinatorial optimization problems have posed numerous challenges to the human mind throughout the past few centuries. From a theoretical perspective, they have a well-structured definition consisting of an objective function that needs to be minimized and/or maximized and a series of constraints. In addition to having useful applications arising from abstract study, the primary reason for which they have been so actively investigated is the tremendous amount of real-life applications that can be successfully modeled in this way. For example, areas like routing or scheduling contain plentiful hard challenges that can be expressed as a combinatorial optimization problem (see [65]).

A considerable number of methods and techniques that search the solution space and try to find the optimum have been developed. In a few cases, the solution space can easily be explored due to certain properties of the functions involved, such as convexity. For those instances, the problem can often be solved efficiently and exactly. However, in other circumstances, the solution space is highly irregular and finding the optimum is in general impossible -at least if it has to be done in a reasonable amount of time. An exhaustive method that checks every single point in the solution space would be of very little help in these difficult cases, since it would take exponential time. Also, some approaches are fairly complex while others need to take into account the particular features of the problem at hand. Therefore, designing such approaches usually takes a substantial amount of time and the methodology has a limited application range. In fact, every method has its drawbacks. We believe that accuracy (quality of results), simplicity of design and implementation (including the avoidance of complex and time consuming fine-tuning processes), efficiency (relation between computational time employed and quality of results), robustness (regarding changes in the inputs and constraints), and flexibility (the ability to deal with general combinatorial optimization problems under different scenarios) are the attributes that can make one approach better or more suitable than another (see [16]).

The main idea of this paper is that combining a heuristic biased randomization with an adapted (problem-specific) local search can be a natural and efficient way to deal with realistic vehicle routing problems under more complex scenarios dominated by non-smooth/non-convex objective functions and non-convex regions. Thus, we propose a method named MIRHA (Multi-start bIased Randomization of classical Heuristics with Adaptive local search), which pertains to the class of nondeterministic or stochastic methods and relies on biased (non-uniform and non-symmetric) random sampling. Therefore, on different runs of the algorithm we get different good solutions that depend on which points are randomly sampled. Having a pool of solutions to choose from can be especially useful in real-life problems, when the best-known solution may be unfeasible due to external constraints. While optimization of non-smooth/non-convex functions is an important issue that has been already discussed in other combinatorial optimization problems, in the context of vehicle routing problems there is very few literature regarding this matter. Therefore, one of the main contributions of this paper is to start covering this lack of discussion with a simple -almost parameter free- and efficient methodology, which can provide pseudo-optimal solutions to difficult problems in reasonable computing times.

This article is structured as follows: Section 2 reviews some basic concepts related to non-convex and non-smooth optimization problems. Section 3 provides a literature review on the use of probabilistic algorithms in non-smooth/non-convex combinatorial optimization. Section 4 offers an overview of the vehicle routing problem and discusses why it might be important to consider non-smooth/non-convex objective functions in some of their practical applications. Section 5 presents a mathematical model for the problem considered in this paper. Section 6 describes the algorithm proposed in detail. Section 7 includes some numerical experiments regarding the solving of non-smooth vehicle routing problems. Finally, the concluding section summarizes the main contributions of this work.

2 Basic concepts on non-convex and non-smooth problems

Optimization problems can be categorized, from a high-level perspective, as either convex or non-convex. In general, convex optimization problems (COPs) have two parts: a series of constraints that represent convex regions, and an objective function to be minimized that is also convex. The dual problem, in which the objective function is concave and the goal is to maximize it, is also often encountered and for the purpose of this paper it will still be considered a member of the convex-like class of problems. COPs are worth studying because they have a wide variety of applications and many problems can be reduced to them via a change of variables. Linear Programming is one well-known example, since linear functions are trivially convex. Other examples of COPs include Quadratic Programming, Geometric Programming, Conic Optimization, Least Squares, etc (see [5]). The main idea, in convex optimization problems, is that every constraint restricts the space of solutions to a certain convex region. By taking the intersection of all these regions we obtain the set of feasible solutions, which is also convex. Due to the nice structure of the solution space, every single local optimum is a global optimum too. This is the key property that permits us to solve COPs exactly and efficiently up to very large sizes. Several algorithms, such as the Interior Point Method (see [90]), have been developed to find the optimal solution. However, almost none of them can be easily extended to the non-convex case.

In non-convex optimization problems (NCOPs) the objective function or even the feasible region are not convex, which results in a far more complex solution space than in the case of COPs. Now we may have many disjoint regions, and multiple locally optimal points within each of them (Figure 1 left). As a result, if a traditional local search is applied, there is a high risk of ending in the vicinity of a local optimum that may still be far away from the global optimum. Another drawback is that it can take exponential time in the size of the input to determine that a NCOP is infeasible, that the objective function is unbounded, or that one of the solutions found so far is the actual global optimum.

A function is smooth if it is differentiable and its derivative function is continuous. Therefore, a non-smooth function is one that is missing some of these properties. Non-smooth optimization problems (NSPs) are similar to NCOPs in the sense that they are much more difficult to solve than traditional smooth and convex problems. The function for which a global optimum needs to be computed is now non-smooth and the solution space might contain again multiple disjoint regions and many locally optimal points within each of them (Figure 1 right). The lack of a nice structure makes the application of traditional mathematical tools, such as gradient information, very complicated or even impossible in these cases. The computational techniques that can be used to solve this type of problems are often fairly complex and depend on the particular structure of the problem. As a result, developing such techniques is in general time-consuming, and the resulting application range is very limited. However, most real-life objective functions are either non-convex, non-smooth or both. Therefore, combinatorial optimization under these complex but common circumstances is an important field to explore that, as far as we know, has not been covered enough in the current literature.

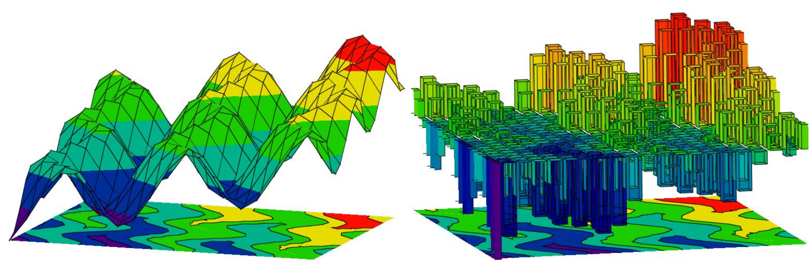

Fig. 1 (left) a non-convex objective function; (right) a non-smooth objective function

3 Use of probabilistic algorithms in non-smooth/non-convex combinatorial optimization

In the context of combinatorial optimization, probabilistic or randomized algorithms make use of pseudo-random numbers or variants during the constructive or local-search processes. In addition to the problem's input data, a probabilistic algorithm use random bits to do random choices during the execution of the algorithm. An important property is that for the same input the algorithm can produce different outputs in different runs. Within these algorithms we can include, among others, the Genetic and Evolutionary Algorithms ([35], [46], [19], [20], [79]), Simulated Annealing ([52], [84], [69]), GRASP ([31], [29], [81]), Variable Neighborhood Search ([43]), Iterated Local Search ([60], [61]), Ant Colony Optimization ([23]), Probabilistic Tabu Search ([59], [34]), or Particle Swarm Optimization ([51]), which is similar to Evolutionary Algorithms but inspired by social behavior of bird flocking or fish schooling. For a detailed review of Randomized Algorithms the reader is referred to Collet and Rennard ([15]).

Probabilistic algorithms have been widely used to solve many combinatorial optimization problems such as, for example: Sequencing and Scheduling Problems ([36], [88], [73]), Vehicle Routing Problems ([55]), Quadratic and Assignment Problems ([58], [70], [34]), Location and Layout Problems ([24], [64]), Covering, Clustering, Packing and Partitions Problems ([11], [66]). They have also been used to solve real combinatorial optimization problems that arise in different industrial sectors, e.g.: Transportation, Logistics, Manufacturing, Aeronautics, Telecommunication, Health, Electrical Power Systems, Biotechnology, etc.

Despite the great success of the application of these methods to the aforementioned combinatorial optimization problems, there exist only a few documented applications of these algorithms to NCOPS or NSPs. In a search done in the ISI Web of Knowledge, we only found a few references regarding the use of probabilistic algorithms to optimize non-convex or non-smooth objective functions. These papers are reviewed next, since they are -at least to some extent- related to our research. The main goals of this literature review are twofold: (a) to identify which randomized algorithms have been applied to NCOPs and NSPs, and (b) to identify the specific combinatorial optimization problems where these algorithms have been successfully applied.

Bagirov and Yearwood ([2]) present a formulation of the Minimum Sum-of-Squares Clustering problem, which is a non-smooth, non-convex optimization problem. The goal of clustering problems is to separate a large set of objects into groups or clusters based on certain similarity criteria. There are many possible applications, especially in fields such as engineering, information and decision sciences or earth sciences. Information retrieval or image segmentation are some particular cases in which clustering techniques can be applied. They also emphasize that a large number of approaches, such as dynamic programming, branch and bound, or K-means algorithms have been used for the clustering problem. However, the authors point out that most of these methods are efficient only in certain special settings. For example, a dynamic programming approach only works when the number of objects is small, so it cannot be used for most real-world problems. Branch and bound methods are effective only if the number of clusters is not too large. The efficiency of K-means algorithms also decreases as the number of clusters is increased. Moreover, alternative techniques such as bilinear programming fail as well when they face non-convex and non-smooth objective functions that have many local optimums. The authors remark that, in general, better results are generated when metaheuristics are used for the clustering problem. A Tabu Search approach that outperforms the K-means has been proposed by Al-Sultan ([1]). However, the algorithm requires three parameters, so an extensive parameter study was necessary to find the best parameter settings. Similarly, Selim and Al-Sultan ([83]) propose a Simulated Annealing approach. However, it also requires rather complex fine-tuning processes, which is a common drawback of most state-of-the-art metaheuristics.

The issue of Optimal Routing in Communication Networks has also received a lot of attention from researchers. The objective here is to find the best path for data transmission in a short amount of time. The routing strategy can greatly affect system performance, so there is a high demand for efficient algorithms. As a result, numerous methods that deal with this challenge have been designed. A method that combines Genetic Algorithms with Hopfield networks is proposed by Hamdan and El-Hawary ([42]). Finally, an approach based on Tabu Search is taken by Oonsivilai et al. ([71]). The main drawbacks for most of these methods are either their inability to efficiently explore the solution space or very long computation times.

Bagirov et al. ([3]) present a non-smooth formulation for the Location Problem in Wireless Sensor Networks. In general, a wireless sensor network can be defined as a distributed collection of nodes (sensors) that have limited resources and operate autonomously. The goal is to find or accurately estimate the position of the nodes. This task could be easily accomplished using a Global Positioning System (GPS) for every node, but the costs would be prohibitive. In addition to this, a lot of research has been done to study both indoor and outdoor localization systems. However, most approaches have assumed accurate range measurements, which is unrealistic for RF (Radio Frequency) signal strength measurements. Ramadurai and Sichitiu ([78]) show that a probabilistic approach can be adopted to deal with range measurements inaccuracy. Their algorithm is“RF” based, robust to range measurement inaccuracies and can be tailored to varying environmental conditions”. GRASP algorithms ([30]) have been applied to solve the Vehicle Routing Problem (VRP). Most applications focus on the basic VRP variants, with some exceptions, which include additional constraints, e.g.: VRP with time-windows or backhauls ([53], [9], [10]). Other metaheuristics have also been applied to the VRP with realistic constraints. For instance, Hashimoto et al. ([44]) describe a Vehicle Routing Problem with soft constraints and propose an Iterated Local Search algorithm to solve it. They consider soft time windows and soft traveling-time constraints. It makes sense to assume that, for most real-world applications, soft constraints are more likely to occur than hard ones, i.e., it might be possible to violate them to some degree, although constraint violation might imply a specific penalty cost. The same idea can be applied to traveling time, e.g., sometimes traveling times can be shortened if paying for a turnpike toll. The relaxation of these constraints might naturally lead to a problem formulation, which includes a more realistic, non-convex or non-smooth, objective function. The authors propose an Iterated Local Search based on a cross exchange, 2-opt and Or-opt neighborhoods that also include a dynamic programming component to compute the optimal service start times. Other extensions of the VRP consist of including time windows, delivery and pickup operations, split deliveries, etc. Some of these problems can be formulated as NCOPs or NSPs ([21], [22]). In particular, Derigs et al. ([21]) present a local search method to solve the split delivery vehicle routing problem.

Potvin ([76]) presents a review of the application of Evolutionary Algorithms to the VRP. According to this work, Evolutionary Algorithms have been applied in many different ways and mostly to analyze the Capacitated VRP or the VRP with time-windows. The Quadratic Assignment Problem (QAP) is one of the most interesting and challenging combinatorial optimization problems in existence. Even though there are convex formulations for this problem ([8]), it is still a very difficult problem to solve. Fleurent and Glover ([34]) propose an improved Constructive Multi-start Algorithm to solve this problem. The method is an extension of the GRASP that includes adaptive memory search principles. These simple principles consist of a biased selection based on the adaptive probabilities of the new element to be included in the partial solution during the constructive method. To obtain good results they also incorporate several other principles as intensification, candidate list strategies and the Proximate Optimality Principles (POPs) that require the fine-tuning of several parameters. The authors conclude that the proposed strategies obtain excellent results for most existing problems. Particle Swarm Optimization (PSO) has been applied frequently to NCOPs and NSPs problems. Hemamalini and Simon ([45]) present an Artificial Bee Colony Algorithm for the Economic Load Dispatch Problem with non-smooth cost functions. The Economic Dispatch Problem is one of the most important problems to be solved in the operation and planning of power systems. The Artificial Bee Colony (ABC) algorithm proposed is a similar but alternative approach to the use of PSO and Differential Evolution (DE) algorithms. ABC uses only common control parameters, such as colony size and maximum cycle number. Another application of Particle Swarm Optimization to the economic dispatch problem is done by Neyestani et al. ([68]). Yang and Chou ([91]) describe the application of Particle Swarm optimization to a special manpower assignment problem in a consulting engineering firm, which is multi-objective, nonlinear, non-smooth and combinatorial. Still within the operation of power systems, Vaisakh and Srinivas ([89]) present a method based on Ant Colony Optimization and an extension of Evolutionary Algorithms to solve an Optimal Power Flow Problem with non-smooth cost functions. General methods based on probabilistic algorithms for non-smooth or non-convex problems are less frequent in the literature. Bresina ([6]) proposes a methodology called Heuristic-Biased Stochastic Sampling that is based on a biased iterative sampling of the search tree according to some heuristic's criteria. The author applies the method to an observation scheduling problem, and concludes that this approach outperforms greedy search within a small number of samples. Schlüter et al. ([82]) present an extended Ant Colony Optimization for non-convex mixed integer nonlinear programming. The results obtained on MINLP benchmark problems and on one engineering design problem proved to be competitive with other approaches based on local search. Toscano and Lyonnet ([86]) propose a new heuristic to solve non-convex optimization problems, designated as Heuristic Kalman algorithm. This method falls in the category of population-based stochastic methods and considers the optimization problem as a measurement process designed to give an estimate of the optimum.

4 A basic description of the Capacitated Vehicle Routing Problem

The Vehicle Routing Problem constitutes a well-known family of combinatorial optimization problems, which have many applications in real scenarios: goods delivery optimization, logistic design of manufacturing processes, design of computer components, garbage collection, etc. The most popular VRP is called the Capacitated Vehicle Routing Problem (CVRP) which constraints the routing activities by the capacity of the distribution vehicles. The CVRP introduces a set of customer demands, which have to be served with a fleet of vehicles from a depot or central node. Each vehicle has the same capacity (homogeneous fleet) and each customer has a certain demand that must be satisfied. Additionally, there is a cost matrix that measures the costs associated with moving a vehicle from one node to another. These costs usually represent distances, traveling times, number of vehicles employed or a combination of these factors.

The CVRP is a very natural interpretation of routing problems because the capacity constraint is very common in practical cases. As a result, having a good knowledge of the CVRP facilitates the understanding of some other VRP variants, which evolve from the CVRP.

The VRP is NP-Hard, which implies a non-polynomial increase of the space of solutions size when increasing the input size. Although this problem has already been studied for decades, it is still attracting a great amount of attention from researchers worldwide due to its potential applications ([41]). In fact, different approaches to the VRP have been explored during the last decades. These approaches range from the use of exact optimization methods, such as linear programming, for solving small- to medium-size problems with relatively simple constraints, to the use of heuristics and metaheuristics that provide quasi-optimal solutions for medium and large-size problems with more complex constraints ([17]). Nevertheless, most of the methods cited before focus on the minimization of an a priori cost function -which usually models tangible costs- subject to a set of well-defined constraints. However, real-life problems are much more complex. These include intangible costs, non-smooth functions, fuzzy constraints and desirable solution properties that are difficult to model ([75], [50]). In other words, it is not always straightforward to build a model, which takes into account all possible costs (e.g., environmental costs, work risks, etc.), constraints and desirable solution properties (e.g., time or geographical restrictions, balanced work load among routes, solution attractiveness, etc.). All in all, as some researchers have pointed out already, there is a need for simpler and more flexible methods. These methods can be used to handle the numerous side constraints that arise in practice ([55]).

Concerning the analysis of methods to solve primitive VRPs, we can mention the first heuristic algorithms especially designed for that purpose: the Clarke and Wright's savings algorithm ([14]) and the Gillet and Miller's procedure ([39]). Both methods combined simplicity and elegance in the resolution of real routing problems of small dimension. The following generation of VRP heuristics was based on double phase methods with one of them being an exact procedure. The most representative algorithm of this family is the Fisher and Jaikumar's method ([33]). During the nineties, metaheuristics were born with a long list of new methods based on different methodologies: Iterative Search ([33]), Tabu Search ([37]), GRASP ([28]), Genetic Algorithms ([40]), Neural Networks ([47]) or Ant Algorithms ([7]). With the beginning of the new century, those methods developed and mixed producing hybrid procedures ([29]) or new strategies based on similar ideas ([85], [74]). It is interesting to highlight that the use of simulation in the development of routing algorithms is a suitable methodology to tackle routing problems with complex formulations ([27], [26], [48]). Similarly, Evolutionary Algorithms proved to be a reliable way to generate solutions to intricate VRPs ([63]).

A plethora of approaches for different VRPs have been explored during the last decades ([87], [41]). Some of these approaches are based on traditional cost functions (distances, drivers' salaries, traveling time, etc), and they range from the use of exact optimization methods -such as linear programming- for solving small-size problems with relatively simple constraints to the use of heuristics and metaheuristics that provide near-optimal solutions for medium and large-size problems with more complex constraints. Most of these methods focus on minimizing an a priori cost function subject to a set of well-defined traditional constraints. However, real-life problems tend to be complex enough so that not all possible costs -e.g., environmental costs, work risks, etc.-, constraints and desirable solution properties -e.g., time or geographical restrictions, balanced work load among routes, solution attractiveness, etc.-, can be considered a priori during the mathematical modeling phase ([75], [50]). For that reason, there is a need for more flexible methods that are able to provide a large set of alternative near-optimal solutions with different properties, so that the decision-makers can choose among different alternative solutions according to their specific requirements and preferences. Furthermore, it is also necessary to describe a new type of costs, which usually are non-convex or non-smooth because they are based neither on linear (or quasi-linear) structures nor functions with derivatives. This kind of non-smooth analysis is already common in other engineering fields ([92], [72], [68]) but they have been seldom considered in the VRP literature. Therefore, one of the main contributions of this paper is to discuss the incorporation of these complex but more real-life objective functions in the VRP arena. As examples of how non-smooth and non-convex functions could naturally appear in VRPs, we can consider the following situations:

i) Minimization of fuel consumptions in surface transportation. Road transportation is the predominant way of transporting goods in Europe and also in other parts of the world. Direct costs associated with this type of transportation have experienced a significant increase since 2000, and more so during these last years due to the rise of oil prices. These costs are not usually easy to describe mathematically because of their continuous change (roads slopes or types of asphalts) or direct dependence to natural conditions (weather or temperature). A similar analysis of fuel consumption control by minimizing speed in ships crossing the Atlantic ocean has been done by Fagerholt et al. ([25]).

ii) Minimization of CO_2 emissions related to road transportation. Furthermore, road transportation is intrinsically associated with a great deal of indirect or external costs, which are usually easily observable -congestion, contamination, security- and safety-related costs, mobility, delay time costs, etc.- but are usually left unaccounted for because of the difficulty to quantify them ([54]). In addition to these easily observable costs, many others, like environmental costs due to the production and use of fossil fuel might be considered. In this scenario, the need for developing new methods to estimate new cost functions related to noise and environmental issues so that optimal (or quasi-optimal) strategies can be chosen in road transportation becomes evident. These estimations can be non-linear and even non-smooth/non-convex, and have been described by different authors ([4], [32]).

To sum up, our point is that non-smooth/non-convex objective functions might also play an important role in the VRP research area, especially when there is a necessity of modeling realistic costs functions or decision-makers' utility functions. These kind of functions have already been discussed in other related research fields, e.g. routing problems in Electrical Engineering ([38]) and other combinatorial optimization problems not directly linked to routing ([12],[13]). In fact, even when we have not been able to find VRP references including the terms non-smooth/non-convex in their titles, it is possible to find works describing vehicle routing problems that make use of non-smooth/non-convex objective functions, e.g., Lourenço and Ribeiro ([62]), even when the authors do not make a direct reference to this fact. In all cases, these non-smooth/non-convex problems are always connected with real distribution or transportation cases.

5 Mathematical model for the CVRP problem with a non-smooth objective function

We assume a complete directed graph $G = (V,\ E)$ with a set of $n+1$ nodes, $V = \{0,\ 1,\ 2,\dots,n\}$, and a set of arcs, $E = \{(i,\ j)|\ i,\ j \in V,\ i \neq j\}$. The vertex 0 is named the depot node and the other vertices represent customers. Each customer and each arc are associated with:

a) a fixed quantity, $q_j \geq 0$, of goods to be delivered to customer j,
b) a cost, $d_{i,j} \geq 0$, (distance-based) of traveling from node i to node j,
c) a cost, $s_j \geq 0$, (time-based) of serving the customer j.

Therefore, given an arc $(i,\ j)$, the cost $c_{i,j}$ associated with this arc is given by $c_{i,j} = d_{i,j} + s_j$. The matrix cost $C := [c_{i,j}]$ is a square matrix of order $n+1$, that is not necessarily symmetric.

In graph theory, a finite path, ϕ, of length r is a sequence of $r+1$ vertices, $\{\alpha_0,\ \alpha_1,\dots,\ \alpha_r\}$, together with a sequence of r arcs, $\{\phi^1,\ \phi^2,\dots,\ \phi^r\}$, such that

$$\phi^k = (\alpha_{k-1},\ \alpha_k),\ k = 1,\ 2,\dots,\ r.$$

Sometimes we will denote a finite path, ϕ, in the form:

$$\phi: \ \alpha_0 \to \alpha_1 \to \alpha_2 \to \ldots \to \alpha_{r-1} \to \alpha_r.$$

The vertex α_0 is called the start vertex and the vertex α_r is called the end vertex of the path. Both of them are called terminal vertices of the path. The other vertices in the path are internal vertices. A finite cycle is a path such that the start vertex and end vertex are the same. Note that the choice of the start vertex in a cycle is arbitrary. A path with no repeated vertices is called a simple path, and a cycle with no repeated vertices or arcs aside from the necessary repetition of the start and end vertex is a simple cycle.

Definition 5.1 *In our context, a route, ρ, of order r is a simple finite cycle of length $r+2$ in which the start vertex and the end vertex is the depot node* 0*,*

$$\rho: \ 0 \to \alpha_1 \to \alpha_2 \to \ldots \to \alpha_{r-1} \to \alpha_r \to 0.$$

We denote, $\mathscr{R}$, the set of all routes of the complete directed graph G. Notice that the cardinality of $\mathscr{R}$ is $|\mathscr{R}| = \sum_{k=1}^{n} P(n,\ k)$, where $P(n,\ k)$ represents the number of k-permutations of a set of n elements. Notice that $|\mathscr{R}| = \sum_{k=1}^{n} P(n,\ k) \approx n!e$, where e represents the Euler's number, $e = \sum_{k=0}^{\infty} \frac{1}{k!}$.

Definition 5.2 *Two routes are independent when they have no internal vertices in common, i.e., the only vertex in common is the depot node. A non-empty set of independents routes, $\mathscr{S} \subset \mathscr{R}$, is named a* **complete system of routes** *when evert customer belongs to a route of $\mathscr{S}$.*

Traditionally the cost of a route, γ_ρ, has been modeled as $\gamma_\rho = \sum_{k=1}^{r} c_{\alpha_{k-1},\ \alpha_k}$. Then the optimization problem that we want to solve consists of finding a complete system of routes, $\mathscr{S}$, minimizing the total cost, $c_T := \sum_{\rho \in \mathscr{S}} \gamma_\rho$ subject to the following two constraints:

i) the cost of each route, $\rho \in \mathscr{S}$, does not exceed a maximum route cost, $C_{\max}$,

$$\gamma_\rho \leq C_{\max},$$

ii) the total demand served in each route $\rho \in \mathscr{S}$, with internal vertices, $\alpha_1, \ldots,\ \alpha_{r-1}$, does not exceed a maximum constant demand Q_{max},

$$\sum_{k=1}^{r-1} q_{\alpha_k} \leq Q_{max}.$$

As mentioned in the Introduction, one of the main goals of this paper is to fill the gap in the CVRP literature regarding the discussion and solving of non-smooth objective functions, and to show the efficiency of our approach to deal with these kind of functions in the CVRP context. In order to test the effectiveness of our procedure and its efficiency in relation to other existing approaches, we relaxed the constraints by violating some conditions, if necessary. We considered soft constraints, which allow conditions to be violated, by incurring some penalty costs that must be added to the objective function rather than considering hard constraints, which constrain the problem to never exceed the maximum route costs (in our model distance and time based). Following Hashimoto et al. ([44]), "*in real-world situations, time window and capacity constraints can be often violated to some extent*". Of course, the same analysis can be applied to constraints associated with maximum route costs. In practice, if a given route exceeds a threshold cost or length, then some penalty costs must be added to the total route costs, and these penalty costs are likely to be defined by a piecewise non-smooth function. Those costs will depend on the size of the gap between the actual route costs and the threshold. We know that the cost of a route ρ is given by $\gamma_\rho = \sum_{k=1}^{r} c_{\alpha_{k-1},\,\alpha_k}$. Additionally, we have considered that the penalty costs, c_ρ, of any route ρ is given by the following expression:

$$c_\rho := \begin{cases} \gamma_\rho & \text{if } \gamma_\rho \le C_{\max}, \\ \lambda(\gamma_\rho, C_{\max}) & \text{otherwise,} \end{cases} \tag{1}$$

where the function $\lambda(\gamma_\rho, C_{\max})$ is a penalty cost function, which will be considered in the soft-constraint scenario whenever actual route costs, γ_ρ, exceed a given threshold or the maximum cost per route, $C_{\max}$. In applications expression (1) represents a piecewise cost function, c_ρ, for any route, $\rho \in \mathscr{S}$, in the soft-constraint scenario which, in the worst case, could be a non-linear and non-smooth function. Finally the optimization problem to solve is

$$\begin{aligned} &\text{minimize} \quad c_T := \sum_{\rho \in \mathscr{S}} c_\rho \\ &\text{subject to: } \sum_{t=1}^{r_j} q_{\alpha_t^j} \le Q_{max},\ j = 1, 2, \ldots, s, \end{aligned} \tag{2}$$

where $\mathscr{S}$ is a complete system of routes, r_j is the number of demand vertices in route j, and s is the number of routes.

6 Multi-start biased Randomization of classical Heuristics with Adaptive local search (MIRHA)

In this section, our probabilistic algorithm is explained. We named our approach Multi-start biased Randomization of classical Heuristics with Adaptive local search (MIRHA) since, as we will explain later, the basic aspects of our proposed algorithm is the use of classical greedy heuristics combined with a biased randomization and a local search. The algorithm starts with the solution generated by a classical heuristic and slightly perturbs it by means of a random biased behavior in order to obtain alternative good solutions. It uses random bits to do random choices during the execution of the algorithm, but instead of using the uniform distribution (as most metaheuristics and probabilistic algorithms do) we consider non-uniform and nonsymmetric (biased) distributions, e.g.: the geometric distribution, the decreasing triangular distribution, etc. The use of these biased distributions guides the local search process better, since the most promising movements (according to some well-tested heuristic) are the ones with higher probabilities of being selected at each step of the MIRHA approach. The local search is an improvement of the adaptive local search proposed in [49] that leads to good and robust solutions for the classical Capacitated Vehicle Routing Problem.

MIRHA is related to other metaheuristics proposed in the literature. The closest ones are the Hybrid GRASPs or, more accurately, to reactive GRASPs (see [77], [80] or [29]), the Heuristic Biased Stochastic Sampling (HBSS) of Bresina (1996) [6] and the Probabilistic Tabu Search by Fleurent and Glover ([34]). The common aspects of MIRHA with GRASP are the construction of an initial solution using randomization and afterwards the application of a local search. But there are relevant differences, as the MIRHA does not use a Restrictive Candidate List (RCL), one main characteristic of the GRASP algorithm, and it uses a biased and adaptive non-uniform distribution to select the next element to be included in the solution, while most GRASP implementations only consider uniform distributions. The GRASP proposed by Prais and Ribeiro (see [77]) uses a different RCL, where its size is randomly selected, but they still use a RCL and the uniform distribution. The HBSS proposed by Bresima (1996) [6] is similar to the MIRHA since it uses a biased distribution function combined with a sampling methodology. In fact, the MIRHA methodology can be seen as a natural extension/enhancement of the HBSS methodology. Our approach is similar to the HBSS in the use on non-uniform distributions, however we incorporate a local search step after each solution obtained by the biased sampling. In [29], the authors mentioned the HBSS and say that "This methodology can be directly applied in a GRASP construction phase, by biasing the selection of RCL elements to favor those with higher greedy function values." However, as far as we know, the use of biased non-uniform distribution in the construction phase was not applied before. Also, the HBSS was only applied to a scheduling problem, and it was never applied to VRPs. Fleurent and Glover ([34]) describe a Probabilistic Tabu Search applied to the Quadratic Assignment Problem, where they use a non-uniform biased distribution to construct solution in the Intensification Strategy Phase based on the history (memory). This tabu search method has in common with the proposed method the use of a non-uniform biased distribution, but the structure of the algorithm is quite different and there is the need to set various parameters, meanwhile MIRHA contains no or very few parameters to set.

We will now describe in detail the three main aspects of the MIRHA: The construction of the initial solution using classical heuristics, the biased randomization applied to construction of a random solution and the local search method (see Figure 2 for a general pseudocode of the MIRHA).

To construct the random solution we apply a classical greedy heuristic. We have chosen classical heuristics as our starting point for several reasons. First of all, there

Fig. 2 MIRHA general framework

Procedure: MIRHA

```
begin
    Initialization:
        inputData ← Data of the instance considered;
        heuristic ← Heuristic choosed;
        prob.Dist ← Distribution probability used to perform the sampling;
        bestSolution ← get a random solution depending of inputData, heuristic and prob.Dist.;
        bestSolution ← adaptiveLocalSearch(bestSolution);
        stop ← false;
    while stop = false do
        solution ← get a random solution depending of inputData, heuristic and prob.Dist.;
        solution ← adaptiveLocalSearch(solution);
        if c_T(solution) < c_T(bestSolution) then
            bestSolution ← solution;
        end if
        stop ← evaluation stop rule (true or false);
    end while;
    return bestSolution;
end
```

are efficient heuristics for almost every combinatorial optimization problem. They usually exhibit excellent computing performance, they provide acceptable results in most cases, and they have been extensively evaluated in different scenarios. In addition, classical heuristics build the solution incrementally using well-tested strategies instead of directly using the objective function itself. Thus, issues such as non-convexity or non-smoothness of the objective function are not likely to have a significant impact on their efficiency. The main idea of these heuristics is to select the next step from a list of available movements, usually according to a greedy criterion. For example, the Clarke and Wright heuristic for the VRP selects the arc with the highest savings, while the NEH heuristic ([67]) for the Flow-Shop Permutation Problem takes the job with the largest total processing time. As commented before, our proposal is to introduce a biased random behavior in the selection step, but still take into account "*common sense*" rules enforced by the deterministic heuristics. More exactly, instead of having a single choice at every step, we will have multiple choices, each with a decreasing probability of being chosen.

Another relevant aspect of MIRHA is the introduction of biased random behavior in the selection step in the construction phase of the algorithm, but still takes into account the common sense rules enforced by the deterministic heuristics. More exactly, instead of having a single choice at every step, we will have multiple choices, each with a decreasing probability of being chosen. On the one hand, MIRHA proposes the use of theoretical (biased) statistical distributions -with few or no parameters- to perform the stochastic sampling. On the other hand, it also proposes the introduction of an adaptive intensification or local search phase which complements each constructive phase inside the multi-start procedure i.e., the multi-start procedure will consist of two phases: (a) a constructive phase, in which a classical heuristic for the considered problem is selected and then randomized using a biased probability distribution, and (b) an improvement phase which is adaptive in the sense that it will also depend on the particular combinatorial optimization problem being considered -a different local search should be defined for each problem. In the VRP case with non-smooth objective functions, we propose an improvement phase based on the combined use of a memory strategy and a splitting strategy, which has proven to be efficient to solve classical VRP benchmarks ([49]).

Figure 2 shows the commented pseudo-code of the MIRHA general framework. After initializing the pseudo-random number generator, the multi-start procedure (while loop) begins. This loop will terminate as soon as a stopping condition (e.g.: number of iterations or maximum running time) will be satisfied. For each iteration of this loop, two processes are carried out. First, a new solution is constructed using a biased randomization version of the selected classical heuristic, which, obviously, will depend on the type of combinatorial optimization problem at hand (e.g. vehicle routing, scheduling, arc routing, p-median, etc.). Second, an adaptive local search is employed in order to improve the randomized solution. This local search will be different for each type of problem. In this paper, the local search is adapted or tailored for the vehicle routing problem. Then, whenever convenient, the reference to the best solution found so far is updated. The multi-start process not only guarantees that the procedure will not get trapped into a local minimum, but also that different feasible regions in the solution space are sampled and explored, which might be especially useful when coping with non-smooth/non-convex problems where little information is obtained from local improvements. Finally, the best found solution is returned to the decision-maker. Notice that since the described constructive method will be typically very fast, a top list containing the best solutions found could be returned instead.

The local search procedure designed for the vehicle routing problem is shown in Figure 3. It basically consist of two parts: a fast memory-based improvement for individual routes -best found ways to travel the nodes of a given route are stored in cache-, and a divide-and-conquer strategy in which each randomized solution is partitioned into different sub-solutions or regions according to a set of policies, and then a new multi-start heuristic biased randomization process is applied on each of these regions to try to improve the sub-solutions. This divide-and-conquer strategy tries to take advantage of the fact that solving smaller problems is much easier than solving the entire problem and also that the union of all regions' sub-solutions will become a global solution for the entire problem. The proposed method is an extension and improvement of the one proposed in [49]; in this work, a probabilistic algorithm (SR-GCWS-CS) is presented that combines Monte Carlo simulation with splitting techniques and the Clarke and Wright savings heuristic to find solutions to the Capacitated Vehicle Routing Problem (CVRP). Results show that the SR-GCWS-CS can be a real alternative to other metaheuristics since it is able to provide top-quality solutions to most tested benchmarks in reasonable times. The main differences between these local search approaches reside in the way the split-

Fig. 3 MIRHA Adaptative Local Search for the CVRP

Procedure: adaptiveLocalSearch

begin
  Initialization:
    solution ← the obtained random solution from MIRHA;
    solution ← improve solution using solutions in the cache;
    bestSolution ← solution;
    $P \leftarrow \{p_1, \ldots, p_N\}$; (for each $i = 1, \ldots, N$, p_i is a policy to split the solution)
  for i = 1 **to** N **do**
    $\{\text{subSols}_1, \ldots, \text{subSols}_M\}$ ← split solution according p_i;
    for i = 1 **to** M **do**
      if $c_T(\text{subSols}_i) < c_T(\text{bestSubSol}_i)$ **then**
        $\text{bestSubSol}_i \leftarrow \text{subSols}_i$;
      end if
      inputData_i ← Data associate with bestSubSol_i;
      stop ← false;
      while stop = false **do**
        newSubSol_i ← get a random bestSubSol_i depending of inputData_i, heuristic and prob.Dist.;
        newSubSol_i ← improve newSubSol_i using solutions in the cache;
        if $c_{\rho_i}(\text{newSubSol}_i) < c_{\rho_i}(bestSubSol_i)$ **then**
          $\text{bestSubSol}_i \leftarrow \text{newSubSol}_i$;
        end if
        stop ← evaluation stop rule (true or false);
      end while
    end for
    newSolution ← unify the best subsolutions $\{\text{bestSubSol}_1, \ldots, \text{bestSubSol}_M\}$;
    if $c_T(\text{newSolution}) < c_T(\text{solution})$ **then**
      solution ← newSolution;
    end if
  end for
  if $c_T(\text{solution}) < c_T(\text{bestSolution})$ **then**
    bestSolution ← solution;
  end if
  return bestSolution;
end

ting is performed, since they differ both in the number and type of policies used as well as in the number of regions or sub-solutions considered.

The main objective of this work was to develop a general solution approach to solve in an efficient way realistic vehicle routing problems under more complex scenarios dominated by non-smooth/non-convex objective functions and non-convex regions. As we have seen in this section, one important advantage of the proposed algorithm is its robustness and simplicity and, in particular, the fact that it employs very few or no parameters, so there is no need to perform a complex fine-tuning process before using it. Also, the MIRHA is simple to design and implement since, for most combinatorial optimization problems, there exist a classical greedy algorithm and a local search. In the next section, we will present the computational results and discuss the accuracy (or effectiveness) and efficiency of the MIRHA approach.

7 Numerical experiments regarding the solving of non-smooth vehicle routing problems

In order to test the effectiveness of our approach and its efficiency as compared with other existing approaches, we used the classical CVRP benchmark instances from Christofides, Mingozzi and Toth ([18]) which feature the special constraints of the problem considered here, namely vrpnc6, vrpnc7, vrpnc8, vrpnc9, vrpnc10, vrpnc13 and vrpnc14 (see Table 1).

Table 1 Characteristics of the vehicle routing instances

Instance	*Nodes*	*Vehicle Capacity*	*Max. Route Costs*	*Service Cost*
vrpnc6	51	160	200	10
vrpnc7	76	140	160	10
vrpnc8	101	200	230	10
vrpnc9	151	200	200	10
vrpnc10	200	200	200	10
vrpnc13	121	200	720	50
vrpnc14	101	200	1040	90
Averages	114	186	393	27

However, instead of considering the maximum route costs (or length) as a hard constraint we have considered it as a soft constraint that could be eventually violated if necessary by incurring some penalty costs. Following the penalty costs function introduced in Section 5, in this example, we have used the specific non-linear and non-smooth function:

$$\lambda(\gamma_\rho, C_{\max}) := \gamma_\rho + \min\{\theta(\gamma_\rho, C_{\max}), 20\}, \tag{3}$$

where

$$\theta(\gamma_\rho, C_{max}) := 5 + 5000\left(\frac{\gamma_\rho - C_{\max}}{\gamma_\rho}\right)^4. \tag{4}$$

Table 1 contains the following information for each instance: name of instance, number of nodes, vehicle capacity, maximum route cost (threshold) and cost per service, given in the benchmark data. We are interested in comparing results when threshold is considered as a hard constraint and as a soft one.

The MIRHA algorithm described in the previous section has been implemented as a Java application. Being an interpreted language, Java-based programs do not execute as fast as other compiled programs such as those developed in C, but Java allows for rapid development of object-oriented prototypes that can be used to test the potential of an algorithm. At the core of our Java application, some state-of-the-art pseudo-random number generators are employed. In particular, we have used some classes from the SSJ library ([56]), among them, the subclass LFSR113, which implements a very fast generator with a period value approximately equal to 2^{113}. Preliminary research indicates that using a high-quality pseudo-random number generator may be especially useful when performing an in-depth random search of the solution space. Moreover, the use of such a long-period RNG has other important advantages: splitting the RNG sequence in different streams and using each stream in different processors or threads can easily parallelize the algorithm. A standard personal computer, with an Intel ®Core^{TM}2 Duo CPU processor at 2.4 GHz and a 2 GB RAM, was used to perform all tests. For each instance, a maximum computational time of 5 minutes was allowed.

Results of these tests are summarized in Tables 2 and 3. For each instance, Table 2 contains the following information: name of instance; best-known solution when considering hard constraints (BKS-H) as published in Lin et al. ([57]); solution provided by the CWS heuristic Clark and Wright ([14]) when considering hard constraints (CWS-H); the gap in percentage between the CWS-H and BKS-H solutions; best solution obtained with our approach when considering hard constraints (MIRHA-H); gap in percentage between the MIRHA-H and BKS-H solutions; best solution obtained by GRASP using a restricted candidate list considering only k percent of edges (GRASP-H), where k has the following values: 10%, 15%, 25%, 50%, 75%, and 100%; and the gap in percentage between (GRASP-H) and BKS-H solutions.

Table 3 contains the information: name of instance; best-known solution when considering hard constraints (BKS-H); solution provided by the CWS heuristic when considering soft constraints (CWS-S); the gap in percentage between the CWS-S and BKS-H solutions; best solution obtained with our approach when considering soft constraints (MIRHA-S); the gap in percentage between MIRHA-S and BKS-H; best solution obtained by GRASP using a restricted candidate list considering only k percent of edges (GRASP-S) when considering soft constraints, where k has the following values: 10%, 15%, 25%, 50%, 75%, and 100%; and the gap in percentage between (GRASP-S) and BKS-H solutions.

In Tables 2 and 3, the gap in percentage between solution value v_Δ produced by a given methodology Δ and the best known solution value v^* of the instance is calculated as

$$100(v_\Delta - v^*)/v^*.$$

From the results in Tables 2 and 3, it follows that the proposed methodology is already quite efficient in the case of the hard constraint scenario since the average gap with respect to the best-known solutions is only 0.91%, which makes our approach competitive with most state-of-the-art metaheuristics. But even more interesting is the fact that by relaxing somewhat the hard constraint and transforming it into a soft one -which, as discussed previously, might make sense in most real-life situations-our approach has been able to cope with a non-smooth objective function and improve the best-known solutions associated with the hard-constraint scenario (average gap = -6.23%). Of course, the numerical experiment presented in this section is just an illustrative example. One might argue that the results depend upon the specific penalty function considered in case a constraint is violated. For example, if the penalty cost is large enough, then no improvement will be obtained. However, for

Table 2 Comparison among BKS-H, CWS-H, MIRHA-H, and GRASP-H algorithms by using hard constraints

Instances	*(1) BKS-H*	*(2) CWS-H*	*gap (1)-(2)*	*(3) MIRHA-H*	*gap (1)-(3)*	*(4) GRASP-H*	*gap (1)-(4)*	*(5) GRASP-H*	*gap (1)-(5)*	*(6) GRASP-H*	*gap (1)-(6)*	*(7) GRASP-H*	*gap (1)-(7)*	*(8) GRASP-H*	*gap (1)-(8)*	*(9) GRASP-H*	*gap (1)-(9)*
						RCL [a] = 10%		*RCL = 15%*		*RCL = 25%*		*RCL = 50%*		*RCL = 75%*		*RCL = 100%*	
vrpnc6	555.43	618.39	11.34%	555.43	0.00%	581.76	4.74%	580.65	4.54%	555.43	0.00%	557.49	0.37%	618.39	11.34%	618.39	11.34%
vrpnc7	909.68	975.46	7.23%	915.4	0.63%	927.26	1.93%	926.92	1.90%	924.04	1.58%	975.46	7.23%	975.46	7.23%	975.46	7.23%
vrpnc8	865.94	973.94	12.47%	867.58	0.19%	899.52	3.88%	884.35	2.13%	883.26	2.00%	973.94	12.47%	973.94	12.47%	973.94	12.47%
vrpnc9	1,162.55	1,287.64	10.76%	1,188.63	2.24%	1,189.85	2.35%	1,193.24	2.64%	1,287.64	10.76%	1,287.64	10.76%	1,287.64	10.76%	1,287.64	10.76%
vrpnc10	1,395.85	1,538.66	10.23%	1,435.79	2.86%	1,457.7	4.43%	1,473.96	5.60%	1,538.66	10.23%	1,538.66	10.23%	1538.66	10.23%	1538.66	10.23%
vrpnc13	1,541.14	1,592.26	3.32%	1,547.79	0.43%	1,592.26	3.32%	1,592.26	3.32%	1,592.26	3.32%	1,592.26	3.32%	1,592.26	3.32%	1,592.26	3.32%
vrpnc14	866.37	875.75	1.08%	866.37	0.00%	866.37	0.00%	866.37	0.00%	866.37	0.00%	875.75	1.08%	875.75	1.08%	875.75	1.08%
Averages			**8.06%**		**0.91%**		**2.95%**		**2.87%**		**3.98%**		**6.49%**		**8.06%**		**8.06%**

[a] RCL represents the percentage of edges considered in the restricted candidate list of GRASP

Results for columns (3) to (9) have been obtained in a maximum computational time of 5 minutes

Table 3 Comparison among BKS-H, CWS-S, MIRHA-S, and GRASP-S algorithms by using soft constraints

Instances	*(1) BKS-H*	*(2) CWS-S*	*gap (1)-(2)*	*(3) MIRHA-S*	*gap (1)-(3)*	*(4) GRASP-S*	*gap (1)-(4)*	*(5) GRASP-S*	*gap (1)-(5)*	*(6) GRASP-S*	*gap (1)-(6)*	*(7) GRASP-S*	*gap (1)-(7)*	*(8) GRASP-S*	*gap (1)-(8)*	*(9) GRASP-S*	*gap (1)-(9)*
						RCL [a] = 10%		*RCL = 15%*		*RCL = 25%*		*RCL = 50%*		*RCL = 75%*		*RCL = 100%*	
vrpnc6	555.43	629.88	13.4%	534.78	−3.72%	558.27	0.51%	556.46	0.19%	534.8	−3.71%	534.8	−3.71%	629.88	13.40%	629.88	13.40%
vrpnc7	909.68	975.89	7.28%	874.84	−3.83%	892.7	−1.87%	884.11	−2.81%	883.53	−2.87%	975.89	7.28%	975.89	7.28%	975.89	7.28%
vrpnc8	865.94	941.17	8.69%	848.73	−1.99%	873.3	0.85%	874.65	1.01%	874.72	1.01%	941.17	8.69%	941.17	8.69%	941.17	8.69%
vrpnc9	1,162.55	1,252.59	7.75%	1,112.38	−4.32%	1,134.34	−2.43%	1132.1	−2.62%	1,252.59	7.75%	1,252.59	7.75%	1,252.59	7.75%	1,252.59	7.75%
vrpnc10	1,395.85	1,475.57	5.71%	1,389.32	−0.47%	1,391.23	−0.33%	1,475.57	5.71%	1,475.57	5.71%	1,475.57	5.71%	1,475.57	5.71%	1,457.57	5.71%
vrpnc13	1,541.14	1,194.52	−22.49%	1,139.22	−26.08%	1,145.58	−25.67%	1,194.52	−22.49%	1,194.52	−22.49%	1,194.52	−22.49%	1,194.52	−22.49%	1,194.52	−22.49%
vrpnc14	866.37	868.68	0.04%	838.63	−3.20%	838.64	−3.20%	838.64	−3.20%	838.63	−3.20%	866.68	0.04%	866.68	0.04%	866.68	0.04%
Averages			**2.91%**		**-6.23%**		**-4.59%**		**-3.46%**		**-2.54%**		**0.46%**		**2.91%**		**2.91%**

[a] RCL represents the percentage of edges considered in the restricted candidate list of GRASP

Results for columns (3) to (9) have been obtained in a maximum computational time of 5 minutes moderated penalty costs, our methodology can provide improvements even when the routing costs are modeled through non-smooth objective functions.

We can summarize our results as follows. Using the GRASP method without a restricted candidate list never outperforms CWS. Thus, one needs to restrict the candidate list, so that it can compete with MIRHA but with a downside. An extra parameter aimed at setting the size of the restricted candidate list must be added which requires a time-consuming fine-tuning process. And the results might significantly depend on that fine-tuned parameter. We should note that since GRASP uses our local search, it is actually an optimized version of GRASP. But, it is still not able to compete against our algorithm, MIRHA. When we use a restrictive candidate list, then GRASP improves and is able to beat the CWS. However, the best values obtained with GRASP are worse over all instances when compared to our best values, even when different restricted candidate list sizes are tried. Moreover, as the size of the problem increases, the gap between our approach and GRASP increases.

8 Conclusions

In this paper, the multi-start biased randomization of classical heuristics with adaptive local search (MIRHA) algorithm is proposed as a method for solving non-smooth/non-convex vehicle routing problems. The key idea in our approach is to employ probability distributions such as the geometric one to add a random biased behavior to classical heuristics, e.g. the savings method. In this way we obtain a large set of alternative good solutions that outperform the initial solution produced by the heuristic. After that, an adaptive local search phase is incorporated to the multi-start process in order to further improve the randomized solution. An overview of non-convex and non-smooth optimization problems has been given to introduce the reader to the topic. We have also discussed how different approaches have been used for solving this type of problems. Among others, Genetic and Evolutionary Algorithms, Tabu Search, or Simulated Annealing. As it has been pointed out, our methodology has similarities with some methods already reported in the literature but, at the same time, it maintains significant differences, as previously discussed. Computational results show the efficiency of our approach when dealing with VRP with non-smooth objective functions, a topic rarely discussed in the VRP literature so far but which, in our opinion, might attract a lot of interest from VRP researchers in the future.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation (grants MTM2008-06695-C03-03, TRA2010-21644-C03 and ECO2009-11307), by the Working Community of the Pyrenees (grants IIQ13172.R11-CTP09-

R2 and CTP09-00007), by the CYTED-IN3-HAROSA network (http://dpcs.uoc.edu) and by the Jerónimo de Ayanz network (Government of Navarre, Spain).

References

1. Al-Sultan, K.,S., 1995. A tabu search approach to the clustering problem. Pattern Recognition, 28 (9), 1443-1451.
2. Bagirov, A., M., and Yearwood, J., 2006. A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems, Eur. J. Oper. Res., 170, 578-596.
3. Bagirov, A., Lai, D. T. H., and Palaniswami, M., 2007. A nonsmooth optimization approach to sensor network localization. In M. Palaniswami, M. Marusic & Y. W. Law (Eds.), Proceedings of the 2007 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 727-732.
4. Bauer, J., Bektas, T., and Crainic, T.G., 2010. Minimizing Greenhouse Gas Emissions in Intermodal Freight Transport: an Application to Rail Service Design. Journal of Operational Research Society, 61, 530-542.
5. Boyd S., and Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press, Cambridge.
6. Bresina, J. L., 1996. Heuristic-Biased Stochastic Sampling. Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference, Vols 1 and 2, 271-278.
7. Bullnheimer, B., Hartl, R.F., and Strauss, C., 1999. An Improved Ant System for the Vehicle Routing Problem. Annals of Operations Research, 89, 319-328.
8. Burkard R.E., Çela, E., Pardalos, P. , and Pitsoulis, L.S., 1998. The quadratic assignment problem. In: D.-Z. Du and P.M. Pardalos, Editors, Handbook of Combinatorial Optimization, Vol. 3, Kluwer Academic Publishers, Dordrecht, 241-339.
9. Carreto, C., and B. Baker., 2002. A GRASP interactive approach to the vehicle routing problem with backhauls. In C. Ribeiro and P. Hansen (Eds.), Essays and surveys in metaheuristics, Kluwer Academic Publishers, Boston, , 185-200.
10. Chaovalitwongse W., Kim, D., and Pardalos, P., 2003. GRASP with a new local search scheme for vehicle routing problems with time windows. J. of Combinatorial Optimization 7, 179-207.
11. Chaves, A.A., and Lorena, L.A.N., 2010. Clustering search algorithm for the capacitated centered clustering problem. Computers & Operations Research, 37(3), 552-558.
12. Chiou, S.W., 2007. A non-smooth optimization model for a two-tiered supply chain network. Information Sciences, 177, 5754-5762.
13. Chiou, S.W., 2008. A non-smooth model for signalized road network design problems. Applied Mathematical Modelling, 32, 1179-1190.
14. Clarke G., and Wright, J., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12 (4), 568-581.
15. Collet P., and Rennard J.P., 2006. Stochastic Optimization Algorithms, in Handbook of Research on Nature Inspired Computing for Economics and Management, Rennard, J.P. (ed.) Idea Group Reference, Hershey-PA, USA.
16. Cordeau J.F., Gendreau M., Laporte G., Potvin J.Y. and Semet F., 2002. A guide to vehicle routing heuristics, Journal of the Operational Research Society, 53, 512-522.
17. Cordeau J.F., Gendreau M., Hertz A., Laporte G., and Sormany J.S., 2004. New Heuristics for the Vehicle Routing Problem. In: Langevin A., and Riopel D. (eds). Logistics Systems: Design and Optimization. Kluwer Academic Publishers.
18. Christofides, N., Mingozzi, A., and Toth, P., 1979. The vehicle routing problem. In Combinatorial Optimization, 315-338, ed. N. Christofides, A. Mingozzi, P. Toth, and L. Sandi. Wiley, Chichester, U.K.

19. Davis L.D., 1991. Handbook of Genetic Algorithms. Van Nostrand Reinhold. New York, NY.
20. De Jong K.A., 2006. Evolutionary Computation: A Unified Approach, The MIT Press. Cambridge, MA.
21. Derigs U., Li B., and Vogel, U,. 2010. Local search-based metaheuristics for the split delivery vehicle routing problem. Journal of the Operational Research Society, 61(9), 1356-1364.
22. Derigs, U., and Metz, A., 1992. A matching-based approach for solving a delivery pick-up vehicle-routing problem with time constraints. OR Spektrum, 14(2), 91-106.
23. Dorigo M., and Stützle T., 2010. Ant Colony Optimization: Overview and Recent Advances. In Handbook of Metaheuristics, 2nd. Edition. Vol. 146. M. Gendreau, and Potvin, J.Y. (eds), Kluwer Academic Publishers, International Series in Operations Research & Management Science, 227-264.
24. Drezner, Z., and Hamacher, H. (eds.), 2002. Facility Location: Applications and Theory, New York, Springer.
25. Fagerholt, K., Laporte, G., and Norstad, I., 2010. Reducing fuel emissions by optimizing speed on shipping routes. Journal of Operational Research Society, 61, 523 -529.
26. Faulin J., Gilibert M., Juan A., Ruiz, R., and Vilajosana, X., 2008. SR-1: A Simulation-based Algorithm for the Capacitated Vehicle Routing Problem. In: Proceedings of the 2008 Winter Simulation Conference, 2708-2716.
27. Faulin, J., and Juan, A., 2008. The ALGACEA-1 Method for the Capacitated Vehicle Routing Problem, International Transactions in Operational Research, 15, 599-621.
28. Feo T.A., Resende M.G.C., 1995. Greedy randomized adaptive search procedures. Journal of Global Optimization, 6, 109-133.
29. Festa, P., and Resende, M.G.C., 2009a. An Annotated Bibliography of GRASP- Part I: Algorithms. International Transactions in Operational Research 16, 1-24.
30. Festa, P., and Resende, M.G.C., 2009b. An Annotated Bibliography of GRASP- Part II: Applications. International Transactions in Operational Research 16, 131-172.
31. Festa, P., and Resende, M. G. C., 2002. GRASP: An annotated bibliography. In C. C. Ribeiro & P. Hansen (Eds.), Essays and Surveys in Metaheuristics, 15, 325-367.
32. Figliozzi, M.A., 2011. The impacts of congestion on time-definitive urban freight distribution networks CO_2 emission levels: Results from a case study in Portland, Oregon. Transportation Research Part C, 19(5), 766-778.
33. Fisher, M.L., and Jaikumar, R., 1981. A Generalized Assignment Heuristic for Vehicle Routing. Networks, 11, 109-124.
34. Fleurent, C., and Glover, F., 1999. Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. INFORMS Journal on Computing, 11(2), 198-204.
35. Fogel, L.J., 1962. Autonomous automata. Industrial Research, 4, 14-19.
36. Funke, B., Grunert, T., and Irnich, S., 2005. Local search for vehicle routing and scheduling problems: Review and conceptual integration. Journal of Heuristics, 11(4), 267-306.
37. Gendreau M., Hertz A., and Laporte G., 1994. A Tabu Search Heuristic for the Vehicle Routing Problem. Management Science, 40, 1276-1290.
38. Gersht, A., and Shulman, A., 1989. Optimal Routing in Circuit Switched Communication Networks, 37, 1203-1211.
39. Gillet, B.E., and Miller, L.R., 1974. A Heuristic Algorithm for the Vehicle Dispatch Problem, Operations Research, 22, 340-349.
40. Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley. Reading, MA. USA
41. Golden B., Raghavan S., and Wasil E.E. (eds.), 2008. The Vehicle Routing Problem: Latest Advances and New Challenges. Springer. New York, USA
42. Hamdan, M., and El-Hawary, M.E., 2002. Hopfield-Genetic Approach for Solving the Routing Problem In computer Networks. Proceedings of the 2002 IEEE Canadian Conference on Electrical and Computer Engineering, 823-827.
43. Hansen P., Mladenovic N., Brimberg J., and Moreno-Pérez J.A., 2010, Variable Neighborhood Search. In Handbook of Metaheuristics, 2nd. Edition. Vol.146. M. Gendreau, and Potvin, J.Y. (eds.), Kluwer Academic Publishers, International Series in Operations Research & Management Science, 61-86.

44. Hashimoto, H., Ibaraki, T., Imahori, S., & Yagiura, M., 2006. The vehicle routing problem with flexible time windows and traveling times. Discrete Applied Mathematics, 154(16), 2271-2290.
45. Hemamalini, S., and Simon, S. P., 2010. Artificial Bee Colony Algorithm for Economic Load Dispatch Problem with Non-smooth Cost Functions. Electric Power Components and Systems, 38(7), 786-803.
46. Holland, J.H., 1975. Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor
47. Hopfield, J.J., and Tank, D.W., 1985. Neural Computation of Decisions in Optimization Problems. Biological Cybernetics, 52, 21-43.
48. Juan A., Faulin J., Ruiz, R., Barrios B., Gilibert M., and Vilajosana X., 2009. Using oriented random search to provide a set of alternative solutions to the capacitated vehicle routing problem. In: Chinneck J, Kristjansson B, Saltzman M. (Eds.): Operations Research and Cyber-Infrastructure. Operations Research/Computer Science Interfaces Series, 47, 331-346. ISBN: 978-0-387-88842-2. Springer, New York, USA.
49. Juan A., Faulin J., Jorba J., Riera D., Masip D., and Barrios B., 2010. On the Use of Monte Carlo Simulation, Cache and Splitting Techniques to Improve the Clarke and Wright Savings Heuristics. Journal of the Operational Research Society, 62(6), 1085-1097.
50. Kant, G., Jacks, M., and Aantjes, C., 2008. Coca-Cola Enterprises Optimizes Vehicle Routes for Efficient Product Delivery. Interfaces, 38: 40-50.
51. Kennedy, J., and Eberhart, R., 1995. Particle swarm optimization 1995 IEEE International Conference on Neural Networks Proceedings, Vols 1-6, 1942-1948.
52. Kirkpatrick, S., Gelatt Jr., C. D., and Vecchi, M.P., 1983. Optimization by simulated annealing. Science, 220.
53. Kontoravdis, G., and Bard, J., 1995. A GRASP for the vehicle routing problem with time windows. ORSA Journal on Computing 7, 10-23.
54. Kumares C., and Labi S., 2007. Transportation Decision Making: Principles of Project Evaluation and Programming. Wiley. Hoboken, NJ. USA
55. Laporte, G., 2009. Fifty Years of Vehicle Routing. Transportation Science, 43(4), 408-416.
56. L'Ecuyer, P., 2006. Random Number Generation in Simulation. Elsevier Science, Amsterdam, Holland
57. Lin, S., Lee, Z., Ying, K. , and Lee., C., 2009. Applying hybrid meta-heuristics for capacitated vehicle routing problem. Expert Systems with Applications 36(2), 1505-1512.
58. Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P. and Querido, T., 2007. A survey for the quadratic assignment problem. European Journal of Operational Research, 176(2), 657-690.
59. Lokketangen, A., and Glover, F., 1998. Solving zero-one mixed integer programming problems using tabu search. European Journal of Operational Research, 106(2-3), 624-658.
60. Lourenço H.R., Martin O., and Stützle T., 2003. Iterated Local Search. In Handbook of Metaheuristics, F. Glover and G. Kochenberger, (eds.), Kluwer Academic Publishers, International Series in Operations Research & Management Science, 321-353.
61. Lourenço H.R., Martin O., and Stützle T., 2010. Iterated Local Search: Framework and Applications. In Handbook of Metaheuristics, 2nd. Edition. Vol.146. M. Gendreau, and J.Y. Potvin (eds.), Kluwer Academic Publishers, International Series in Operations Research & Management Science, 363-397.
62. Lourenço, H.R., and Ribeiro, R., 2011. Strategies for an Integrated Distribution Problem, In: Montoya-Torres, J. , Juan, A., Huaccho-Guatuco, L., Faulin, J., and Rodríguez-Verján, G. (eds) Hybrid Algorithms for Service, Computing and Manufacturing Systems: Routing, Scheduling and Availability Solutions. IGI Global Books. Hershey. PA. (to appear)
63. Mester D., and Bräysy O., 2007. Active-guided evolution strategies for the large-scale capacitated vehicle routing problems. Computers and Operations Research, 34, 2964-2975.
64. Mladenovic, N., Brimberg, J., Hansen, P., and Moreno-Perez, J. A., 2007. The p-median problem: A survey of meta-heuristic approaches. European Journal of Operational Research, 179(3), 927-939.

65. Montoya J., Juan A., Huaccho L., Faulin J., Rodriguez G. (eds.) 2011. Hybrid Algorithms for Service, Computing and Manufacturing Systems: Routing and Scheduling Solutions. IGI Global, Hershey, Pennsylvania, USA.
66. Muter, I., Birbil, S. I., and Sahin, G., 2010. Combination of Metaheuristic and Exact Algorithms for Solving Set Covering-Type Optimization Problems. Informs Journal on Computing, 22(4), 603-619.
67. Nawaz, M., Enscore, Jr, E. E., and Ham, I., 1983. A heuristic algorithm for the m-machine, n-job flowshop sequencing problem. OMEGA, The International Journal of Management Science, 11(1), 91-95.
68. Neyestani, M., Farsangi, M.M., and Nezamabani-Pour, H., 2010. A Modified Particle Swarm Optimization for Economic Dispatch with Non-smooth Cost Functions. Engineering Applications of Artificial Intelligence, 23, 1121-1126.
69. Nikolaev A.G., and Jacobson S.H., 2010. Simulated Annealing. In Handbook of Metaheuristics, 2nd. Edition. Vol.146. M. Gendreau and J.Y. Potvin (eds.), Kluwer Academic Publishers, International Series in Operations Research & Management Science, 1-40.
70. Oncan, T., 2007. A survey of the generalized assignment problem and its applications. Infor, 45(3), 123-141.
71. Oonsivilai, A., Srisuruk, W., Marungsri, B., and Kulworawanichpong, T., 2009. Tabu Search Approach to Solve Rou-ting Issues in Communication Networks. World Academy of Science, Engineering and Technology, 1174-1177.
72. Park J.B., Lee K.S, Shin J.R., Lee K. Y., 2005. A Particle Swarm Optimization for Economic Dispatch with Non-smooth Cost Functions. IEEE Transactions on Power Systems, 20, 34-42.
73. Pinedo, M., 2008. Scheduling: Theory, Algorithms and Systems: Springer. New York.
74. Pisinger, D., and Ropke, S., 2007. A general heuristic for vehicle routing problems. Computers and Operations Research, 34, 2403-2435.
75. Poot A., Kant G., and Wagelmans A., 2002. A savings based method for real-life vehicle routing problems. Journal of the Operational Research Society, 53, 57-68.
76. Potvin, J. Y., 2009. State-of-the Art Review Evolutionary Algorithms for Vehicle Routing. INFORMS Journal on Computing, 21(4), 518-548.
77. Prais M. e Ribeiro C.C. , "Variação de parâmetros em procedimientos GRASP", Investigación Operativa 9 (2000), 1-20.
78. Ramadurai, V., and Sichitiu, M. L., 2003. Localization in wireless sensor networks: a probabilistic approach. International conference on Wireless Networks (ICWN03), 275-281.
79. Reeves C.R., 2010. Genetic Algorithms. In Handbook of Metaheuristics, 2nd. Edition. Vol.146. Gendreau, M., and Potvin, J.Y. (eds.), Kluwer Academic Publishers, International Series in Operations Research & Management Science, 109-140.
80. Resende M.G.C.. Metaheuristic hybridization with Greedy Randomized Adaptive Search Procedures. TutORials in Operations Research, Shi-Long Chen and S. Raghavan (Eds.). 295-319, INFORMS, 2008.
81. Resende M., and Ribeiro C., 2010. Greedy Randomized Adaptive Search Procedures: Advances, Hybridizations and Applications. In Handbook of Metaheuristics, 2nd. Edition. M. Gendreau, and J.Y. Potvin (eds.), Kluwer Academic Publishers, International Series in Operations Research & Management Science, 146, 27-264.
82. Schlüter, M., Egea, J. A., and Banga, J.R., 2009. Extended ant colony optimization for non-convex mixed integer nonlinear programming. Computers & Operations Research, 36(7), 2217-2229.
83. Selim, S.Z., and Al-Sultan, K.S., 1991. A simulated annealing algorithm for the clustering problem. Pattern Recognition, 24 (10), 1003-1008
84. Suman, B., and Kumar, P., 2006. A survey of simulated annealing as a tool for single and multiobjective optimization. Journal of the Operational Research Society, 57(10), 1143-1160.
85. Tarantilis C.D., and Kiranoudis C.T., 2002. Boneroute: an Adaptative Memory-Based Method for Effective Fleet Management. Annals of Operations Research, 115, 227-241.
86. Toscano, R., and Lyonnet, P., 2010. A new heuristic approach for non-convex optimization problems. Information Sciences, 180(10), 1955-1966.

87. Toth P., and Vigo D. (Eds.), 2002. The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications. SIAM. Philadelphia, PA. USA.
88. Urlings, T., and Ruiz, R., 2007. Local search in complex scheduling problems. In T. Stutzle, M. Birattari & H. H. Hoos (Eds.), Engineering Stochastic Local Search Algorithms: Designing, Implementing and Analyzing Effective Heuristics, 4638, 202-206.
89. Vaisakh, K., and Srinivas, L. R., 2011. Genetic evolving ant direction HDE for OPF with non-smooth cost functions and statistical analysis. Expert Systems with Applications, 38(3), 2046-2062.
90. Wright, S., 1997. Primal-Dual Interior-Point Methods. SIAM. Philadelphia, PA.
91. Yang, I. T., and Chou, J. S., 2011. Multiobjective optimization for manpower assignment in consulting engineering firms. Applied Soft Computing, 11(1), 1183-1190.
92. Yang, H. T., Yang, P. C., and Huang, C.L., 1996. Evolutionary Programming Based Economic Dispatch for Units with Non-smooth Fuel Cost Functions. IEEE Transactions on Power Systems, 11, 112-118.