

This item is the archived peer-reviewed author-version of:

The **k**-dissimilar vehicle routing problem

Reference:

Talarico Luca, Sörensen Kenneth, Springael Johan.- The **k**-dissimilar vehicle routing problem European journal of operational research - ISSN 0377-2217 - 244:1(2015), p. 129-140 Full text (Publisher's DOI): https://doi.org/10.1016/J.EJOR.2015.01.019 To cite this reference: http://hdl.handle.net/10067/1249950151162165141

uantwerpen.be

Institutional repository IRUA

The k-dissimilar vehicle routing problem

L. Talarico*, K. Sörensen, J. Springael

University of Antwerp Operations Research Group ANT/OR Prinsstraat 13, 2000 Antwerp, Belgium

July 2014

In this paper we define a new problem, the aim of which is to find a set of k dissimilar solutions for a vehicle routing problem (VRP) on a single instance. This problem has several practical applications in the cash-in-transit sector and in the transportation of hazardous materials. A min-max mathematical formulation is proposed which requires a maximum similarity threshold between VRP solutions, and the number k of dissimilar VRP solutions that need to be generated. An index to measure similarities between VRP solutions is defined based on the edges shared between pairs of alternative solutions. An iterative metaheuristic to generate k dissimilar alternative solutions is also presented. The solution approach is tested using large and medium size benchmark instances for the capacitated vehicle routing problem.

Keywords: Decision support systems, Combinatorial optimization, Metaheuristic, Logistics, Risk management.

1. Introduction

In many European countries, cash-in-transit companies must by law determine several alternative routes for each of their vehicles when transporting cash. The aim of this measure is to allow a firm to easily change its plans in case of unforeseen circumstances (e.g., accidents, road works) and to increase security by making the vehicle routes more unpredictable. In this paper, we define a new vehicle routing problem — the *k*-dissimilar vehicle routing problem or *k*d-VRP — to support this optimization problem.

A solution of this novel problem consists of k feasible solutions of a single capacitated vehicle routing problem (VRP). Each of these VRP solutions (which we will consistently call *alternative solutions*) must obey the traditional constraints of the VRP: all customers are visited exactly

^{*}Corresponding author: University of Antwerp, Prinsstraat 13, 2000 Antwerp, Belgium, Tel:+3232654177, Email:luca.talarico@uantwerpen.be

once, all vehicles begin and end at the depot, and the capacity of the vehicle is not exceeded. The quality of an alternative solution is measured as the total distance travelled by all vehicles.

Assuming that a similarity metric can be calculated between any pair of alternative solutions, a feasible solution to the kd-VRP is a set of k feasible alternative solutions for which the difference between each pair of alternative solutions is larger than a certain threshold. The objective of the kd-VRP is to minimize the cost of the worst alternative solution in the set. To the best of our knowledge, this problem has never been studied before in the literature.

The *k*d-VRP is closely related to the *m*-peripatetic vehicle routing problem (m-PVRP) studied in Ngueveu et al. (2010a,b). This problem consists in finding a set of edge-disjoint routes of minimal total cost over *m* periods so that each customer is visited exactly once per period and the edge between a pair of customers can be used at most once during the *m* periods. "Periods" in the *m*-PVRP are essentially the same concept as "alternative solutions" in the *k*d-VRP. The difference between the *k*d-VRP and the *m*-PVRP is twofold. In a feasible solution of the *m*-PVRP, no edge is used twice, while in the *k*d-VRP the multiple usage of an edge is not explicitly forbidden, but rather limited by a constraint on the similarity between alternative solutions. Second, the *m*-PVRP minimizes the total cost over all periods, while the *k*d-VRP minimizes the worst-case cost over all alternative solutions. The motivation for the *k*d-VRP is that for some real-life applications (e.g., money collection, transportation of hazardous materials), the constraint that imposes *k* edge-disjoint VRP solutions in which not a single edge is shared between the alternative solutions might be too stringent. The min-max objective function of the *k*d-VRP generalizes the *m*-PVRP.

Figure 1 shows an example of a solution for the kd-VRP with k = 3, for which a set containing three dissimilar, but not disjoint, alternative solutions (see Figures 1(a), 1(b) and 1(c)) has been generated.



Figure 1: A solution for the *k*d-VRP with k = 3, instance F-n45-k4

The *k*d-VRP has many practical applications. A first application, which provided the motivation for this work, can be found in the context of money collection and distribution, also known

as the *cash-in-transit* sector. In this context, a set of customers (e.g., banks, shops, casinos, jewellers) needs to be visited to pick up valuables and cash. As mentioned, companies in the cash-in-transit sector are often required by law (see e.g., the Belgian law reported in Service Public Federal Interieur (2003) for more details) to determine several alternative solutions in advance. Additionally, the same alternative solution cannot be used more than two consecutive times. On the other hand, the travel costs of each of these alternative solutions should be minimized for obvious economic reasons. The aim is thus to decrease the predictability of the chosen alternative solution, reducing the risk to be assaulted, while maintaining economic viability.

A second application can be found in the transportation of dangerous materials (e.g., the routing of tankers which must serve a set of petrol stations). In this case a limitation of the number of shared edges between the routes might be required in order to better share and mitigate the risk of accidents and spread the risk equitably among the geographical regions crossed by vehicles along different routes (Gopalan et al., 1990). In addition, supposing that the probability of an accident increases considerably in case of adverse weather conditions and/or depending on the type of road segment, it is crucial to have several dissimilar routing plans to reduce the risk of potential fatalities (Akgün et al., 2000; Sattayaprasert et al., 2008).

A third application concerns the design of patrol routes for security agents who must follow partially different routes over time (Wolfler Calvo and Cordone, 2003). The design of (spatial and/or temporal) dissimilar patrol routes, that are not necessarily disjoint, might help companies to guarantee high security standards, minimizing at the same time their operative costs.

The contributions of this paper are the following: (1) A new NP-hard combinatorial optimization problem is proposed, the *k*-dissimilar vehicle routing problem or *k*d-VRP. The *k*d-VRP requires a similarity measure that can calculate the difference between alternative solutions. We define such an index, starting from the similarity index used for the dissimilar *k*-shortest path problem. (2) A mathematical formulation for the *k*d-VRP is presented. (3) A metaheuristic to solve medium and large instances of the *k*d-VRP is described, implemented and tested.

The remainder of the paper is organized as follows. In Section 2, the literature on the peripatetic VRP and similar problems such as the dissimilar k paths problem and the disjoint paths problem is presented. In Section 3, some indices used to measure the (dis)similarity between paths and alternative solutions are introduced. In Section 4, the k-dissimilar vehicle routing problem (kd-VRP) is described in detail and a mathematical formulation is developed. In Section 5, an iterative metaheuristic to find solutions for the kd-VRP is developed. In Section 6 the solution approach is tested using 51 benchmark instances from the VRP library. Finally, Section 7 concludes the paper and provides some suggestions for future research.

2. Literature review

The problem of finding dissimilar solutions within the transportation sector has received some attention in the literature on shortest path problems and, more recently, on vehicle routing

problems.

The first attempt to introduce the concept of similar solutions within shortest path problems is the *k*-shortest paths problem (*k*-SPP) in which the shortest, the second shortest, until the *k*-th shortest path from an origin to a destination node are sought, in increasing order of length (see, e.g., Yen (1971); Hershberger et al. (2007); Gotthilf and Lewenstein (2009); Di Puglia Pugliese and Guerriero (2013)). However, the *k* shortest paths are likely to share a large number of edges, and tend to be very similar to each other. For some applications in which dissimilar alternatives are needed, a different approach is required.

To find dissimilar paths, the disjoint-path problem (DPP), introduced by Suurballe (1974), can be used. Differently from the classical k shortest paths problem, the DPP allows no common edges (edge-disjoint paths) or shared vertices (vertex-disjoint paths) among alternative paths. However, for many real-life applications (e.g., hazardous material transportation, couriers, routing in congested network) the constraint imposing no edges in common among paths may be too stringent and, due to the impossibility to reuse all the shortest edges employed in the previous solutions, the resulting disjoint paths may be impractically long (Kuby et al., 1997). In fact, in the majority of real-life transportation applications, in which a minimum number of dissimilar solutions, not necessarily disjoint, is required, the cost of each alternative should be as small as possible. Therefore, a valid alternative to the DPP is represented by the path dissimilarity problem (PDP) in which a set of dissimilar solutions with minimum costs are generated. The PDP is a bi-objective routing problem in which a set of k paths, from an origin to a destination, must be generated with a minimum length and a maximum dissimilarity (see, e.g., Akgün et al. (2000) and Dell'Olmo et al. (2005)).

In the PDP a set of k alternative paths from an origin to a destination is generated, by using specific indices to measure the similarity between alternative solutions. In Park et al. (2002) dissimilarities among alternative paths are analysed by evaluating, from an individual perspective, multiple attributes (e.g., distance, travel time, variability) associated to the edges. In the domain of hazardous materials transportation, spatially dissimilar paths which minimize the risk of accidents (distributing the risk over all regional zones to be crossed uniformly) need to be obtained. However, for routing hazardous materials, the spatial dissimilarity between alternative paths may depend on how localized the effects of a spill are.

Only few papers in the literature address the issue of finding dissimilar solutions in the context of vehicle routing problems. The goal of this paper is to contribute to this research area by proposing a general model that we named the kd-VRP. It shares several properties with a set of routing problems called *peripatetic*. A solution of the *m*-peripatetic vehicle routing problem (*m*-PVRP) consists of a set of *m* different VRP solutions for which it holds that each pair of alternative solutions is edge-disjoint. In other words, each edge is used at most in one of the *m* VRP solutions. The *m*-PVRP generalizes two well-known NP-hard problems: the vehicle routing problem (VRP) and the *m*-peripatetic salesman problem (*m*-PSP). The latter is a special case of the *m*-PVRP with one single vehicle of infinite capacity.

An approach suggested in the literature for cash-in-transit applications attempts to reduce the risk of being attacked by generating dissimilar solutions. These solutions are most likely to be

"unpredictable" for criminals and can be obtained by solving a specific *m*-PVRP problem. In Ngueveu et al. (2010a,b), customers are visited several times during a fixed planning horizon, but the use of the same arc twice is explicitly forbidden.

In Wolfler Calvo and Cordone (2003); Michallet et al. (2014) the dissimilarity between m-PVRP solutions is enforced by introducing time windows with a minimum and maximum time lag between two consecutive visits of the same customer. In this way it is possible to generate a wide variety of solutions, as required for security reasons. In Yan et al. (2012) a specific application of a m-PVRP to the cash-in-transit sector is proposed aimed at reducing operating costs and ensuring safe cash conveyance. A model, that incorporates a new concept of similarity by considering both temporal and spatial measures, is developed. In this model, dissimilarities between the m vehicle routes are incorporated into the m VRP solutions by imposing different visit times of the same customer during the planning horizon.

The m-PVRP can be extended and generalized by making the constraint that two solutions cannot have any edge in common less stringent. Some attempts in this direction are followed in the literature on VRP applications in which specific indices have been defined and used to measure similarities between solutions.

For example, in Løkketangen et al. (2012), a multiobjective decision support system tool is developed to produce a set of k dissimilar VRP solution. The dissimilarity between the k VRP solutions is based on an attribute distance function. The distance function includes some measures typical of the decision process and comprise, for example, road accessibility, type and amount of load, road length, road quality, vehicle, and driver.

In Sörensen (2006) a multiobjective optimization approach is proposed in order to find a set of k VRP solutions that are "close" (in the solution space) to a given baseline VRP solution and at the same time have a high quality in the sense that their total distance travelled is small. In particular a memetic algorithm with population management is implemented in order to offer to the decision maker a choice of Pareto-optimal solutions, allowing him to make a trade-off between changing his existing solution (i.e., baseline VRP solution) and allowing a longer travel distance.

3. Similarity indices

In order to measure the dissimilarities between solutions several methods and indices have been proposed in the literature on shortest path problems in which the concept of dissimilarity between solutions has been widely studied. The dissimilarity measure, that is used in this paper, is discussed in Section 4 and can be found in Eq. (6).

In Lombard and Church (1993) the concept of "area under the path" is introduced. If the network is assumed to be representable on a plane, the "area under the path" is the area between the path and the *x*-axis. Therefore, the dissimilarity between two paths is measured by the absolute difference between the areas under the paths. In Martí et al. (2009) the dissimilarity $Dis(P_i, P_j)$ between two paths P_i and P_j , is computed as the average of the distances between each vertex in P_i to the path P_j plus the average of the distances between each vertex in P_j to the path P_i . The dissimilarity measure is given by the formula:

$$Dis(P_{i}, P_{j}) = \frac{1}{2} \left[\frac{\sum_{v_{l} \in P_{i}} \phi(v_{l}, P_{j})}{|P_{i}|} + \frac{\sum_{v_{l} \in P_{j}} \phi(v_{l}, P_{i})}{|P_{j}|} \right]$$
(1)

where the value $\phi(v, P_j)$ represents the Euclidean distance from a vertex v to a path $P_j = \{v_1, v_2, \dots, v_n\}$ expressed as:

$$\phi(v, P_j) = \min_{v_l \in P_j} \phi(v, v_l) \tag{2}$$

The similarity index in Eq. (1) considers spatial information hence the resulting paths will be dispersed to each other from a spatial point of view. In Akgün et al. (2000) and Vanhove (2012) the dissimilarity is measured in terms of shared edges between paths, without considering spatial information concerning the physical location of the vertices. The expression to compute the dissimilarity between two paths P_i and P_j , considering only the length (denoted by letter *L*) of the shared edges, is as follows:

$$Dis(P_i, P_j) = 1 - \frac{1}{2} \left[\frac{L(P_i \cap P_j)}{L(P_i)} + \frac{L(P_i \cap P_j)}{L(P_j)} \right]$$
(3)

In Dell'Olmo et al. (2005) the concept of "buffer zone" is included in Eq. (3) to embed spatial information in the measure of similarity. The "buffer zone" is an area determined by moving a circle along the path, whose center is the vehicle on the path itself and whose radius is proportional to the impact area due to a possible accident.

In Thyagarajan et al. (2005) an extension of the dissimilarity measure in Eq. (3) is proposed considering the time context. This is useful in military missions, in which the time difference between routes must be considered. Other works include the temporal dimension into the similarity index (see, e.g., Yan et al. (2012); Wolfler Calvo and Cordone (2003); Michallet et al. (2014)). Supposing that each customer must be visited several times during a given time horizon, the hours of any two visits to the same customer, over the planning horizon, must be spread in the customer's time window.

In order to find k dissimilar alternative solutions a distance measure to calculate the difference (or similarity) between two given solutions is required. Besides measuring the dissimilarity between two alternative solutions based on the number (or the length) of the common edges, the *edit distance* might be adopted (Levenshtein, 1966). The edit distance can be used for permutation problems i.e., problems of which the solutions are most naturally represented as a permutation of a set of items (i.e., problem attributes), representing the order in which the items appear in each solution. For example a VRP solution can be represented as a set of permutations, one for each trip. Each trip is determined by the order in which the customers appear in it. This distance measure is based on the idea that the distance between two solutions is equal to the "cost" required to transform the first solution into the second one.

A comparison between these distance measures is beyond the scope of this paper. For a more elaborate discussion of some issues related to distance measures, including some other distance measures for permutation problems, we refer to Sörensen (2007) and Løkketangen et al. (2012).

4. Problem description

In this section a mathematical formulation of the kd-VRP is proposed, based on an MIP formulation for the VRP. The VRP is seen as a subproblem, the solutions of which are the input of a master problem in which the k dissimilar alternative solutions are selected.

The VRP is defined on a graph G = (V, E) with vertex set $V = \{0, ..., n\}$ and edge set E. Vertices $\{1, ..., n\}$ correspond to the customers, while vertex 0 corresponds to the depot. A non-negative cost d_{ij} is associated with each edge $(i, j) \in E$. The cost structure is assumed to be symmetric, i.e., $d_{ij} = d_{ji} \forall i, j \in V$. Moreover with each customer $i \in V \setminus \{0\}$ a known demand $m_i \ge 0$ is associated, which represents the quantity of goods to be delivered. A set N of identical vehicles, each with capacity C, is available at the depot. It is assumed that $m_i \le C$, $\forall i \in \{1, ..., n\}$.

A general MIP formulation of the VRP (see, e.g., Van Leeuwen and Volgenant (1983) for more details) is presented in Eq. (4a)–(4f). This formulation uses a three-index decision variable x_{ij}^h , which assumes value 1 if edge (i, j) is traversed by vehicle h, and 0 otherwise.

$$\min \sum_{h \in N} \sum_{(i,j) \in E} d_{ij} x_{ij}^h$$
(4a)

s.t.

$$\sum_{j \in V \setminus \{0\}} x_{0j}^h = \sum_{i \in V \setminus \{0\}} x_{i0}^h = 1 \qquad \qquad \forall h \in N$$
(4b)

$$\sum_{h \in N} \sum_{j \in V} x_{ij}^h = \sum_{h \in N} \sum_{j \in V} x_{ji}^h = 1 \qquad \qquad \forall i \in V \setminus \{0\}$$
(4c)

$$\sum_{i \in V} \sum_{j \in V \setminus \{0\}} m_j \, x_{ij}^h \le C \qquad \qquad \forall h \in N \qquad (4d)$$

$$\sum_{h \in N} \sum_{i \in Q} \sum_{j \notin Q} x_{ij}^h \ge 1 \qquad \qquad \forall Q \subset V; Q \neq \emptyset$$
 (4e)

$$x_{ij}^h \in \{0,1\} \qquad \qquad \forall (i,j) \in E; \forall h \in N \qquad (4f)$$

The objective function (4a) minimizes the total travel costs of all vehicles combined. Constraints (4b) force each vehicle to start and finish its route at the depot, visiting at least one vertex along its tour. Constraints (4c) state that every vertex must be visited exactly once, implying that only one vehicle may arrive at a given vertex and depart from it. Constraints (4d) impose a restriction on the maximum load of each vehicle. Finally Constraints (4e) ensure that no subtour occurs in the solution. Constraints (4f) limit the domain of the decision variables. The output of the VRP subproblem in Eq. (4a)–(4f) is the input of the master problem in Eq. (5a)–(5c). The latter requires two more parameters to be set: the value of the similarity threshold T_S and the number of alternative solutions (k > 1). T_S represents the maximum similarity between alternative solutions and it is a parameter of the problem that should be defined as an input by the user. We also define a similarity index $\delta(y_i, y_j)$ that measures the similarity between pairs (y_i, y_i) of VRP solutions.

A mathematical formulation of the *k*d-VRP is:

s.t.

$$\min\max_{i\in\{1,\ldots,k\}} f(y_i) \tag{5a}$$

$$\delta(y_i, y_j) \le T_S \qquad \forall i, j \in \{1, \dots, k\}; i \ne j \qquad (5b)$$

$$y_i \in \Omega \qquad \forall i \in \{1, \dots, k\} \qquad (5c)$$

A feasible solution for the kd-VRP consists of a set $S = \{y_1, \ldots, y_k\}$ containing k alternative solutions $y_1, \ldots, y_k \in \Omega$ with Ω being the set of all the feasible VRP solutions that satisfy constraints in Eq. (4b)–(4f). The objective function (5a) minimizes the cost of the worst alternative solution in S. Constraints (5b) impose that all the pairs of alternative solutions in S are dissimilar. More specifically, similarities among alternative solutions can be at most equal to the threshold T_S . In the remainder of the paper we refer to constraints (5b) as *similarity constraints*. Since the concepts of similarity and dissimilarity are perfectly complementary, for the sake of clarity, from here on, we will adopt the point of view of similarity. The dissimilarity between alternative solutions can be trivially derived from the similarity measure if required. Finally constraints (5c) restrict the domain of the decision variables.

In principle, if the objective function in Eq. (5a) is replaced with the minimization of the total cost of the alternative solutions $(\sum_{i \in \{1,...,k\}} f(y_i))$ then the *k*d-VRP can be considered as the generalization of the *k* disjoint VRP problem. In addition, when the similarity threshold in Eq. (5b) is equal to zero the *k*d-VRP is reduced to a standard *m* peripatetic vehicle routing problem (*m*-PVRP). For this reason, the same considerations about the feasibility of the *m*-PVRP are also valid for the *k*d-VRP. More specifically, for each alternative solution, the total minimal number of routes necessary to satisfy the total demand of customers in $V \setminus \{0\}$ is equal to $\tau = \lceil \sum_{i \in V \setminus \{0\}} \frac{m_i}{C} \rceil$. Given a complete graph, any instance of the *k*d-VRP with *n* customers may admit a feasible solution if $k \leq \frac{n}{2\tau}$. The reader is referred to Ngueveu et al. (2010b) for further details.

Both the mathematical model in Eq. (5a)–(5c) and the solution approach (developed in Section 5) are independent of the specific similarity index used. Hence, all of the similarity indices discussed in Section 2 could be used in both the formulation and the heuristic optimization algorithm.

The similarity index used in this paper is derived from Eq. (3) described in Section 3. This index can be used to measure similarities between alternative solutions by comparing the structure

of the routes contained in these solutions. Given two alternative solutions y_i and y_j , this index compares each route of y_i with each route of y_j . Let $r_{y_i}^l$ and $r_{y_j}^m$ be the *l*-th route of y_i and the *m*-th route of y_j respectively, the similarity between these routes is calculated as follows:

$$\delta(y_i, y_j) = \max_{l,m \in \mathbb{N}} \frac{1}{2} \left[\frac{c_s(r_{y_i}^l, r_{y_j}^m)}{c(r_{y_i}^l)} + \frac{c_s(r_{y_i}^l, r_{y_j}^m)}{c(r_{y_j}^m)} \right]$$
(6)

The term $c_s(r_{y_i}^l, r_{y_j}^m)$ represents the cost of the edges shared between routes $r_{y_i}^l$ and $r_{y_j}^m$, while $c(r_{u_i}^l)$ (or $c(r_{u_i}^n)$) is the cost of route $r_{u_i}^l$ (or $r_{u_i}^m$).

According to Eq. (6), the similarity between two VRP solutions $\delta(y_i, y_j)$ is a value between 0 and 1. For this reason, T_S should also be chosen in the interval [0, 1]. If $T_S = 0$, the alternative solutions are forbidden to have any shared edge like in the *m*-PVRP. If $T_S = 1$, identical *k* alternative solutions may be generated. It would be also possible to define a multi-objective version of the *k*d-VRP in which the similarity threshold represents an objective to be minimized. In Section 6.3 we tested various scenarios considering different values of T_s .

The cost values used to compute $\delta(y_i, y_j)$ rely on the cost d_{ij} that is defined for each edge $(i, j) \in E$. The value of d_{ij} is commonly interpreted in the literature on VRP as either a time, a cost, or a distance. However, the measure used to calculate the objective function and that to calculate $\delta(y_i, y_j)$ does not necessarily have to be the same. In other words, an additional attribute not related to d_{ij} may be assigned to each edge in *G* and used to compute the value $\delta(y_i, y_j)$. For example, the similarity between alternative solutions could be calculated based on the distance between edges, while the objective function could use the travel costs. A different option (which is not considered in the remainder of this paper) could consider the number of shared edges between routes instead of their cost.

For the sake of simplicity we computed the similarities between alternative solutions on a graph G = (V, E). We are aware that in real-life vehicles do not travel on this graph but on a road network, where arcs between nodes do not always correspond to independent road segments. Therefore, it may happen that two different edges in graph G correspond to almost the same path on the road network, which may be problematic from a security point of view. To overcome this problem, the similarity between any couple of routes $r_{y_i}^l$ and $r_{y_j}^m$ should be computed by first analysing the corresponding physical paths on the street network and then evaluating function $c_s(r_{y_i}^l, r_{y_j}^m)$ with respect to these paths. This would correctly account for the real overlap between routes $r_{y_i}^l$ and $r_{y_j}^m$. An alternative approach consists in exploiting the freedom degree in the definition of the path corresponding to each arc of graph G so as to reduce the overlapping, though increasing the arc cost.

Another potential problem is highlighted by Martí et al. (2009), who found that the similarity measure in Eq. (3), from which Eq. (6) is derived, might generate routes that are not spatially dispersed. This may represent a problem in applications such as hazmat transportation, where the effects of a disastrous event may spread over a relatively large area. In cash-in-transit applications, however, this issue does not present a problem since an attack on a vehicle is a

localized event the consequences of which do not spread out. For applications in the chemical sector, it may be appropriate to incorporate spatial or/and temporal information into the similarity index in order to adequately evaluate the exposure to accidents of the population that lives in the areas traversed by the routes (see, e.g., Martí et al. (2009); Thyagarajan et al. (2005); Dell'Olmo et al. (2005)).

Another potential disadvantage of the similarity index in Eq. (6) is that it theoretically allows the same edge to appear in all the alternative solutions. If this is unacceptable, a different measure should be used. However in cash-in-transit applications alternative solutions presenting all a certain edge in common are considered admissible as long as: (a) the edge is traversed by a different vehicle, (b) the edge is traversed by the same vehicle in a different time frame, (c) the routes with the common edge contain different customers to be visited (or the same customers in a different order). Moreover it should be noticed that the penalty function adopted inside the solution approach, described in Section 5, discourage the selection of edges already used in other alternative solutions. In so doing the use of the same edge within all the alternative solutions is unlikely to happen.

In the remainder of the paper we use the similarity index in Eq. (6) because of its advantages such as: (a) it does not require extra parameters, and (b) it is easy to compute. The algorithm developed in the next section, however, does not depend in any way on the chosen distance measure.

5. Metaheuristic description

In this section an iterative metaheuristic to solve the kd-VRP is presented. The metaheuristic developed in this paper is similar to the iterative penalty method used in Johnson et al. (1992) and Barra et al. (1993) to find k dissimilar shortest paths. We call this method the Iterative Penalty Method for the kd-VRP (IPM_kd).

Our version of the iterative penalty method differs from the one proposed in Johnson et al. (1992); Barra et al. (1993) because at each iteration alternative solutions are selected under the conditions of a maximum degree of similarity. To this end the metaheuristic examines solution cost and solution overlaps simultaneously while searching for k alternative solutions. The IPM_kd metaheuristic is schematically shown in Algorithm 1.

The IPM_kd algorithm sequentially generates k alternative solutions, forming together a solution S for the kd-VRP. At the end of each iteration, the current solution S is compared with the best solution found so far (S^{*}). This comparison is done by evaluating the costs of solutions S and S^{*} following the objective function described in Eq. (5a) and denoted by $F = \min \max_{i \in \{1,...,k\}} f(y_i)$. At the end of the algorithm, after I iterations, S^{*} is reported.

The generation of each alternative solution is done in two consecutive steps. In the first step an iterative local search approach is used. An initial solution (y_i) is generated by the *Lin–Kernighan heuristic* followed by the *Prins splitting procedure* (see Section 5.1). This solution is then improved by using the *Variable Neighbourhood Descent (VND) heuristic* (see Section 5.3)

that is repeated *P* times starting from a new solution obtained after a diversification mechanism (see Section 5.4) to escape from local optima.

The best alternative solution (y_i^*) found during this step represents the input of the second stage of the procedure. If *S* is empty y_i^* is added directly to *S* otherwise a similarity check is performed following the similarity constraints in Eq. (5b). If y_i^* does not satisfy the similarity constraints a *penalization function* is used to increase the cost of the edges that y_i^* shares with the previously generated alternative solutions $(y_h^*, \forall h \in 1, ..., i - 1)$. Subsequently a *VND heuristic* replaces costly edges with edges that have a lower cost, guiding the algorithm towards an alternative solution that satisfies the similarity constraints. When the best alternative solution y_i^* is added to *S*, the cost matrix of the underlying VRP is updated, by penalizing all edges that have been used in y_i^* . The procedure is repeated until *k* alternative solutions are added in *S*.

Algorith	m 1: IF	PM_kd	metal	heuristic
----------	----------------	-------	-------	-----------

Read instance and initialize both kd-VRP and Heuristic parameters k, T_s , I, P, α , β and ω ; Let S^* be the best set containing k alternative solutions found so far and $F(S^*)$ its cost; $S^* \leftarrow \emptyset, F(S^*) \leftarrow \infty, l \leftarrow 0;$ while (l < I) do Let *S* be the current candidate set containing *k* alternative solutions and F(S) its cost; $S \leftarrow \emptyset, F(S) \leftarrow \infty;$ while (|S| < k) do $i \leftarrow |S|, p \leftarrow 0;$ Let y_i^* be the best *i*-th alternative solution found so far and $f(y_i^*)$ its cost; Let y_i be the current *i*-th alternative solution and $f(y_i)$ its cost; $y_i^* \leftarrow \emptyset, \ y_i \leftarrow \emptyset, \ f(y_i^*) \leftarrow \infty, \ f(y_i) \leftarrow \infty;$ while (p < P) do if (p == 0) then $y_i \leftarrow Splitting(Lin-Kernighan());$ else $y_i \leftarrow Perturbation(y_i^*, \alpha, \omega);$ $y_i \leftarrow VND(y_i);$ **if** $(f(y_i) < f(y_i^*))$ **then** $y_i^* \leftarrow y_i;$ *p* + +; if (i > 0) then while $(\delta(y_i^*, y_h^*) > T_s \ \forall h \in 1, ..., i-1)$ do PenalizationFunction (y_i^*, β) ; $y_i^* \leftarrow VND(y_i^*);$ add y_i^* to S; *PenalizationFunction*(y_i^*, β); **if** $(F(S) < F(S^*))$ **then** $S^* \leftarrow S;$ l + +;Return S^{*}

The internal parameters used by the IPM_kd algorithm (referred to as *heuristic parameters*) as well the kd-VRP key controls (referred to as *kd-VRP parameters*) are summarized in Table 1.

Four basic heuristic components, described in the following sections, are applied in the solution approach: (1) The *Lin–Kernighan heuristic* followed by the *Prins splitting procedure*; (2) a *Penalization function*; (3) a *Variable Neighbourhood Descent (VND) heuristic*; (4) a *Perturbation*.

Parameter	Description										
kd-VRP para	meters										
k	Number of alternative solutions to be generated										
T_s	Maximum similarity threshold										
Heuristic parameters											
Ι	Number of restarts of the algorithm										
Р	Number of times the Perturbation heuristic is applied										
α	Number of closer neighbour vertices to be considered in the Perturbation heuristic										
β	Penalty factor used in the Penalization function										
ω	Maximum percentage number of routes to be destroyed										

Table 1: Heuristic and kd-VRP parameters

5.1. Lin-Kernighan heuristic plus Prins splitting procedure

To find an initial alternative solution y_i the algorithm implements the *Lin-Kernighan heuristic*, followed by the *Prins splitting procedure*, both using the current cost matrix.

The Lin-Kernighan heuristic described in Lin and Kernighan (1973) is a deterministic approach generally considered to be one of the most effective methods to generate optimal or nearoptimal solutions for the symmetric travelling salesperson problem (TSP). The *Lin-Kernighan heuristic* employed in this paper uses the modified version as implemented in Helsgaun (2000).

To transform this TSP solution into a VRP solution the *Prins splitting procedure* is used. This algorithm generates a feasible VRP solution by splitting the giant TSP tour into feasible routes for which the capacity of the vehicle is not violated. For a more detailed explanation of this procedure we refer to Prins (2004).

5.2. Penalization function

The *Penalization function* is employed to update the cost matrix of the VRP, using a multiplicative penalty structure, i.e., the new cost of each edge is based on the current cost (which may have been penalized before) multiplied by $(1 + \beta)$ where β represents a penalty factor. The aim of the *Penalization function* is twofold:

- (1) Once the best alternative solution y_i^* is added to set *S*, the *Penalization function* increases the cost of all edges used in y_i^* . In doing so, the search process is forced to move to a different part of the search space. In order to speed up the search for a new feasible (with regards to the similarity constraints) alternative solution y_{i+1} (with i < k), the *Penalization function* also penalizes the edges that can be obtained by combining the vertices which are in each route of y_i^* by half of the penalty (i.e., the cost of these edges is multiplied by $(1 + \beta/2)$).
- (2) If the alternative solution y_i^{*} does not respect the similarity constraints, the *Penalization function* penalizes the use of the edges which are in common between y_i^{*} and the alternative solutions that are already contained in S (y_h^{*} ∀h ∈ 1,...,i − 1). This operation forces the

algorithm to discard the shared edges, guiding the VND heuristic (see Section 5.3) towards a feasible solution.

It should be noted that if a relatively large value of β is chosen, edges that appear in the alternative solutions already contained in *S* are discouraged more heavily, while a smaller penalty allows for more frequent recurrence of edges in the *k* alternative solutions.

5.3. Variable Neighbourhood Descent (VND) heuristic

The *VND heuristic* has a dual purpose. First, it is used every time a current alternative solution is generated in order to improve it. Second, it is applied after the *Penalization function* to guide the algorithm towards a best alternative solution that satisfies the similarity constraints. Our implementation of the *VND heuristic* uses seven different local search operators:

- Intra Route Local Search Operators which attempt to improve a single route: Internal Or-Opt and Internal Relocate, Internal 2-Opt (shown in Figure A.1 contained in Appendix A).
- *Inter Route Local Search Operators* which change more than one route simultaneously: *External Exchange, External Relocate, External 2-Opt* and *External Cross-Exchange* (shown in Figure A.2 in Appendix A).

The *VND heuristic*, shown in Algorithm 2, stops when the current solution cannot be further improved by any of the local search operators, and thus a local optimum has been reached. Each local search operator uses a first-improvement descent strategy, accepting a move that improves the current solution as soon as it is found and restarting the *VND heuristic* from the new current solution.

Finding the right order in which the local search operators are used in a deterministic *VND heuristic* may be of considerable importance for the quality of the solution. We tested different configurations and on average, the most promising order of neighbourhoods is reported in Table 2.

```
      Algorithm 2: VND heuristic

      Let y_i be the current i-th alternative solution and f(y_i) its cost;

      Let y' be a neighbouring solution of y_i and f(y') its cost;

      Let \lambda be the index used for the local search operator;

      \lambda \leftarrow 1;

      while (\lambda \leq 7) do

      y' \leftarrow N_\lambda(y_i);

      if (f(y') < f(y_i)) then

      y_i \leftarrow y';

      \lambda \leftarrow 1;

      else

      | \lambda + +;

      Return y_i
```

N_{λ}	Local Search Operator
N_1	Internal Or-Opt
N_2	Internal Relocate
N_3	Internal 2-Opt
N_4	External Exchange
N_5	External Relocate
N_6	External Cross-Exchange
N_7	External 2-Opt

Table 2: Order of the local search operators used inside the VND heuristic

5.4. Perturbation

The *perturbation heuristic* is used inside the IPM_kd algorithm as a diversification mechanism to escape from local optima, while looking for the best *i*-th alternative solution. During the perturbation heuristic a *destroy-and-repair operator*, similar to the one described in Talarico et al. (2013), is used.

First, the best alternative solution found so far (y_i^*) is partially destroyed and then it is repaired obtaining a new current solution (y_i) . The destroy-and-repair operator takes ω as a parameter, which is the number of routes to be destroyed from y_i^* , as a percentage of the total number of routes. The solution is then repaired with new routes that are generated by using a greedy randomized nearest neighbourhood heuristic. In this heuristic parameter α controls the balance between greediness and randomness. During the generation of a new route r, the next vertex to be added in r is randomly selected from a restricted candidate list. This list contains the first α closest unvisited vertices passing all a feasibility check with respect to the capacity constraint.

After the application of the destroy-and-repair operator, the newer generated solution y_i is saved as the current alternative solution and is improved by using the *VND heuristic*.

6. Computational experiments

The IPM_kd metaheuristic, described in Section 5, has been coded in Java language. In this section we report the results of the experiments performed using a machine with an Intel core i7-2760QM 2.40GHz processor with 4GB RAM. The test instances used to run the experiments are described in Section 6.1.

The results of the experiments are carried out in three different phases. In the first phase we tune the IPM_kd metaheuristic by identifying the best settings for the *heuristic parameters* (see Section 6.2). In the second phase we examine the effects of the *kd-VRP parameters* on the solutions (see Section 6.3). In the third phase (see Section 6.4) we analyse the performance of the IPM_kd metaheuristic by solving all the test instances.

6.1. Test instances

Since the *k*d-VRP has not been studied before, no test instances are available in the literature. Real-life problem instances could be generated by using the publicly accessible information (e.g., the geographical information system) to locate potential customers. However, since no benchmark solutions (or satisfactory upper/lower bounds) are known, the analysis of the IPM_kd metaheuristic, based on the quality of the obtained solutions, could not be performed. Moreover, real problems faced by cash-in-transit companies, whether available, might unlikely be made publicly available due to security and non-disclosure reasons.

For these reasons the IPM_kd metaheuristic has been extensively tested using a large set of benchmark instances taken from the VRP library. We used 51 medium and large instances from different sources (Augerat et al., 1998; Christofides et al., 1979; Fisher, 1994; Taillard, 1993; Golden et al., 1998)¹ ranging from 45 to 484 nodes. The characteristics of the instances are reported in Table 3. As mentioned, given a similarity measure between alternative solutions, an instance of the VRP can easily be transformed into an instance of the kd-VRP by adding only two parameters: the number of alternative solutions *k* and the similarity threshold T_s .

The computational experiments performed on the VRP instances are aimed at validating our proposed metaheuristic in a general setting. Further studies can be addressed at involving companies in the cash-in-transit and/or in the chemical sectors in order to incorporate additional sector specific insights into the kd-VRP and into the related solution approach.

6.2. Impact of the heuristic parameters

In the first phase of the computational experiment the *heuristic parameters* of the IPM_kd are tuned by running a full factorial experiment on a subset of the benchmark instances called *sample set*². The heuristic parameters are tuned to obtain on average, the best possible solutions (lower values of the objective functions measured by the cost of the worst solution in the set of k alternative solutions where k has been chosen equal to 3 and $T_S = 0.2$) over the sample set. A summary of the heuristic parameters, as well as the tested values, the number of tested values, and the best parameter configuration is given in Table 4.

As shown in Figure 2, if both the number of times that the IPM_kd algorithm is restarted (*I*) and the number of times that the *Perturbation heuristic* is applied (*P*) are increased, the quality of the solutions improves, but at the expense of increasing the running time. However, there is a point from which the marginal reduction of the objective function due to the increasing of these parameters diminishes while the running time increases. An analysis of convergence has been performed using the sample set showing that after a value of I = 50 and P = 10 the results are stable. Moreover the running time increases with higher values of *I*, while for values of *P* smaller than 10 the running time appears stable.

¹The instances are available at http://neo.lcc.uma.es/vrp.

²This subset contains 10% of all the test instances. In particular one instance has been randomly selected from each source. The resulting sample set is made by the following instances: *A-n53-k7*; *vrpnc1*; *F-n72-k4*; *tai100a*

Author	Name	# Nodes	Best known VRP									
			Solution									
	A-n53-k7	53	1010.0*									
	A-n65-k9	65	1177.0*									
	A-n80-k10	80	1763.0*									
	A-n69-k9	69	1159.0*									
Augerat et al	B-n56-k7	56	707.0*									
	B-n68-k9	68	1272.0*									
	B-n78-k10	78	1221.0*									
	B-n63-k10	63	1496.0*									
	vrpnc1	51	524.6									
	vrpnc2	76	835.26									
	vrpnc3	101	826.14									
Christofides et al.	vrpnc4	151	1028.42									
,	vrpnc5	200	1291.29									
	vrpnc11	121	1042.11									
	vrpnc12	101	819.56									
	F-n45-k4	45	724.0*									
Fisher	F-n72-k4	72	237.0*									
	F-n135-k7	135	1162.0*									
	tai75a	76	1618 36									
	tai75b	76	1344.62									
	tai75c	76	1291.01									
	tai75d	76	1365.42									
	tai/Ju	101	2041 34									
	tai100a	101	1040.61									
Taillard	tai100b	101	1/406.2									
Tamara	tai1000	101	1400.2									
	tail50a	101	2055.22									
	tai150a	151	2656 47									
	tai150b	151	2030.47									
	tail500	151	2541.04									
	tai1300	386	2043.35									
-	laisos	241	5497.54									
	lealler?	241	8447.02									
	keny2	321	0447.92									
	kenys	401	11030.25									
	1 11 5	461	13024.32									
	keny5	201	0400.90									
	kelly6	281	8412.88 10105.57									
	kelly/	361	10195.56									
	kelly8	441	11003.55									
	kelly9	256	583.39									
Golden et al.	Kelly10	324	/42.03									
	Kelly11	400	918.45									
	kelly12	484	1107.19									
	kelly13	253	859.11									
	kelly14	321	1081.31									
	kelly15	397	1345.23									
	kelly16	481	1622.69									
	kelly17	241	707.79									
	kelly18	301	998.73									
	kelly19	361	1366.86									
	kelly20	421	1821.15									

Table 3: VRP instances. The solutions marked with an asterisk are known to be optimal

Table 4: Heuristic parameters

Parameter	Values	#	Best setting
Ι	50	1	50
Р	10	1	10
ω	$20, 30, 40, \ldots, 90, 100\%$	9	40%
α	1, 2, 3,,9, 10	10	4
β	0.025, 0.05, 0.1, 0.2, 0.4, 0.5, 1	7	0.05

The penalty factor β presents a significant impact on both the quality of the solutions and the running times. On the one hand, smaller values of β encourage alternative solutions with

and *kelly5*.



Figure 2: Impact of *I* and *P* on the average solution quality (a) and on the running time (b)

lower costs to be generated. However, given that edges which are already used may still be selected, the similarity between the alternative solutions may be close to T_s . On the other hand, larger values of β discourage the selection of already used edges, supporting lower similarities between the alternative solutions at the expense of generating solutions with higher cost. The lower the penalty, the longer the search for feasible solutions and the better the quality of the solutions (see Figure 3 on the left). Higher penalties might allow to escape from infeasible solutions in shorter time by strongly penalizing edges already used in the current alternative solutions. Given a fixed computational time (8 seconds) the best solutions are obtained with $\beta = 0.05$. It should be noted that on average the quality of solutions is really similar for values of β between 0.025 and 0.20 as shown in the right panel of Figure 3.



(a) Average solution quality and average running time for different values of β

(b) Average solution quality in a fixed running time for different values of β

Figure 3: Impact of the heuristic parameter β , while k = 3 and $T_s = 0.2$.

6.3. Impact of the *k*d-VRP parameters

In the second step of the computational experiments, by using the best configuration of the heuristic parameters, we test the IPM_kd algorithm on the sample set (described before), using

different values of the kd-VRP parameters: (1) k (number of alternative solutions to be generated) and (2) T_s (similarity threshold). Parameters k and T_s are part of the problem input data and on a tactical level, their related values can be changed to test a different scenario by generating a different instance (less or more constrained) with a different optimal solution.

The computational time needed to generate an additional alternative solution increases with k, and depending on the similarity threshold, the cost of the k-th alternative solution increases as well. If T_S assumes higher values, the cost of the k-th alternative solution decreases, as well as the computational time (see Figure 4). In fact, the higher the similarity threshold, the less constrained the problem. If T_S is set equal to zero, the kd-VRP problem is reduced to find k disjoint alternative solutions. If T_S assumes values close to 1, the resulting k alternative solutions will share a high number of edges and the cost of the k-th alternative solution will tend to the cost of the best known VRP solution.



Figure 4: Relationship between the average solution cost (a) and the average computational time (b) for different values of T_S , while k = 3

A final remark concerns the kd-VRP problem described in Eq. (5a)–(5b) where a min-max objective function, aimed at reducing the cost of the worst alternative solution, is used. The impact of such an objective on the solutions is shown in Figure 5 where the quality of the best, the average and the worst alternative solution among the set of k alternative solutions is represented for different values of k. These results have been averaged over all the solved instances and the percentage gaps are computed considering the best, the average and the worst alternative solutions and the best known solution for the underlying VRP instance. As expected, when k increases the quality of the best alternative solution in the set of k alternative solutions remains stable, while the values of the average and the worst alternative solution in the set.

6.4. Results

In the third step of the computational experiments, using the best configuration for the heuristic parameters (see Table 4), we solve the benchmark instances described in Section 6.1. For each



Figure 5: Cost of the alternative solutions in the set of k VRP solutions by varying k

instance we execute 15 runs fixing the similarity threshold $T_S = 0.20$ (and thus the *k* alternative solutions must be different from each other by at least 80%).

The results obtained are summarized in Table 5, where for each value of k and for each instance we report: the percentage gap between the best kd-VRP solution (i.e., the worst alternative solution with the highest cost within the set of k alternative solutions) obtained after 15 runs and the best known VRP solution (column % BestGap); the percentage gap between the average cost of the kd-VRP solution obtained over 15 runs and the best known VRP solution (column % AvgGap); the average computational time in seconds (column AvgTime).

In order to evaluate the effectiveness of our solution approach we also reported the results obtained for k = 1 (see Table 5). In this case the *k*d-VRP is reduced to a standard capacitated vehicle routing problem for which the IPM_*k*d appears to be a competitive solution approach. On average the optimal gap from best known VRP solutions is only 1.79% with an average running time of 1.283 seconds. Moreover if we focus on the VRP instances for which the optimal solutions are known (instances denoted with an asterisk in Table 5), we obtained 2 optimal solutions (highlighted in bold) over 11 instances, with an average optimal gap of 0.58%.

As shown in Table 5, instances having a smaller number of vertices present higher gaps between the cost of the *k*-th alternative solution (with k > 1) and the best known VRP solution than bigger instances. In fact, when the number of vertices increases, the number of potential edges ($E = V \times V$) increase even more. Hence the possibility to select non-shared edges, with a relative low cost, is much higher for larger instances.

As expected, the average computational time grows with k (see Figure 6 on the right). The average running times needed to solve an instance in the benchmark set, when k = 5 and $T_s = 0.20$, is below 16 seconds. The relationship between the quality of solutions and k is shown in Figure 6 on the left.

Considering all the 51 benchmark instances, the average percentage gap from the best known VRP solutions is only 22.8% when 5 alternative solutions (that differ by 80% from each other) are generated. The algorithm also presents a good level of robustness since the average difference between the costs of the best and the average kd-VRP solutions remains limited to 3.8%.

Finally, given the best setting for the IPM_kd metaheuristic, the effect of the different components of the algorithm on the solutions is investigated. We observed that on average the VND



Figure 6: Best (% BestGap) and average (% AvgGap) percentage gaps (a) from the best known VRP solutions and average computing time (b) for different values of k

heuristic is able to improve the quality of the current alternative solutions by 26% while the perturbation heuristic is shown to be an effecting way to escape from local optima by improving the quality of the initial alternative solutions by on average 3%.

	AvgTime	2.038	3.066	1.788	1 543	3.099	4.525	6.62	1.592	1.863	1.719	2.201	4.656	1.746	3.632	3.729	4.784	5.779	3.093	6.833	1.06/	017 1	1./49	104.2	VL7 1	1.0/4	/07/01	720.0 13 205	6.531	91.605	7.695	21.092	18.488	27.985	5.39	14.263	53.225	18.066	11.178	18.659	26.768	32.81	20.502	31.284	49.254	78.817	11.814	21.956	30.981	//2.0/4	15.423
k = 5	% AvgGap	48.02%	43.53%	42.35% 23.16%	23 55%	33.55%	42.57%	30.29%	38.75%	46.24%	25.23%	21.31%	19.22%	8.96%	32.53%	17.04%	35.44%	13.76%	41.88%	13.84%	20.40%	40.7.45	40.04%	%C1.97	0/ 7/ 1/2%	24.10% 24 E900	%7C.0C	10.50%	12.25%	51.13%	16.31%	7.48%	15.74%	22.89%	44.97%	28.85%	18.54%	21.81%	27.93%	26.26%	23.43%	23.03%	60.06%	57.52%	55.49%	52.99%	22.58%	23.18%	21.25%	16.85%	30.19%
	% BestGap	24.91%	24.64%	%cV.cč 2977-31	15.90%	24.70%	23.88%	20.00%	29.01%	35.37%	19.39%	15.55%	14.65%	6.64%	27.01%	13.24%	27.54%	11.10%	21.48%	10.02%	21.07%	%00.10	21.41%	21.72%	14.05%	%c0.41	%6C.67	0.09% 8 51%	10.02%	20.87%	13.58%	6.39%	14.44%	20.78%	40.78%	25.87%	16.45%	21.15%	22.51%	20.39%	18.16%	16.37%	53.59%	51.10%	52.29%	43.54%	15.53%	21.00%	17.91%	15.30%	22.76%
	AvgTime	1.337	2.309	962.1 0635	1 033	1.814	1.882	1.165	1.014	1.095	1.234	1.885	3.07	1.162	2.989	0.876	1.076	1.437	1.61	1.799	1.133	5.142	1.429	C08.1	2.000	0.00	70.6	1.424 8 031	1.715	64.57	4.607	14.559	17.23	25.115	4.858	8.969	41.54	13.628	8.899	12.51	16.112	21.394	15.05	23.585	32.862	48.938	8.655	12.926	13.857	41.285	10.096
k = 4	% AvgGap	30.27%	30.61%	22.30%	16.70%	13.06%	18.40%	19.12%	31.04%	30.61%	20.41%	16.23%	15.26%	6.74%	24.94%	12.44%	30.90%	11.22%	22.33%	10.40%	20.34%	%01.C 1	21.20%	10.03%	10.5 <i>2%</i>	10.02%	0.01%	9.00%	9.55%	35.16%	13.30%	5.90%	13.55%	21.38%	39.88%	26.37%	16.60%	16.50%	14.31%	17.24%	17.35%	16.24%	46.87%	42.18%	38.60%	37.54%	14.27%	14.68%	13.18%	14.87%	20.70%
	% BestGap	18.75%	20.90%	13.65%	12.92%	10.61%	13.65%	14.23%	22.41%	21.93%	12.98%	13.13%	12.02%	5.02%	22.24%	8.78%	27.26%	8.74%	12.29%	7.30%	%8C.CI	20.00 m	0.14.40%	%UC.4 %00 8	0.79%	%7C.11	11.40%	7.96%	7.81%	18.37%	12.46%	5.06%	11.03%	19.17%	33.99%	20.99%	15.01%	16.50%	12.98%	11.91%	13.17%	12.82%	38.21%	39.28%	34.45%	32.99%	11.42%	13.50%	8.11%	13.20%	16.07%
	AvgTime	0.317	0.329	0.325	0.31	0.216	0.247	0.28	0.201	0.277	0.318	0.622	1.413	0.27	0.27	0.562	0.372	0.838	0.301	1.37	C65.0	2.191	104.0	0.441	04-1	0.022	1120	110.0	0.611	32.115	3.425	7.359	13.28	19.482	4.558	7.427	34.922	8.17	7.128	9.108	11.892	14.575	11.026	16.54	22.537	32.707	6.692	9.824	12.095	/.96.72	6.649
k = 3	% AvgGap	19.02%	20.57%	12.57%	12 90%	8.07%	11.02%	12.15%	26.60%	19.13%	14.77%	13.71%	12.86%	5.28%	16.78%	7.68%	26.21%	8.15%	10.36%	6.66%	11 200%	%6C.11	12.21%	11.37% 8 87%	0.01%	10.60%	%00.01	2070 L	7.75%	21.84%	13.11%	4.78%	11.83%	15.87%	34.56%	22.72%	12.95%	12.16%	11.13%	11.89%	12.25%	11.44%	29.16%	27.40%	23.88%	21.27%	9.91%	10.38%	10.41%	10./3%	14.13%
	% BestGap	13.84%	11.05%	0 2802	0.05%	6.76%	7.39%	9.03%	16.76%	11.20%	10.11%	8.64%	8.93%	3.49%	14.49%	5.81%	16.53%	4.44%	6.77%	4.56%	10.8/%	0.05%	%c0.6	0.81% 6.95%	%C7.0	0.1/%	%/0.0 MLC 7	0.21%	5.91%	13.60%	9.66%	2.39%	9.05%	15.55%	27.72%	18.51%	9.41%	9.49%	9.17%	9.24%	9.94%	9.75%	23.61%	21.88%	19.51%	16.74%	8.01%	8.21%	%69%/	9.38%	10.34%
	AvgTime	0.246	0.249	0.318	0.207	0.156	0.168	0.204	0.143	0.222	0.245	0.458	1.095	0.226	0.224	0.434	0.296	0.628	0.233	1.028	0.284	2.134	CI C.O	0.312	1.427	0.48/	0.110	0.410 A 21	0.501	25.807	2.912	5.387	10.556	13.73	3.134	5.346	25.135	6.385	5.2	6.583	10.265	10.843	9.229	11.549	15.727	24.942	5.545	6.649 9.94	8.84	16.658	4.992
k = 2	% AvgGap	10.08%	11.37%	8 11%	8 23%	5.45%	5.38%	8.56%	17.44%	10.47%	8.34%	8.52%	8.97%	2.67%	7.34%	5.31%	17.90%	5.58%	5.66%	3.85%	8.23%	0.94% 5.88%	0.00% 7.77.77	%C/.C	0.21% 6 019	0.91%	%CC./	0.29% 5 10%	5.18%	13.50%	10.09%	2.31%	8.88%	10.80%	25.70%	15.51%	8.99%	9.60%	8.92%	8.99%	9.47%	9.31%	12.56%	12.25%	11.86%	11.06%	7.75%	8.45%	8.56%	9.35%	8.91%
	% BestGap	3.80%	26L.L	4.66%	6 13%	3.93%	3.03%	5.56%	11.25%	5.45%	4.70%	5.02%	5.42%	1.64%	6.76%	4.34%	12.51%	3.18%	3.09%	3.11%	0.40% 5.00%	%00.C	0/0% 0/0%	5.22% A A3 07	4.43% 2 0207	3.83% 5.24%	% 77.C	4.90% 3.66 <i>%</i>	3.71%	11.02%	6.67%	0.49%	5.04%	10.22%	21.86%	12.48%	7.35%	6.98%	5.96%	6.71%	7.52%	8.00%	10.40%	9.83%	8.90%	9.25%	5.20%	6.80%	0./2% 7.00~7	7.02%	6.36%
	AvgTime	0.061	0.064	0.0/8	0.059	0.044	0.044	0.058	0.04	0.057	0.065	0.123	0.265	0.054	0.054	0.119	0.073	0.158	0.059	0.264	0.076	266.0	/0.00/	0.051	616.0	1 707	1./0/	0.102	0.127	6.452	0.734	1.325	2.554	3.779	0.855	1.348	6.337	1.545	1.458	1.863	2.524	2.666	2.215	3.208	4.409	6.184	1.434	1.797	2.306	4.2	1.283
k = 1	% AvgGap	1.35%	1.26%	1.32%	0.01%	1.73%	1.67%	1.82%	1.71%	2.43%	1.79%	3.17%	3.73%	1.63%	0.68%	0.19%	3.58%	0.52%	1.52%	0.52%	1.9/%	1.20%	4.04%	1.24%	0.19%	2.19%	0.04%	%C6.C	3.04%	4.84%	1.36%	1.41%	3.45%	4.74%	1.20%	1.28%	2.75%	4.65%	4.19%	4.79%	5.28%	5.25%	4.86%	5.56%	4.87%	4.85%	4.41%	5.00%	7.48%	5.47%	2.79%
	% BestGap	0.23%	0.40%	0.63%	0.84%	1.08%	0.67%	0.23%	0.06%	0.36%	0.79%	1.52%	3.36%	0.61%	0.17%	0.00%	2.10%	0.18%	0.00%	0.00%	0.82%	%CC.0	1.90%	0.35%	0.10%	%co.u	2.44%	0.09% 1530	1.53%	3.59%	0.00%	0.16%	1.20%	3.32%	0.09%	0.00%	1.32%	3.65%	3.07%	3.71%	4.48%	4.87%	4.29%	4.94%	4.27%	3.88%	3.52%	4.03%	4.74%	4.93%	1.79%
	id	A-n53-k7*	A-n65-k9	A-n80-k10" A-n60-l-0*	B-n56-k7*	B-n68-k9*	B-n78-k10*	B-n63-k10*	vrpnc1	vrpnc2	vrpnc3	vrpnc4	vrpnc5	vrpnc11	vrpnc12	F-n45-k4*	F-n72-k4"	F-n135-k7"	tai75a	tai/5b	Later	DC/IB1	ta1100a	tai1000	tail000	tai1000	4031504	tai150c	tai150d	tai385	kelly1	kelly2	kelly3	kelly4	kelly5	kelly6	kelly7	kelly8	kelly9	kelly10	kelly11	kelly12	kelly13	kelly14	kelly15	kelly16	kelly17	kelly18	kelly19	kelly20	Average

06 0	21.1
Ι	
F	4
0 05 and	o.vo allu
1	
<u> </u>	2
nocino	diffeo d
, mi	
Ĕ	5
Ľ	2
4	H
ć	2
ç	j
.	i
- 1	I
4	2
÷	5
، عصبيات	arter
1	>
lifforent	
د	ر ـ
foi	5
inctance	u to tanto
	Ż
remhmar	ULIVIIIII
1	<u>,</u>
Ļ	
÷	
C	2
Zeenlte	or the or
	!
L.	Ś
1	
L,	T

7. Conclusions

In this paper we have presented a new combinatorial optimization problem, including a mathematical formulation the aim of which is to generate a set of k alternative solutions of a single vehicle routing problem instance, in such a way that each alternative solution differs from all the others by at least a given threshold. A distance metric between alternative solutions has also been defined. A min-max objective function minimizes the cost of the worst solution in this set.

The kd-VRP is applicable in several practical situations. In the cash-in-transit sector, the kd-VRP can be used to define a set of alternative routes to collect or deliver cash and valuables, something which is often required by law as a preventive measure to add variability and increase unpredictability for crime organizations.

Other applications can be found in the fuel distribution or in the transportation of dangerous goods, in which all customer must be served and for specific reasons (e.g., accidents, unavailability of some route edges, security reasons), several alternative routes need to be generated.

To solve this problem, we have developed an iterative method (IPM_kd) based on a similar method for the k dissimilar shortest paths problem. The IPM_kd metaheuristic has been tested using 51 VRP benchmark instances varying the number of alternative solutions that need to be generated. The results obtained are encouraging. In a limited computational time we have generated k alternative solutions also for large and medium VRP instances, being $k \in \{1, ..., 5\}$. In particular, for k = 5 and a maximum similarity threshold $T_s = 0.20$, the VRP alternative presenting the highest cost is, on average, only 22.7% worse than the best known solutions of the original VRP problems.

In the future, we plan to investigate extensions of the *k*d-VRP, e.g., by including more sophisticated similarity indices (that include spatial and temporal information) and/or additional real-life constraints such as time windows, route length restrictions, and precedence relations between vertices. In addition, more robust metaheuristics which use complex inter-solution local search operators can be developed and compared with the iterative algorithm that is presented in this paper. The development of exact methods based e.g., on a column generation approach and the definition of optimal solutions and bounds for the *k*d-VRP problem, both present promising research avenues.

A. Local search operators used inside the VND heuristic

Here we report some illustrative examples of the intra route (Figure A.1) and inter route (Figure A.2) local search operators that are used inside the VND heuristic. Each figure shows a VRP solution before (left side) and after (right side) the application of the corresponding local search operator. The depot is represented by a black square while black circles denote the customers. The length of the edges is proportional to their cost.



Figure A.1: Intra route local search operators



Figure A.2: Inter route local search operators

References

- V. Akgün, E. Erkut, and R. Batta. On finding dissimilar paths. European Journal of Operational Research, 121(2):232–246, 2000.
- P. Augerat, J.M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical report, Research Report 949-M, Universite Joseph Fourier, Grenoble, France, 1998.
- T. Barra, B. Perez, and J. Anez. Multidimensional path search and assignment. In *21st PTRC Summmer Annual Conference*, 1993.
- N. Christofides, A. Mingozzi, P. Toth, and C. Sandi. The vehicle routing problem. In *Combinatorial Optimization*, pages 315–338. John Wiley and Sons, Chichester, 1979.
- P. Dell'Olmo, M. Gentili, and A. Scozzari. On finding dissimilar Pareto-optimal paths. European Journal of Operational Research, 162(1):70–82, 2005.
- L. Di Puglia Pugliese and F. Guerriero. Dynamic programming approaches to solve the shortest path problem with forbidden paths. *Optimization Methods and Software*, 28(2):221–255, 2013.
- M.L. Fisher. Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research*, 42(4):626–642, 1994.
- B.L. Golden, E.A. Wasil, J.P. Kelly, and I.M. Chao. *Metaheuristics in vehicle routing*, volume Fleet Management and Logistics. Kluwer, Boston, Crainic, T.G., and Laporte, G. (eds) edition, 1998.
- R. Gopalan, S.K. Kolluri, R. Batta, and M.K. Karwan. Modeling equity of risk in the transportation of hazardous materials. *Operations Research*, 38(6):961–973, 1990.
- Z. Gotthilf and M. Lewenstein. Improved algorithms for the k simple shortest paths and the replacement paths problems. *Information Processing Letters*, 109(7):352–355, 2009.
- K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.
- J. Hershberger, M. Maxel, and S. Suri. Finding the k shortest simple paths: A new algorithm and its implementation. ACM Transactions on Algorithms (TALG), 3(4):45-56, 2007.
- P.E. Johnson, D.S. Joy, and D.B. Clarke. Highway 3.01, an enhancement routing model: program, description, methodology and revised user's manual. Technical report, Oak Ridge National Laboratories, 1992.
- M. Kuby, X. Zhongyi, and X. Xiaondon. A minimax method for finding the k best 'differentiated' paths. *Geoghaphical Analysis*, 29(4):298–313, 1997.
- V.I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Dokladyy*, 10:707–710, 1966.
- S. Lin and B.W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):498–516, 1973.
- A. Løkketangen, J. Oppen, J. Oyola, and D.L. Woodruff. An attribute based similarity function for VRP decision support. *Decision Making in Manufacturing and Services*, 6(2):65–83, 2012.
- K. Lombard and R.L. Church. The gateway shortest path problem: Generating alternative routes for a corridor location problem. *Geographical Systems*, 1(1):25–45, 1993.

- R. Martí, J.L. González-Velarde, and A. Duarte. Heuristics for the bi-objective path dissimilarity problem. *Computers & Operations Research*, 36(11):2905–2912, 2009.
- J. Michallet, C. Prins, L. Amodeo, F. Yalaoui, and G. Vitry. Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services. *Computers & Operations Research*, 41(0):196–207, 2014.
- S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. A hybrid tabu search for the m-peripatetic vehicle routing problem. *Matheuristics*, 10:253–266, 2010a.
- S.U. Ngueveu, C. Prins, and R. Wolfler Calvo. Lower and upper bounds for the m-peripatetic vehicle routing problem. 4OR, 8(4):387–406, 2010b.
- D. Park, S.L. Sharma, L.R. Rilett, and M. Chang. Identifying multiple reasonable alternative routes: Efficient vector labeling approach. *Transportation Research Record: Journal of the Transportation Research Board*, 1783(1):111–118, 2002.
- C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985 –2002, 2004.
- W. Sattayaprasert, P. Taneerananon, S. Hanaoka, and R. Pradhananga. Creating a risk-based network for hazmat logistics by route prioritization with AHP: case study: Gasoline logistics in Rayong Thailand. *International Association of Traffic and Safety Sciences*, 32(1):74–87, 2008.
- SPFI Service Public Federal Interieur. Arrêté royal réglant certaines méthodes de surveillance et de protection du transport de valeurs et relatif aux spécificités techniques des véhicules de transport de valeurs. Moniteur Belge, 7 April 2003.
- K. Sörensen. Route stability in vehicle routing decisions: a bi-objective approach using metaheuristics. *Central European Journal of Operations Research*, 14(2):193–207, 2006.
- K. Sörensen. Distance measures based on the edit distance for permutation-type representations. *Journal* of *Heuristics*, 13(1):35–47, 2007.
- J.W. Suurballe. Disjoint paths in a network. Networks, 4(2):125-145, 1974.
- E.D. Taillard. Parallel iterative search methods for vehicle routing problems. Networks, 23(8):661–676, 1993.
- L. Talarico, K. Sörensen, and J. Springael. The risk-constrained cash-in-transit vehicle routing problem with time window constraints. In A. Fink and M.J. Geiger, editors, *14th Workshop of the EURO Working Group "EU/ME: the metaheuristics community*", pages 104–109, 2013.
- K. Thyagarajan, R. Batta, M.H. Karwan, and R.J. Szczerba. Planning dissimilar paths for military units. *Military Operations Research*, 10(1):25–42, 2005.
- P.H. Van Leeuwen and A. Volgenant. Solving symmetric vehicle routing problems asymmetrically. *European Journal of Operational Research*, 12(4):388–393, 1983.
- S. Vanhove. *Alternative Routing Algorithms for Road Networks*. phd thesis, Ghent University, Department of Applied Mathematics and Informatics, 2012.
- R. Wolfler Calvo and R. Cordone. A heuristic approach to the overnight security service problem. Computers & Operations Research, 30(9):1269–1287, 2003.
- S. Yan, S.S. Wang, and M.W. Wu. A model with a solution algorithm for the cash transportation vehicle routing and scheduling problem. *Computers & Industrial Engineering*, 63(2):464 473, 2012.

J.L. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.