Ann Math Artif Intell
DOI 10.1007/s10472-011-9261-y

Combining probabilistic algorithms, Constraint Programming and Lagrangian Relaxation to solve the Vehicle Routing Problem

Daniel Guimarans · Rosa Herrero · Daniel Riera · Angel A. Juan · Juan José Ramos

Abstract This paper presents an original hybrid approach to solve the Capacitated Vehicle Routing Problem (CVRP). The approach combines a Probabilistic Algorithm with Constraint Programming (CP) and Lagrangian Relaxation (LR). After introducing the CVRP and reviewing the existing literature on the topic, the paper proposes an approach based on a probabilistic Variable Neighbourhood Search (VNS) algorithm. Given a CVRP instance, this algorithm uses a randomized version of the classical Clarke and Wright Savings constructive heuristic to generate a starting solution. This starting solution is then improved through a local search process which combines: (a) LR to optimise each individual route, and (b) CP to quickly verify the feasibility of new proposed solutions. The efficiency of our approach is analysed after testing some well-known CVRP benchmarks. Benefits of our hybrid approach over already existing approaches are also discussed. In particular, the potential flexibility of our methodology is highlighted.

Keywords Hybrid algorithms · Variable Neighborhood Search · Vehicle Routing Problem · Probabilistic algorithms

Mathematics Subject Classification (2010) 90

1 Introduction

The growing flows of freight have been a fundamental component of contemporary changes in economic systems at the global, regional and local scales. Road

D. Guimarans (✉) · R. Herrero · J. J. Ramos
Dpt. de Telecomunicació i Enginyeria de Sistemes, Universitat Autònoma de Barcelona (UAB), Cerdanyola del Vallès, 08193, Barcelona, Spain
e-mail: Daniel.Guimarans@uab.cat

D. Riera · A. A. Juan
Universitat Oberta de Catalunya (UOC), Barcelona, Spain

Published online: 04 August 2011

transportation is nowadays the predominant way of transporting goods in many parts of the world. Direct costs associated with road transportation have experienced a significant increase in the last decade due to the rise of oil price, among other economical factors. Furthermore, road transportation faces new challenges related to other indirect or external related costs, which usually are easily observable—noise, pollution, accidents, etc.—but difficult to quantify. The role of transport and logistics as an economic sector can not be nowadays neglected since new modes of production are concomitant with new modes of distribution. Achieving flexible, efficient and sustainable routing is a complex strategy requiring a high level of logistical integration to properly respond to variations of the freight transport demand. The necessity for optimizing the road transportation affects to both the public and the private sectors, and constitutes a major challenge for most industrialized regions.

The Vehicle Routing Problem (VRP) provides a theoretical framework for approaching the class of logistic problems dealing with physical distribution. This is among the most popular research areas in combinatorial optimization. It was first defined by Dantzig and Ramser in 1959 [10], and several variants of the basic problem have been proposed and studied later. These variants represent different types of operational constraints such as, for instance, time windows, pick up and delivery, heterogeneous fleets or multi-depot problems.

From the industrial applicability perspective, the VRP characterizes a family of different distribution problems which, one way or another, are present in real industrial problems. However, in most of the application cases none of the classical VRP variants can represent uniquely the real problem. That is, a combination of different operational constraints are present in many realistic cases. In this scenario, it becomes evident the need of developing new methods, models and systems to give support to the decision-making process so that optimal strategies can be chosen in physical distribution, in particular, in road transportation.

This paper presents an original hybrid approach to solve the VRP. This methodology has been especially designed for being flexible in the sense that it can be used, with minor adaptations, for solving different variants of the VRP present in industrial application cases. The approach is based on the classical decomposition into two subproblems: a resource allocation problem (to fit operational constraints), and a routing problem (to minimize the associated traveling costs).

The Capacitated version of the VRP (CVRP) has been chosen in order to illustrate the benefits of the proposed methodology. The CVRP is the most basic VRP variant, which assumes a fleet of vehicles of homogeneous capacity housed in a single depot. The CVRP is a generalization of the Traveling Salesman Problem (TSP) and is therefore NP-hard [41]. The CVRP is defined over a complete graph $G = \{I, E\}$, where $I = \{1, 2, \ldots, n\}$ is the node set representing clients to be served plus the depot (node 1), and $E = \{e_{ij} = (i, j) | i, j \in I\}$ is the edge set representing connecting roads, streets, etc. Edges e_{ij} in E have an associated cost $c_{ij} > 0$, which is the traveling cost from node i to node j. It is usual to consider symmetric costs (i.e., $c_{ij} = c_{ji}, \forall i, j \in I$). Moreover, each node i in I has a demand $q_i \geq 0$. A fixed fleet of m identical vehicles, each one with capacity $Q \gg \max\{q_i\}$, is available at the depot to accomplish the required delivery task. Solving the CVRP consists of determining a set of $k \leq m$ routes with minimum total traveling cost and such that (a) each customer is visited exactly once by a single vehicle, (b) each route starts and ends at the depot, and (c) the total demand of the customers assigned to a route does not exceed the vehicle capacity Q. Therefore, a solution of a given CVRP instance is a set of k routes sharing a common starting and finishing node (the depot).

The CVRP has been selected mainly because there are huge amounts of models, techniques, benchmarks, and research on this topic. Hence, the proposed methodology can be easily compared—in terms of computational efficiency and solution quality—with previously existing approaches. Nevertheless, from the perspective of its industrial applicability, the basic CVRP model can be extended to tackle different realistic cases by means of the proposed optimization scheme. These cases include operational constraints beyond the basic vehicle capacity. Affordable examples are limitations on the total driving time of each route, incompatible customer-driver associations or constraints on the customer visiting periods (e.g. customer forbidden visit day when building daily routes for a distribution problem). The proposed optimization approach, as presented in this paper, is specifically designed to deal with those operational constraints that involve the allocation part of the VRP.

This work proposes a Multi-Start Variable Neighborhood Descent (VND) [21] structure whose local search process is supported by Constraint Programming (CP) [39] and Lagrangian Relaxation (LR) [16]. Using the CP paradigm provides the required flexibility to model those operational constraints, beyond vehicle capacity, that are usually present in most real application cases. Due to this approach, adding these constraints is just a constraint modeling issue, i.e., no change on the solving strategy is required to deal with more realistic problems. The LR based algorithm is used to efficiently find the optimal routing solution for each transportation resource. A probabilistic (Randomized) Clarke and Wright Savings (RCWS) [28] constructive method is used to generate initial solutions. This algorithm provides different good quality solutions that are used as seeds to launch the exploration of different regions of the search space. Therefore, the RCWS probabilistic behavior introduces a natural diversification mechanism and turns the scheme into an approach likely to be parallelized.

The main goal of the work presented in this paper is to introduce a general algorithmic framework that integrates randomization, CP, LR, and VNS to efficiently solve industrial VRPs with realistic constraints.

The remainder of this article is structured as follows. The next section provides a literature review on the topic. Section 3 presents the technologies used in this research, while Section 4 explains the adopted approach in detail. Section 5 contains some numerical experiments and the corresponding discussion. Section 6 discusses some of the main benefits of the presented approach. Finally, Section 7 summarizes the main contributions of the paper.

2 Previous work on the Capacitated Vehicle Routing Problem

The Clarke and Wright's Savings (CWS) constructive algorithm [8] is probably the most cited heuristic to solve the CVRP. The CWS is an iterative method that starts out by considering an initial dummy solution in which each customer is served by a dedicated vehicle. Next, the algorithm initiates an iterative process for merging some of the routes in the initial solution. Merging routes can improve the expensive initial solution so that a unique vehicle serves the nodes of the merged route. The merging criterion is based upon the concept of savings. Roughly speaking, given a pair of nodes to be served, a savings value can be assigned to the edge connecting these two nodes. This savings value is given by the reduction in the total cost function due to serving both nodes with the same vehicle instead of using a dedicated vehicle to serve each node—as proposed in the initial dummy solution. This way, the algorithm constructs a list of savings, one for each possible edge connecting two demanding nodes. At each iteration of the merging process, the edge with the largest possible savings is selected from the list as far as the following conditions are satisfied: (a) the nodes defining the edge are adjacent to the depot, and (b) the two corresponding routes can be feasibly merged—i.e., the vehicle capacity is not exceeded after the merging. The CWS algorithm usually provides relatively good solutions, especially for small and medium-size problems, but it also presents difficulties in some cases [17]. Many variants and improvements of the CWS have been proposed in the literature. For a comprehensive discussion on the various CWS variants, the reader is referred to Toth and Vigo [45] and Laporte [30].

Monte Carlo Simulation (MCS) can be defined as a set of techniques that make use of random numbers and statistical distributions to solve certain stochastic and deterministic problems [31]. MCS has proved to be extremely useful for obtaining numerical solutions to complex problems that cannot be efficiently solved by using analytical approaches. Buxey [6] was probably the first author to combine MCS with the CWS algorithm to develop a procedure for the CVRP. This method was revisited by Faulin and Juan [11], who introduced an entropy function to guide the random selection of nodes. MCS has also been used by Fernández de Córdoba et al. [14], Juan et al. [26], Faulin et al. [12] and Juan et al. [27] to solve the CVRP. In this last paper, the authors make use of MCS to develop an efficient randomized version of the CWS heuristic, which we use in our approach to efficiently generate initial solutions.

Another way to address the VRP has been the use of complete methods, which ensure not only to find the solution but also, to prove its optimality. The main drawback of these techniques is that they may only deal with small instances, up to 100 customers [9]. Numerous heuristics (like the ones mentioned above) and metaheuristics have also been studied for different VRP variants. In most cases, these methods may solve larger instances but loosing optimality guarantees.

Using constructive heuristics as a basis, metaheuristics became popular for the VRP during the nineties. Some early examples are the Tabu Route method by Gendreau et al. [18] or the Boneroute method of Tarantilis and Kiranoudis [44]. Tabu search algorithms, like those proposed by Taillard [43] or Toth and Vigo [46] are among the most cited metaheuristics. Genetic algorithms have also played a major role in the development of effective approaches for the VRP. Some examples are the studies of Berger and Barkaoui [4], Prins [36], Mester and Braysy [33] or Nagata [35]. Another important approach to the VRP is given by the Greedy Randomized Adaptive Search Procedure or GRASP [13, 15, 38]. A GRASP algorithm is a multi-start or iterative process in which each GRASP iteration consists of two phases: a construction phase—in which a feasible solution is produced—and a local search phase—in which a local optimum in the neighborhood of the constructed solution is sought. The best overall solution is kept as the result. In the construction phase, a feasible solution is iteratively constructed, one element at a time. At each construction iteration, the choice of the next element to be added is determined by ordering all candidate elements in a candidate list according to a greedy function.

This function measures the (myopic) benefit of selecting each element. The heuristic is adaptive because the benefits associated with every element are updated at each iteration of the construction phase to reflect the changes brought on by the selection of the previous element. The probabilistic component of a GRASP is characterized by the random choice of one of the best candidates in the list, but not necessarily the top candidate. This choice technique allows for different solutions to be obtained at each GRASP iteration.

Among metaheuristics, Variable Neighborhood Search (VNS), introduced for the first time by Mladenovic and Hansen [34], is a quite recent method with far less application examples in VRP research. However, interesting results have been obtained even applying the simplest VNS algorithms, e.g. [22]. Embedding CP and LR approaches into a general VNS framework has also demonstrated to be an effective yet slow method to solve medium and large instances [20]. Combining these techniques provided a methodology able to reach good quality results and even to overcome some best-known solutions. However, the computational efficiency of this methodology is far from state-of-the-art algorithms and becomes an important issue to be addressed.

In this paper we present a hybrid approach combining a randomized version of the CWS savings heuristic, the VNS metaheuristic, CP, and LR. Our approach aims at being an efficient procedure for obtaining quasi-optimal solutions in small- and medium-size CVRP instances and, at the same time, offers some additional advantages over other existing metaheuristics, namely: (a) it is a robust and flexible methodology that can be easily adapted to consider additional constraints and costs; (b) it is able to generate a set of alternative good solutions in a reasonable time period; and (c) it can be easily executed in parallel. As mentioned, we already combined VNS with CP and LR in some previous work [20], but the algorithm presented in this paper is much more competitive with state-of-the-art metaheuristics. Its efficiency has been significantly enhanced by including a multi-start procedure which makes use of a randomized CWS heuristic in order to quickly provide a set of different "good" initial solutions, over which a flexible local-search process is applied. Thus, the VNS diversification procedure is substituted by a multi-start approach, where different regions are explored thanks to the diversity of solutions provided by the randomized CWS algorithm. The local search process has also been enhanced with respect to the previous work by incorporating new data structures, which permit reducing the computational complexity. Finally, the methodology described in the present work has been parallelized to improve its efficiency.

3 Technologies used

3.1 Probabilistic Clarke and Wright Savings algorithm

As discussed in Section 2, in the classic CWS algorithm, the edge with the largest possible savings is selected from the list at each iteration of the merging process, as far as the following conditions are satisfied: (a) the nodes defining the edge are adjacent to the depot, and (b) the two corresponding routes can be feasibly merged—i.e., the vehicle capacity is not exceeded. The approach presented in [29], instead, assigns a selection probability to each edge in the savings list. This probability should be coherent with the savings value associated with each edge, i.e., edges with larger savings will be more likely to be selected from the list than those with smaller savings. In addition, this approach adds this biased random behavior without introducing too many parameters in the algorithm. Basically, different geometric statistical distributions during the randomized CWS solution-construction process are employed: every time a new edge is selected from the list of available edges, a value α is randomly selected from a uniform distribution in (a, b), where $0 < a \leq b < 1$. The α parameter defines the specific geometric distribution that will be used to assign exponentially diminishing probabilities to each eligible edge according to its position inside the sorted savings list. This way, edges with higher savings values are always more likely to be selected from the list, but the exact probabilities assigned are variable and they depend on the concrete distribution selected at each step.

3.2 Constraint programming

CP is a powerful paradigm for representing and solving a wide range of combinatorial problems. Problems are expressed in terms of three entities: variables, their corresponding domains and constraints relating them. The problems can then be solved using complete techniques such as depth-first search for satisfaction and branch and bound for optimization, or even tailored search methods for specific problems. Rossi et al. [39] present a complete overview of CP modeling techniques, algorithms, tools, and applications.

3.3 Lagrangian Relaxation

LR is a well-known method to solve large-scale combinatorial optimization problems. It works by moving hard-to-satisfy constraints into the objective function associating a penalty in case they are not satisfied. An excellent introduction to the whole topic of LR can be found in [16].

LR exploits the structure of the problem, so it reduces considerably problem's complexity. However, it is often a major issue to find optimal Lagrangian multipliers. The most commonly used algorithm is the Subgradient Optimization (SO). Its main difficulty lays on choosing a correct step-size λ_k aiming to ensure algorithm's convergence [37].

In order to address this limitation, the method introduced in [24] combines the SO algorithm with a heuristic to obtain a feasible solution from a dual solution. It can get a better upper bound (UB), so it improves the convergence on the optimal solution starting at an initial UB obtained with a Nearest Neighbor Heuristic. Although optimality may not always be reached, this method is able to provide a feasible solution with a tight gap between the primal and the optimal cost in a reasonable number of iterations.

3.4 Variable Neighborhood Search

A general VNS, as explained in [21], is a recent metaheuristic which exploits systematically the idea of neighborhood change. The Variable Neighborhood Descent (VND) method starts from an initial solution x' and it is improved by a local search process.

The local search process for each neighborhood $N(x')$ of x' performs an exhaustive exploration. All improving movements are recorded and sorted, so the best neighbor $x'' \in N(x')$ is constructed applying all independent changes in descending order. This way, solution values are improved faster than applying single movements.

If this neighbor is better than the incumbent, the current solution is updated and neighborhoods' exploration is restarted. Otherwise, the algorithm keeps x' as the best solution found so far and continues exploring the next neighborhood. When the VND process reaches a local optimum, no solution improvement may be found according to defined neighborhoods.

3.5 Multi-start strategy

The VND-based local search process requires some type of diversification in order to overcome local optimality. Many techniques have been suggested to avoid getting trapped into a local optimum and aspire to find a global one. Among others, one possible way to achieve diversification is using a shaking mechanism within the VNS procedure. However, as more constraints are introduced in the problem, it usually becomes more efficient—in terms of computational time employed—to generate new feasible solutions from scratch than to apply complex shaking processes that might end in non-feasible solutions. This is especially certain if we consider that the Randomized version of the CWS used in this paper is a really fast method for generating different feasible and good solutions that can serve as initial solutions in our multi-start approach.

Thus, the Multi-Start strategy provides an appropriate framework which achieves diversification by re-starting the search from a new solution once a region has been extensively explored. Notice that each iteration includes two phases: a first one in which a new feasible solution is constructed, and a second one in which the initial solution is improved through a local search process.

4 The methodology in detail

The CVRP problem has been tackled using a significant modification of the approach presented in [20]. There, the authors introduced a first methodology which combined CP and LR within a VNS framework. The methodology presented in this paper builds upon the aforementioned one, but it is significantly more efficient—both in terms of computational times and solutions' quality—since it also integrates the RCWS-based Multi-Start approach. As it has been previously discussed, this strategy allows to ensure an efficient diversification of the search space in order to: (a) avoid local minima, and (b) reach near-optimal solutions in reasonable times.

CP and LR are used in the local search process within a VND structure. CP is used to check solutions feasibility. This formalism provides a fast and flexible method, able to model and include complex constraints while keeping a reasonable computational efficiency. In turn, a tailored LR method is applied to calculate routes every time a partial solution is generated. Using LR allows reducing the computation time and algorithm's definition and complexity when compared to other routing post-optimization methods [40].

Table 1 Multi start approach

Let x be the best solution.
Create a thread pool with $Total_{\text{Threads}}$ threads.
Repeat the following steps until $Total_{\text{Threads}}$ threads end or until Max_{Seconds} time is consumed:
    Execute Max_{Threads} simultaneous threads:
        Generate an initial feasible solution x' using RCWS.
        Improve x' to obtain x'' by using VND+CP+LR.
        If x'' is better than x, let $x \leftarrow x''$.

4.1 Pseudo-code for the multi-start strategy

A simplified scheme of the Multi-Start strategy is presented in Table 1. The RCWS algorithm is used to find a good initial solution. Then, the VND method helps to reach a local minimum in the neighborhood of the solution.

The Multi-Start strategy generates $Total_{\text{Threads}}$ tasks within a thread pool. If a thread is not available for the task, the task waits in a queue for an active task to end. The algorithm stops when all tasks have been completed, or the maximum execution time is reached, whichever happens first. Each task executes two phases: find an initial solution and improve it in the search process. Starting from a different initial solution ensures certain diversification, overcoming local optimality.

4.2 Pseudo-code for the Variable Neighborhood Descent procedure

A general VND has been implemented embedding CP and LR methods. In the implemented algorithm, outlined in Table 2, all four described moves (see Section 4.3) have been selected to be used in local search neighborhoods.

In the exploration neighborhood (N_k), starting from the solution x', the kth move is applied and the new solution's feasibility is checked using CP. Whenever it is proved feasible, LR is used to recalculate only modified routes. This approach permits to consider only two routes per solution, reducing the computation time.

Table 2 Variable Neighborhood Descent Algorithm

Initialize the set $LastModified \leftarrow V$; let x' be the initial solution.
Repeat the following sequence until the stopping condition is met:
    Set $k \leftarrow 1$;
    Repeat the following steps until $k = k_{\max}$:
        Exploration of Neighborhood.
            Find all neighbors $x'' \in N_k(x', LastModified)$.
            Check feasibility of capacity constraints using CP.
            Calculate the cost of modified routes using LR
            If the solution x'' is better than x', include it in a list of improving changes.
        Choose the best compatible neighbors.
            Set $LastModified \leftarrow \emptyset$;
            Sort the list of improving changes.
            Apply the first improving changes.
            Add in descending order the next compatible improvements.
            Add the modified routes to $LastModified$.
        If the list is empty, set $k \leftarrow k + 1$; otherwise set $k \leftarrow 1$.

Improvements are stored in a sorted list until no more feasible solutions are left in the kth neighborhood. Then, all those which are independent, i.e., affect different route pairs, are applied in descending order on x' to get a better solution x''. This way, solution improvement is faster than applying a single change at each iteration.

After the first exhaustive exploration of each neighborhood, only those changes affecting routes modified by previous movements are explored in order to reduce the computation time. The modified routes are stored in the set $LastModified$. A similar approach may be found in Zachariadis and Kiranoudis [48].

4.3 Inter-route moves

The VNS metaheuristic is based on the exploration of different neighborhoods around a given feasible solution. In order to establish these neighborhoods, several moves are defined. In our approach, four different inter-route classical moves [42] have been identified to be used within the local search process: (a) *Relocate* moves a customer from one route to another, (b) *Swapping* exchanges two customers belonging to different routes, (c) *Chain* is a specialization of 3-opt that swaps sections of two contiguous customers from different routes, and (d) *Ejection chain* swaps the end portions of two different routes.

The use of LR ensures the partial optimality of most solutions from the routing perspective. The reason is that, since we are considering a relatively small number of customers per route, the proposed approach can quickly find the optimal solution to most TSP instances. In effect, the respective lower bounds (LB) and upper bounds (UB) converge rapidly, keeping their gap between 0 and 10^{-10}, which guarantees the solution optimality. In addition, LR solves all routes in negligible times. Thus, LR is an efficient alternative for intra-route optimization processes and avoids defining intra-route moves.

4.4 Pseudo-code for the Lagrangian Relaxation procedure

The LR-based method is used within the local search process to solve the routing problem to optimality. It can be considered a specification of the Lagrangian Metaheuristic presented in Boschetti and Maniezzo [5]. As mentioned in Section 3.3, it combines the SO algorithm with a heuristic aiming at improving algorithm's convergence to the optimum. The stopping criterion is based on the maximum number of iterations ($k < max_{\text{iterations}}$) and also on a floating-point exception ($\lambda_k < 10^{-10}$). The applied LR-based method procedure is shown in Table 3.

The proposed LR relaxes the constraint set requiring that all customers must be served by weighting them with a multiplier vector u, since all subcycles can be avoided constructing the solution x as a 1-tree [23]. The Lagrangian Dual problem obtained is $\max_{u \in R^n} L(u)$ where $L(u) = \min_{x1-tree} \sum_{e \in E_v} c_e x_e + \sum_{i \in I_v} u_i (2 - \sum_{e \in \delta(i)} x_e)$.

The proposed heuristic to improve the UB is applied when the solution is nearly a Hamiltonian route (step 8), i.e., the solution has few vertices without two incident edges. This heuristic replaces an edge $e = (i, j)$, where j has some extra edges, for an edge $e = (i, l)$, where l has one single edge. Before applying the exchange, the heuristic checks if the new solution is a 1-tree. Otherwise, the heuristic can obtain an unconnected subtree.

Table 3 The Applied LR-based Algorithm

Initialization
Initialize parameters $u^0 = 0$; $\delta_0 = 2$; $\alpha_L = 1/3$
Obtain an UB applying Nearest Neighbor Heuristic
Initialize $\overline{L} = L(u^0) + \alpha_L(UB - L(u^0))$
Iteration k, repeat until the stopping condition is met:
Solve the Lagrangian function $L(u^k)$
Check the subgradient $\gamma_i^k = 2 - \sum_{e \in \delta(i)} x_e$
if $\parallel \gamma^k \parallel^2 = 0$ then Optimal solution is found $\Rightarrow$ EXIT
if $\parallel \gamma^k \parallel^2 < \xi$ then apply a heuristic to improve the UB
Check the parameter $\overline{L}$
Calculate the step-size $\lambda_k = \delta_k \frac{\overline{L} - L(u^k)}{\parallel \gamma_k \parallel^2}$
Update the multiplier $u^{k+1} = u^k + \lambda_k \gamma^k$
$k \leftarrow k + 1$

A good estimation of ξ will avoid increasing the computation time excessively. First, its value may be large, but it should be updated whenever a feasible solution is found according to $\xi = \parallel \gamma^k \parallel^2$. If this parameter is not correctly updated, the heuristic becomes time consuming. Eventually, the heuristic could find the optimal solution without detecting it, so the method would continue iterating until $LB = UB$.

As mentioned, the convergence of the algorithm is critically influenced by the step-size λ_k. This value relies on either the LB or the UB, which are normally unknown or bad estimated. Therefore, convergence may not be assured for all cases. In order to overcome this limitation, a parameter $\overline{L}$, such that $LB \leq \overline{L} \leq UB$, is introduced. By definition, this parameter corresponds to a better estimation of the optimum L^* than those obtained for LB and UB. The calculation of the step-size turns into:

$$\lambda_k = \delta_k \frac{\overline{L} - L(u^k)}{\parallel \gamma_k \parallel^2} \tag{1}$$

Convergence is guaranteed if the term $\overline{L} - L(u^k)$ tends to zero. In turn, convergence efficiency can be improved as long as the new $\overline{L}$ parameter gets closer to the (unknown) optimal solution. Finally, the parameter δ_k is initialized to the value 2 and is updated as suggested by Zamani and Lau [49].

5 Computational results

The methodology described in this paper has been implemented in Java and linked to the open-source CP software system ECLiPSe 6.0 [3]. All tests have been performed on a dedicated server with an Intel i5 processor at 2.66GHz and 16GB RAM. A total of 91 classical CVRP benchmark instances available at www.branchandcut.org have been used to test the efficiency of the proposed approach when dealing with this simple (in terms of constraints) but extensively tested scenario. In order to ensure fulfillment of the triangular inequality property, only those instances using Euclidean metrics have been selected. The selected problems also include 7 instances from [7] (denoted in tables as C1–C5, C11, and C12) for further comparison with some recent metaheuristics.

As the algorithm has been designed to be run in a parallel computing environment, a test has been done over the set A of benchmark problems to determine the most suitable number of simultaneous threads. This parameter is to be fixed mainly according to computer's characteristics. In the particular server used in this work, up to 4 threads may be executed in parallel in order to keep a reasonable computational efficiency. In the performed test, adopting a parallelized approach permits reducing the total computation time significantly. In particular, for problems from the set A, the total computation time is 41% lower, on average, than the total time spent using a sequential approach. For this reason, all results presented in this paper correspond to a Multi-Start VND implementation with 4 parallel processes, since this approach has demonstrated to keep a reasonable balance between the time spent on calculating one single solution and the total execution time.

5.1 Discussion of results

Table 4 shows results obtained for some representative problems from the selected benchmark sets. Due to algorithm's probabilistic behavior, the final solutions' quality depends on the total number of threads. For this reason, 100 total tasks have been considered for each problem, i.e., 100 pseudo-optimal solutions have been generated for each benchmark instance. Table 4 summarizes information regarding the best solution found (OBS) for each problem, as well as the time required to reach this solution. These results are compared to the best known solutions (BKS) so far. Most sources give these values as integer numbers, obtained by rounding costs, except for the problems from [7] where real values are usually given. From the detailed integer solutions, real costs have been calculated and reported. It should be remarked that the real cost of an integer optimal solution might not correspond to the optimal solution considering real costs. For this reason, negative gaps appear on this table. Thus, it can be deduced that the Multi-Start VND is able to match, and in many cases improve, the real value associated to the best known integer solutions. Concretely, the presented approach has been able to improve 23 best known solutions, considering real costs, out of the 91 tested instances. In addition, the gap is kept reasonably low for all considered instances, being the average gap 0.65%. It remains lower, 0.17%, for the problems selected in Table 4, which include most of the largest instances.

Furthermore, it should be remarked that these results have been obtained in competitive times even for some large instances. As shown in Table 4, most solutions for small problems are obtained in less than a second, while larger instances require higher yet reasonable computational times. In most cases, higher times are closely related to higher quality solutions, i.e., solutions with a negative gap.

These results are similar to other state-of-the-art metaheuristics. Table 5 provides a comparison between the proposed approach and some recent publications. The first two selected metaheuristics correspond to the previous work by the authors: a hybrid VNS (HVNS) presented in [20] and the randomized Clarke and Wright Savings (SR-GCWS) algorithm by [29]. The next three metaheuristics are a hybrid algorithm of Simulated Annealing and Tabu Search (SA-TS) introduced in [32], a hybrid Electromagnetism-like heuristic (HEMA) proposed by [47], and a Particle Swarm algorithm (SR-2) described in [25]. Most publications only report results corresponding to the 14 instances from [7]. For this reason, few results corresponding

Table 4 Results for 50 classical benchmark instances

Problem	# Nodes	BKS	OBS	Gap (%) BKS-OBS	# Routes	Time (s)
A-n32-k5	31	787.81	787.08	−0.09	5	0.633
A-n33-k5	32	662.76	662.11	−0.10	5	0.842
A-n33-k6	32	742.83	742.69	−0.02	6	0.480
A-n37-k5	36	672.59	673.59	0.15	5	1.948
A-n37-k6	36	952.22	950.85	−0.14	6	1.631
A-n38-k5	37	734.18	733.95	−0.03	5	2.546
A-n45-k6	44	944.88	944.88	0.00	6	1.622
A-n46-k7	45	918.46	918.13	−0.04	7	2.062
A-n54-k7	53	1,171.78	1,171.78	0.00	7	4.007
A-n55-k9	54	1,074.46	1,076.85	0.22	9	5.544
A-n63-k9	62	1,622.14	1,622.14	0.00	9	8.073
B-n31-k5	30	676.76	676.09	−0.10	5	0.657
B-n34-k5	33	791.24	789.84	−0.18	5	0.497
B-n35-k5	34	956.29	958.94	0.28	5	1.174
B-n38-k6	37	809.45	809.45	0.00	6	1.211
B-n39-k5	38	553.27	553.16	−0.02	5	1.577
B-n43-k6	42	747.54	746.98	−0.07	6	1.520
B-n45-k5	44	755.43	753.96	−0.19	5	1.011
B-n50-k7	49	744.78	744.23	−0.07	7	1.721
B-n50-k8	49	1,316.20	1,319.53	0.25	8	7.069
B-n51-k7	50	1,035.71	1,037.54	0.18	7	597.915
B-n57-k9	56	1,603.63	1,604.88	0.08	9	7.653
B-n64-k9	63	869.32	868.31	−0.12	9	287.953
E-n22-k4	21	375.28	375.28	0.00	4	0.337
E-n23-k3	22	568.56	568.56	0.00	3	0.422
E-n33-k4	32	838.72	837.67	−0.13	4	0.819
E-n51-k5 (C1)	50	524.61	527.98	0.64	5	17.164
E-n76-k10 (C2)	75	835.26	843.49	0.99	10	28.941
E-n101-k8 (C3)	100	826.14	841.16	1.82	8	195.271
F-n45-k4	44	724.57	727.75	0.44	4	4.459
F-n135-k7	134	1,170.65	1,179.09	0.72	7	630.427
G-n262-k25	261	5,685.00	5,722.00	0.65	25	1,651.360
M-n101-k10 (C12)	100	819.81	821.40	0.19	10	51.395
M-n121-k7 (C11)	120	1,042.11	1,045.14	0.29	7	137.553
M-n151-k12 (C4)	150	1,028.42	1,052.52	2.34	12	834.642
M-n200-k17 (C5)	199	1,291.45	1,324.91	2.59	17	243.789
P-n16-k8	15	451.95	451.95	0.00	8	0.019
P-n19-k2	18	212.66	212.66	0.00	2	0.243
P-n20-k2	19	217.42	217.42	0.00	2	0.148
P-n21-k2	20	212.71	212.71	0.00	2	0.275
P-n22-k2	21	217.85	217.85	0.00	2	0.277
P-n23-k8	22	531.17	531.17	0.00	8	1.447
P-n40-k5	39	461.73	461.73	0.00	5	6.189
P-n45-k5	44	512.79	512.79	0.00	5	10.016
P-n50-k7	49	559.86	560.15	0.05	7	5.155
P-n51-k10	50	742.48	742.36	−0.02	10	5.156
P-n55-k10	54	697.81	698.00	0.03	10	5.331
P-n55-k8	54	592.17	581.17	−1.86	7	14.703
P-n76-k5	75	635.04	633.32	−0.27	5	92.627
P-n101-k4	100	692.28	693.54	0.18	4	839.622
Average				0.17		

Table 5 Comparison between the proposed algorithm and other approaches for some classical benchmark instances

Problem	BKS	HVNS		SR-GCWS		SA-TS		HEMA		SR-2		Our approach	
		BS	t (s)	BS	t (s)	BS	t (s)	BS	t (s)	BS	t (s)	BS	t (s)
A-n32-k5	787.81	787.81	96.6	787.08	6.0	–	–	–	–	–	–	787.08	0.6
A-n33-k5	662.76	662.76	42.7	662.11	2.0	–	–	–	–	662.76	13.0	662.11	0.8
A-n33-k6	742.83	742.83	34.9	742.69	3.0	–	–	–	–	–	–	742.69	0.5
A-n37-k6	952.22	952.22	59.7	–	–	–	–	–	–	–	–	950.85	1.6
A-n38-k5	734.18	735.22	69.0	733.95	7.0	–	–	–	–	–	–	733.95	2.6
A-n45-k6	944.88	951.77	141.7	944.88	31.0	–	–	–	–	–	–	944.88	1.6
A-n46-k7	918.46	919.20	265.9	–	–	–	–	–	–	918.46	23.0	918.13	2.1
A-n63-k9	1,622.14	1,637.58	1,254.3	1,622.14	145.0	–	–	–	–	–	–	1,622.14	8.1
B-n31-k5	676.76	–	–	676.09	1.0	–	–	–	–	–	–	676.09	0.7
B-n39-k5	553.27	–	–	553.16	17.0	–	–	–	–	–	–	553.16	1.6
B-n45-k5	755.43	–	–	754.22	20.0	–	–	–	–	755.43	20.0	753.96	1.0
B-n50-k7	744.78	–	–	744.23	2.0	–	–	–	–	–	–	744.23	1.7
E-n33-k4	838.72	–	–	837.67	7.0	–	–	–	–	–	–	837.67	0.8
E-n51-k5 (C1)	524.61	–	–	524.61	32.0	524.61	38.1	524.61	13.0	524.61	24.0	527.98	17.2
E-n76-k10 (C2)	835.26	–	–	835.26	21.7	835.26	118.3	849.77	19.0	844.42	57.0	843.49	28.9
E-n101-k8 (C3)	826.14	830.53	40,888.9	–	–	826.14	293.3	844.72	41.0	829.40	101.0	841.16	195.3
M-n101-k10 (C12)	819.81	841.37	12,897.9	819.56	338.0	819.56	316.0	847.56	190.0	819.56	88.0	821.40	51.4
M-n121-k7 (C11)	1,042.11	1,092.13	39,383.9	1,043.88	74.5	1,045.50	332.8	1,042.11	319.0	1,052.34	93.0	1,045.14	137.6
M-n151-k12 (C4)	1,028.42	1,035.80	76,933.9	–	–	1,038.71	701.4	1,059.03	132.0	1,048.89	223.0	1,052.52	834.6
M-n200-k17 (C5)	1,291.45	1,325.07	195,727.3	–	–	1,311.70	1,844.3	1,302.33	201.0	1,323.89	413.0	1,324.91	243.8
P-n20-k2	217.42	217.42	0.9	217.42	41.0	–	–	–	–	–	–	217.42	0.2
P-n22-k2	217.85	217.85	4.7	217.85	9.0	–	–	–	–	–	–	217.85	0.3
P-n51-k10	742.48	742.48	427.3	741.50	19.0	–	–	–	–	–	–	742.36	5.2
P-n55-k8	592.17	580.96	247.7	–	–	–	–	–	–	–	–	581.17	14.7
P-n76-k5	635.04	633.32	10,784.6	633.32	73.0	–	–	–	–	–	–	633.32	92.6
Average gap (%)		0.61		–0.05		0.41		1.60		0.69		0.23	

to other problem sets are reported for the latter three metaheuristics in Table 5. Moreover, some of the instances from [7] include an additional constraint on the maximum route length that is not handled in the proposed version of the algorithm. Therefore, results for these instances have been omitted in this table.

It may be observed that the proposed approach is comparable in terms of quality and computational efficiency to these recent metaheuristics. Times needed by our approach to reach a pseudo-optimal solution are in most cases lower than those required by means of the other algorithms. It should be remarked that the proposed approach clearly improves the efficiency of the previous algorithms HVNS and SR-GCWS. Furthermore, the Multi-Start VND provides the lowest gap among all selected metaheuristics, only beated by the SR-GCWS approach. However, most of the higher gaps obtained with the proposed approach correspond to some of the largest instances, whose results are not reported for the SR-GCWS algorithm.

As a final remark, it can be observed that the lowest gap (−1.86%) corresponds to the problem P-n55-k8, where a solution considering only seven vehicles (# routes) has been obtained. Although the best known solution for this problem uses eight vehicles, feasible solutions with seven vehicles and lower costs may be reached, as the one obtained with this approach. However, if only seven vehicles are considered, the Multi-Start VND has finished slightly over the value 580.96 (576 considering integer costs), published for this problem in [1, 2, 20].

6 Scope and limitations of our approach

The described hybrid algorithm embeds CP and LR within the VND metaheuristics framework by decomposing the CVRP into two subproblems concerning customers' allocation and routing optimization separately. A fast and efficient algorithm such as the RCWS is used to feed the multi-start scheme by generating good initial solutions. Thus, the proposed optimization approach implements a flexible, efficient and robust optimization algorithm able to deal with some realistic problems, which means both the ability to tackle large instances and to represent real operational constraints. The characteristics of the resulting algorithm can be explained in the following way: flexibility involves the quality of the algorithm to be adapted to real problems; efficiency is related to the easiness of the algorithm to obtain optimal or quasi-optimal solutions in reasonable computation times; and robustness is related to the fact that the algorithm performs well even when no extense fine-tuning processes are carried out on its parameters.

Regarding flexibility, this approach benefits from the CP capabilities to model different operational constraints. This constraints are present in most of the real application cases and, in general, affect to the allocation decisions. CP, which is not restricted by modeling limitations such as constraint linearity, facilitates the representation of allocation constraints without requiring any specific action on the solving method. Hence, the hybrid scheme can be easily adapted to different CVRP variants by simply adding the allocation constraints which properly model the feasible solutions of the problem. Since the VND optimization scheme is able to reach feasible solutions starting from non-feasible initial solutions, e.g. not fulfilling the maximum number of vehicles [19], the RCWS algorithm does not need to be modified in order to include operational constraints other than capacities. However, other capacity-like constraints, such as total driving time of each route, can be translated to a capacity constraint in order to obtain feasible solutions by means of the RCWS algorithm. Additional operational constraints may be added to the CP model, which will ensure solutions' feasibility along the local search process. Thus, this hybrid approach will be able to tackle complex instances related to real application cases by adding little modifications into the problem modeling, but neither into the optimization scheme nor algorithms.

The efficiency of the proposed algorithm is supported by the results presented in the previous section. As discussed, the presented approach is able to match the best known solutions for benchmark problems of different sizes in reasonable computation times. The provided comparison proves that its efficiency is similar to other state-of-the-art metaheuristics, both in terms of time and solutions' quality.

The robustness of the algorithm is a consequence of the light requirements for fine tuning. The LR-based algorithm does not require any specific adjustment since all the convergence parameters are self-tuned. The CP-based subproblem depends just on the quality of the defined constraint model to properly describe the feasible solutions. The RCWS does not require any adaptation either. Only the VND movements could require different prioritization depending on the problem being solved in order to get a better solution quality.

Facing other relevant VRP variants, such as those involving Pick-Up & Delivery or Time-Windows, would imply the modification of the LR-based method (in addition to the constraint model modification) and the implementation of new neighborhoods in the VND metaheuristic. The RCWS-based algorithm should be also adapted and would require a proof of its efficiency. Authors are currently working on these new implementations.

7 Conclusions

This paper has presented a hybrid methodology which combines a randomized version of the CWS heuristic with Constraint Programming and Lagrangian Relaxation to efficiently solve CVRP instances. These techniques have been embedded into a Multi-Start Variable Neighborhood Descent framework. According to the tests performed, the proposed algorithm is competitive with state-of-the-art metaheuristics.

In the proposed approach, the CVRP has been decomposed into two separate subproblems, where CP and LR techniques are combined to ensure capacity constraints fulfillment and calculate all involved routes. This approach allows reducing the computation time during local search processes, since problems to be solved are far less complex than the original CVRP. The randomized CWS algorithm is used to quickly provide several "good" initial solutions to start the search in a multi-start environment. This algorithm has shown to be an efficient alternative to other existing approaches due to its capability to generate quasi-optimal solutions in a reasonable time. In addition, it is a robust algorithm, since it is almost parameter-free and only requires a light fine tuning. Finally, it should be noticed that due to its modular design, the proposed approach is flexible and can be easily adapted to solve other VRP with additional constraints or multi-criteria objective functions.

Acknowledgements This work was supported in part by the IN3-UOC Knowledge Community Program under grant HAROSA09. It has also been supported by the Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) of the Generalitat de Catalunya (project 2009 SGR 629).

References

1. Alba, E., Dorronsoro, B.: A hybrid cellular genetic algorithm for the capacitated vehicle routing problem. In: Abraham, A., Grosan, C., Pedrycz, W. (eds.) Engineering Evolutionary Intelligent Systems (Studies in Computational Intelligence, 82), pp. 379–422. Springer, New York (2008)
2. Altinel, I., Oncan, T.: A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. J. Oper. Res. Soc. **56**, 954–961 (2005)
3. Apt, K., Wallace, M.: Constraint Logic Programming using ECLiPSe. Cambridge University Press, Cambridge (2007)
4. Berger, J., Barkaoui, M.: A hybrid genetic algorithm for the capacitated vehicle routing problem. In: Cantó-Paz, E. (ed.) Proceedings of the International Genetic and Evolutionary Computation Conference, Chicago, IL, USA, pp. 646–656. Springer, New York (1996)
5. Boschetti, M., Maniezzo, V.: Benders decomposition, Lagrangean Relaxation and metaheuristic design. Journal of Heuristics **15**, 283–312 (2009)
6. Buxey, G.: The Vehicle Scheduling Problem and Monte Carlo Simulation. J. Oper. Res. Soc. **30**, 563–573 (1979)
7. Christofides, N., Mingozzi, A., Toth, P.: The Vehicle Routing Problem. In: Combinatorial Optimization, pp. 315–338. Wiley (1979)
8. Clarke, G., Wright, J.: Scheduling of vehicles from a central depot to a number of delivering points. Oper. Res. **12**, 568–581 (1964)
9. Cordeau, J., Laporte, G., Savelsbergh, M., Vigo, D.: Vehicle routing. In: Barnhart, C., Laporte, G. (eds.) Handbook in Operations Research and Management Science, vol. 14, pp. 367–428. Elsevier, Amsterdam (2007)
10. Dantzig, G., Ramser, J.: The truck dispatching problem. Manage. Sci. **6**, 80–91 (1959)
11. Faulin, J., Juan, A.A.: The algacea-1 method for the capacitated Vehicle Routing Problem. Int. Trans. Oper. Res. **15**, 1–23 (2008)
12. Faulin, J., Gilibert, M., Juan, A.A., Ruiz, R., Vilajosana, X.: Sr-1: a simulation-based algorithm for the capacitated Vehicle Routing Problem. In: Proceedings of the 2008 Winter Simulation Conference (2008)
13. Feo, T., Resende, M.: Greedy randomized adaptive search procedures. J. Glob. Optim. **6**, 109–133 (1995)
14. Fernández, P., García, L., Mayado, A., Sanchís, J.: A real delivery problem dealt with Monte Carlo techniques. Top **8**, 57–71 (2000)
15. Festa, P., Resende, M.: An annotated bibliography of grasp—part I: algorithms. Int. Trans. Oper. Res. **16**, 1–24 (2009)
16. Fisher, M.: The Lagrangean Relaxation method for solving integer programming problems. Manage. Sci. **27**, 1–18 (1981)
17. Gaskell, T.: Bases for the vehicle fleet scheduling. Oper. Res. Q. **18**, 281–295 (1967)
18. Gendreau, M., Hertz, A., Laporte, G.: A tabu search heuristic for the Vehicle Routing Problem. Manage. Sci. **40**, 1276–1290 (1994)
19. Guimarans, D., Herrero, R., Riera, D., Juan, A., Ramos, J.: Combining Constraint Programming, Lagrangian Relaxation and probabilistic algorithms to solve the Vehicle Routing Problem. In: Proceedings of the 17th RCRA International Workshop. Bologna, Italy (2010)
20. Guimarans, D., Herrero, R., Ramos, J., Padrón, S.: Solving vehicle routing problems using Constraint Programming and Lagrangean Relaxation in a metaheuristics framework. Int. J. Inf. Syst. Supply Chain Manage. **4**(2), 61–81 (2011)
21. Hansen, P., Mladenovic, N.: A tutorial on variable neighborhood search. Tech. Rep. G-2003-46, Groupe d'Études et de Recherche en Analyse des Décisions (GERAD), Montreal, Canada. URL http://www.gerad.ca/fichiers/cahiers/G-2003-46.pdf (2003)
22. Hasle, G., Kloster, O.: Industrial vehicle routing. In: Hasle, G., Lie, K., Quak, E.: (eds.) Geometric Modelling, Numerical Simulation, and Optimization, pp. 397–435. Springer, Berlin (2007)
23. Held, M., Karp, R.: The travelling salesman problem and minimum spanning trees: part II. Math. Program. **1**, 6–25 (1971)

24. Herrero, R., Ramos, J., Guimarans, D.: Lagrangean metaheuristic for the travelling salesman problem. In: Extended Abstracts of Operational Research Conference 52. Royal Holloway, University of London (2010)
25. Jin Ai, T., Kachitvichyanukul, V.: Particle swarm optimization and two solution representations for solving the capacitated Vehicle Routing Problem. Comput. Ind. Eng. **56**, 380–387 (2009)
26. Juan, A., Faulin, J., Jorba, J., Grasman, S., Barrios, B.: Sr-2: a hybrid intelligent algorithm for the Vehicle Routing Problem. In: Proceedings of the 8th International Conference on Hybrid Intelligent Systems, pp. 78–83. IEEE Computer Society, Barcelona (2008)
27. Juan, A., Faulin, J., Ruiz, R., Barrios, B., Gilibert, M., Vilajosana, X.: Using oriented random search to provide a set of alternative solutions to the capacitated vehicle routing problem. In: Operations Research and Cyber-Infrastructure, pp. 331–346 (2009)
28. Juan, A., Faulin, J., Jorba, J., Riera, D., Masip, D., Barrios, B.: On the use of Monte Carlo Simulation, cache and splitting techniques to improve the Clarke and Wright savings heuristics. J. Oper. Res. Soc. **62**, 1085–1097 (2011)
29. Juan, A., Faulin, J., Ruiz, R., Barrios, B., Caballe, S.: The SR-GCWS hybrid algorithm for solving the capacitated Vehicle Routing Problem. Appl. Soft Comput. **10**(1), 215–224 (2010)
30. Laporte, G.: What you should know about the vehicle routing problem. Nav. Res. Logist. **54**, 811–819 (2007)
31. Law, A.: Simulation Modeling & Analysis. McGraw-Hill, New York (2007)
32. Lin, S., Lee, Z., Ying, K., Lee, C.: Applying hybrid meta-heuristics for capacitated vehicle routing problem. Expert Syst. Appl. **2**(36), 1505–1512 (2009)
33. Mester, D., Bräysy, O.: Active-guided evolution strategies for the large-scale capacitated Vehicle Routing Problems. Comput. Oper. Res. **34**, 2964–2975 (2007)
34. Mladenovic, N., Hansen, P.: Variable neighborhood search. Comput. Oper. Res. **24**(11), 1097–1100 (1997)
35. Nagata, Y.: Edge assembly crossover for the capacitated Vehicle Routing Problem. Lect. Notes Comp. Sci. **4446**, 142–153 (2007)
36. Prins, C.: A simple and effective evolutionary algorithm for the Vehicle Routing Problem. Comput. Oper. Res. **31**, 1985–2002 (2004)
37. Reinelt, G.: The Traveling Salesman: computational solutions for TSP applications. Lecture Notes in Computer Science, vol. 840 (1994)
38. Resende, M.: Metaheuristic hybridization with Greedy Randomized Adaptive Search Procedures. In: Tutorials in Operations Research, pp. 295–319 (2008)
39. Rossi, F., van Beek, P., Walsh, T. (eds.): Handbook of Constraint Programming. Elsevier, Amsterdam (2006)
40. Rousseau, L., Gendreau, M., Pesant, G.: Using constraint-based operators to solve the Vehicle Routing Problem with time windows. Journal of Heuristics **8**, 43–58 (2002)
41. Savelsbergh, M.: Local search in routing problems with time windows. Ann. Oper. Res. **4**, 285–305 (1985)
42. Savelsbergh, M.: Computer Aided Routing. Tech. rep., Centrum voor Wiskunde en Informatica (1988)
43. Taillard, E.: Parallel iterative search methods for vehicle routing problems. Networks **23**, 661–673 (1993)
44. Tarantilis, C., Kiranoudis, C.: Boneroute: an adaptative memory-based method for effective fleet management. Ann. Oper. Res. **115**, 227–241 (2002)
45. Toth, P., Vigo, D.: The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia (2002)
46. Toth, P., Vigo, D.: The granular tabu search and its application to the vehicle routing problem. INFORMS J. Comput. **15**, 333–346 (2003)
47. Yurtkuran, A., Emel, E.: A new hybrid electromagnetism-like algorithm for capacitated routing problems. Expert Syst. Appl. **37**(4), 3427–3433 (2010). doi:10.1016/j.eswa.2009.10.005
48. Zachariadis, E., Kiranoudis, C.: A strategy for reducing the computational complexity of local search-based methods for the Vehicle Routing Problem. Comput. Oper. Res. **37**, 2089–2105 (2010)
49. Zamani, R., Lau, S.: Embedding learning capability in Lagrangean Relaxation: an application to the travelling salesman problem. Eur. J. Oper. Res. **201**(1), 82–88 (2010)