



An exponential (matching based) neighborhood for the vehicle routing problem

Eric Angel, Evripidis Bampis, Fanny Pascual

► To cite this version:

Eric Angel, Evripidis Bampis, Fanny Pascual. An exponential (matching based) neighborhood for the vehicle routing problem. *Journal of Combinatorial Optimization*, 2008, 15 (2), pp.179–190. 10.1007/s10878-007-9075-3 . hal-00341352

HAL Id: hal-00341352

<https://hal.science/hal-00341352>

Submitted on 27 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An exponential (matching based) neighborhood for the Vehicle Routing Problem

Eric Angel¹, Evripidis Bampis¹, and Fanny Pascual¹

Abstract

We introduce an *exponential neighborhood* for the Vehicle Routing Problem (VRP) with *unit* customers' demands, and we show that it can be explored efficiently in polynomial time by reducing its exploration to a particular case of the Restricted Complete Matching (RCM) problem that we prove to be polynomial time solvable using flow techniques. Furthermore, we show that in the general case with non-unit customers' demands the exploration of the neighborhood becomes an \mathcal{NP} -hard problem.

Keywords: local search, exponential neighborhood, vehicle routing problem, matching

1 Introduction

The intractability of many combinatorial optimization problems motivated the use of heuristic algorithms that aim to find a nearly optimal solution in a reasonable amount of computation time. An important class of heuristics is the class of *local search algorithms* [AL97]. A local search algorithm starts with a feasible solution and iteratively tries to improve it by searching at each iteration the “neighborhood” of the current solution, i.e. a set of feasible solutions that are close to the current solution. The quality of the returned solution (*local optimum*) as well as the computation time of such an algorithm are very closely related to the choice of the neighborhood structure. An important parameter of the structure of a neighborhood is its size: intuitively, the larger the neighborhood is, the better the quality of the returned solution is, but at the same time the longer it takes to search the neighborhood at each iteration. Thus, a larger neighborhood does not necessarily implies a more efficient heuristic algorithm. This is the reason why many recent works are focused on exponential neighborhoods and try to identify the borderline between neighborhoods that can / that cannot be explored efficiently, i.e., in polynomial time with respect to the size of the input [AEOP02]. Studies in this vein have been conducted for different combinatorial optimization problems, including scheduling problems [CPvdV02, Hur99], the traveling salesman problem [Gut99, GYZ02] and generalizations such as the quadratic assignment problem [DW00]. For some of these problems there exists exponential neighborhoods that can be searched in polynomial time (TSP, scheduling) whereas for others, like the quadratic assignment problem, the situation is more difficult since the search of various types of exponential neighborhoods leads to \mathcal{NP} -hard optimization problems.

In this paper, we focus on the *Vehicle Routing Problem* (VRP) [TV01]. The design of exponential neighborhoods for this problem has been previously addressed. Ergun et al. [EOAF02] have proposed several exponential neighborhoods, and they showed how to find a good neighboring solution by proposing a heuristic for searching a constrained shortest path on an auxiliary graph (a similar approach was used in [TP93]). Xu and Kelly [XK96] proposed a tabu search in

¹IBISC – Université d'Évry Val d'Essonne, CNRS FRE 2873 –, 523 Place des Terrasses, 91000 Évry, France. Tel: 33 1 60 87 39 06; Fax: 33 1 60 87 37 89; E-mail: {e.angel, bampis, f.pascual}@ibisc.univ-evry.fr

which a large neighborhood is searched using a minimum cost flow problem. However, in case several deletions/insertions of clients occur in a single route simultaneously, the cost of the flow may only approximate the actual move cost.

In this paper, we consider a new neighborhood of exponential size defined by *matchings* that adapts the exponential neighborhood introduced by Gutin for the traveling salesman problem [Gut99] to the VRP problem. Gutin proved that, in the case of the traveling salesman problem, the neighborhood is polynomial time searchable using a reduction to the minimum weighted perfect matching problem in bipartite graphs. We prove in what follows that the exponential neighborhood that we introduce here for the VRP can also be (fully) explored in polynomial time in the case where each customer asks for exactly one unit of goods. To do so, we give a reduction to a particular case of the *Restricted Complete Matching* (RCM) problem [IRT78], the MIN WPRCM problem defined below, that we solve by using flow techniques. Furthermore, we show that the problem becomes \mathcal{NP} -hard for non-unit demands.

Overview. In the remaining part of this section, we introduce formally the variants of the VRP that we consider throughout this paper, the MIN WPRCM problem and some notations that we use in the sequel. In Section 2, we define the exponential neighborhood for the VRP. In Section 3, we give a reduction of the problem of exploring the proposed neighborhood for *unit* demands to the MIN WPRCM problem, and in Section 4, we solve this problem using flow techniques. In Section 5 we show that the considered neighborhood cannot be explored in polynomial time in the general case of the VRP where the customers are allowed to ask for any arbitrary quantity of goods. Section 6 concludes the paper.

Notation and definitions. The *Vehicle Routing Problem* (VRP) is defined as follows: n customers must be served from a single depot. Each customer asks for some amount of goods and a vehicle of capacity C is available to deliver goods. Since the vehicle capacity is limited, the vehicle has to periodically return to the depot for reloading. It is not possible to split customer delivery. Therefore, a VRP solution is a collection of tours where each customer is visited only once, and the quantity of goods delivered along a tour does not exceed the vehicle capacity C .

In the first part of the paper, we will consider a restricted version of the VRP in which the demand of each customer is exactly one, and thus the number of customers served per tour is at most C . We call this variant the VRP *with unit demands*. In the last part we consider the general VRP problem as defined above.

From a graph theory point of view the VRP *with unit demands* may be stated as follows: Let $G(V, E)$ be a complete graph with node set $V = \{0, 1, 2, \dots, n\}$ and arc set E . In this graph model, 0 is the depot and the other nodes are the customers to be served. Each arc (i, j) is associated with value $d_{i,j}$ representing the distance (or travel time) between i and j . The goal is to find a set of tours of minimum total distance (travel time). Each tour starts from and terminates at the depot 0, each node must be visited exactly once and the length of each tour is at most C .

Example 1. Figure 1 shows an instance of the VRP with unit demands, where there are 7 customers. The capacity of the vehicle is 4 and the distance matrix is shown on Figure 1 *Left*. A solution of cost 16 is shown on Figure 1 *Right* (the cost is the total length of the tours, i.e.

	0	1	2	3	4	5	6	7
0	0	2	1	4	4	1	4	2
1	4	0	3	4	1	4	4	4
2	4	4	0	2	4	4	1	4
3	1	4	4	0	1	4	4	4
4	1	4	4	2	0	4	4	4
5	4	1	4	4	4	0	3	4
6	4	4	4	4	4	4	0	1
7	2	4	4	4	4	4	4	0

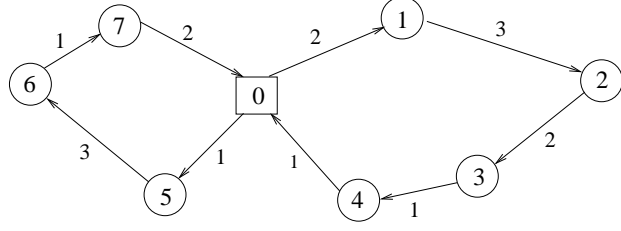


Figure 1: *Left*: Distance matrix. *Right*: A solution of the VRP with unit demands.

the sum of the lengths of the arcs of the tours). There are 2 tours: in the first one the vehicle visits the customers 1, 2, 3, 4 and go back to the depot, and in the second tour it visits the clients 5, 6, 7 and go back to the depot.

The *Restricted Complete Matching* (RCM) problem can be defined as follows: in a *bipartite* graph $G(V, E)$, i.e. in a graph where V can be partitioned into two disjoint sets, V_1 and V_2 with all edges having one endpoint in V_1 and the other endpoint in V_2 , a set $M \subseteq E$ is a *matching* if no vertex of V is incident with more than one edge of M . The size of a matching M is the number of its edges. If $|V_1| \leq |V_2|$ and every vertex $x \in V_1$ is incident with an edge of M , then the matching is said to be *complete*. Let E_1, E_2, \dots, E_k be subsets of E , and r_1, r_2, \dots, r_k positive integers. The *restricted complete matching* problem (RCM) consists in determining whether there exists a complete matching M for G which also satisfies the restrictions:

$$|M \cap E_j| \leq r_j \quad \forall j \in \{1, \dots, k\}.$$

It has been shown in [IRT78] that this problem is \mathcal{NP} -complete (reduction from the satisfiability problem of Boolean expressions). For the case of a single restriction, the authors presented a polynomial time algorithm.

We are interested in the following subproblem of RCM: The set of subsets E_1, E_2, \dots, E_k is not arbitrary but it corresponds to a partition of V_2 : T_1, T_2, \dots, T_k . For every $j \in \{1, \dots, k\}$, let E_j be the set of all the edges of E which have an endpoint in T_j . The *Particular Restricted Complete Matching* problem (PRCM) is to determine whether there exists a complete matching M for G which also satisfies the restrictions:

$$|M \cap E_j| \leq r_j \quad \forall j \in \{1, \dots, k\}.$$

In the weighted version of PRCM, every edge $[v, w]$ has a weight c_{vw} . Our aim is to find a minimum-weight particular restricted complete matching. We will call this version MIN WPRCM in the sequel. As usually, m will denote the number of edges of G .

Example 2. Figure 2 shows an example of an instance of MIN WPRCM: we have a (complete) bipartite graph with $V_1 = \{2, 4, 5, 7\}$, $V_2 = T_1 \cup T_2$, $T_1 = \{0, 6\}$ and $T_2 = \{0', 1, 3\}$. E_1 is the set of all the edges which have 0 or 6 as endpoint: $E_1 = \{[2, 0], [2, 6], [4, 0], [4, 6], [5, 0], [5, 6], [7, 0], [7, 6]\}$,

and E_2 is the set of edges which have $0'$, 1 , or 6 as endpoint, that is the remaining edges. Figure 2 *Right* represents the distance matrix (or the cost matrix), i.e. it indicates the costs between vertices of V_1 and vertices of V_2 . We fix $r_1 = r_2 = 2$. Figure 2 *Left* represents a solution of the MIN WPRCM problem: edges $\{[2, 0], [4, 1], [5, 0'], [7, 6]\}$ form a minimum-weight particular restricted complete matching of cost -4 .

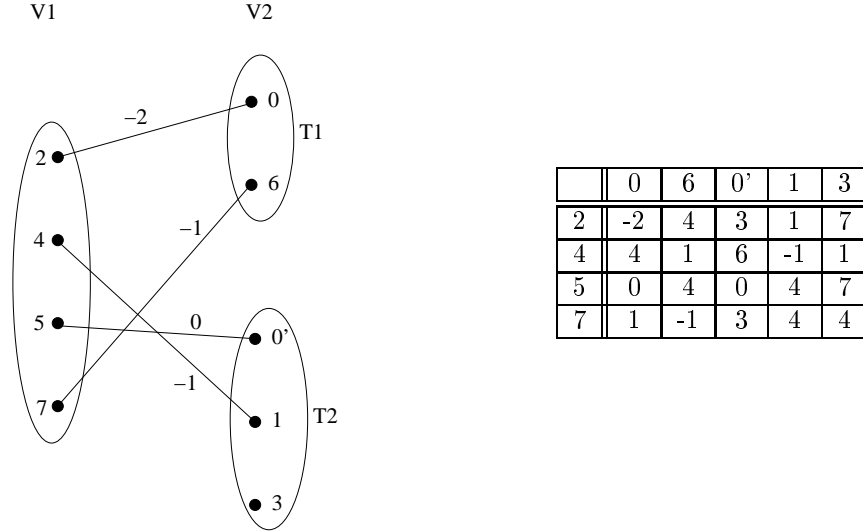


Figure 2: An instance of the MIN WPRCM problem, with $V_1 = \{2, 4, 5, 7\}$, $V_2 = \{0, 0', 1, 3, 6\}$, $T_1 = \{0, 6\}$, $T_2 = \{0', 1, 3\}$ and $r_1 = r_2 = 2$. *Left*: An optimal solution with cost -4 . *Right*: Distance matrix.

2 An exponential neighborhood for the VRP

Local search remains the main practical tool for finding near optimal solutions for large instances of the VRP. Let us consider the following local search algorithm:

We start with an arbitrary feasible solution S of the VRP. Then, in this solution, we choose a subset of customers, which will be considered as *mobile*. The other customers and the depot will be *fixed*. Let us call vertices the set of clients and the depot. A *neighbor solution* of S is a solution S' in which each mobile vertex has been inserted between two fixed vertices of the initial solution (these two vertices may possibly be both the depot in the case where the only fixed vertex of a tour is the depot). Note that we can insert at most one vertex between two fixed vertices and so the number of mobile vertices must be smaller than or equal to the number of arcs in the graph of fixed vertices, which is smaller than or equal to twice the number of fixed vertices. Note also that the number of tours cannot increase: it is constant, or decreases in the case where the only fixed vertex of a tour is the depot and no mobile vertex is inserted in this tour.

We do not focus in this paper on the best way to determine the initial solution and the set of mobile vertices. The set of mobile vertices may be for example randomly chosen among the

set of all vertices. It would certainly be preferable not to let all the vertices fixed in a tour if the quantity of delivery goods in this tour (number of vertices in the case of unit demands) is equal to the capacity of the vehicle. Indeed no mobile vertex could be inserted in this tour. In such a tour, moving at least a few vertices from the set of fixed vertices to the set of mobile vertices could only improve the quality of the best neighbor solution, since it would just add some new possible solutions in the neighborhood. It is easy to check whenever a couple (initial solution, set of mobile vertices) has a non-empty neighborhood. This is the case if and only if the two following conditions are satisfied. First, it must be possible to insert all the mobile vertices between two fixed vertices, i.e. the number of arcs in the graph where the mobile vertices have been removed has to be larger than or equal to the number of mobile vertices. Moreover it must be possible to obtain a neighbor solution where every tour has a quantity of delivered goods (length, in the case of unit demands) smaller than or equal to the vehicle capacity C . In the case of the VRP with unit demands, if, step by step, we add a mobile vertex to the tour which has the smallest length, then the length of the longest tour must be smaller than or equal to C , when all the mobile vertices have been inserted.

Example 3. Figure 3 shows an initial solution of cost 16 (*Left*). The fixed vertices form a new graph (*Right*), in which we can at most insert one mobile vertex between two (fixed) vertices. Figure 4 shows a neighbor solution of Figure 3 *Left*. The cost of this new solution is 11 (the distance matrix is the one given in Figure 1). The orientation of each tour is preserved during the construction of the neighbor solution.

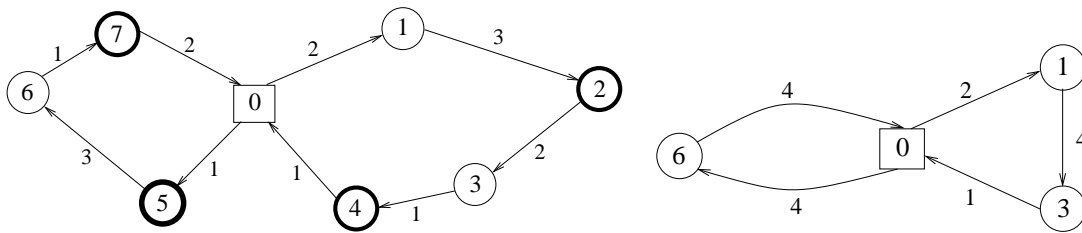


Figure 3: *Left*: a solution of the VRP. Vertices circled with thick lines are mobile vertices. *Right*: the corresponding graph where mobile vertices have been removed.

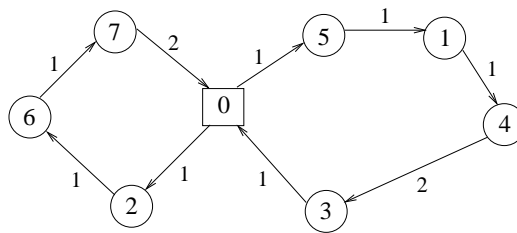


Figure 4: A solution of cost 11 for the VRP.

Let us denote the above defined neighborhood the *multi-insertion neighborhood*. We consider

a deepest local search procedure, and therefore at each step we look for the best neighbor solution of the current solution. This solution (ties are broken arbitrarily) is the current solution of the next step. To obtain an efficient algorithm, the time needed to find a best neighbor solution at each step must be polynomial.

Theorem 2.1 *In the multi-insertion neighborhood, the number of neighbor solutions of a solution may be exponential, even for the VRP with unit demands.*

Proof: Let us consider the following Vehicle Routing Problem: we have n clients to supply (where n is an even number) and the capacity of the vehicle is at least n . Let us suppose that the initial solution is a solution in which there is only one tour and that we have $\frac{n}{2}$ mobile vertices. The number of neighbor solutions is then $(\frac{n}{2} + 1)!$ \square

Since the number of neighbor solutions is possibly exponential, we cannot enumerate all the solutions and look for the best one, but we have to find a method to get the best neighbor solution in a polynomial time. In order to do that, we will now show that it is possible to reduce the search of the best solution of the multi-insertion neighborhood into a particular restricted weighted matching problem.

3 Reduction of the neighborhood exploration to the MIN WPRCM in the case of unit demands

In this section, we will say that, given a solution S and a set of mobile vertices, S' is a neighbor of S if and only if S' can be obtained by a step of the following local search algorithm: S' is obtained from S by removing the mobile vertices and by inserting each of them between two fixed vertices. S' is a best neighbor of S if and only if S' is a neighbor of S and there is no other neighbor of S which has a cost (a total distance of the tours) smaller than the cost of S' .

Let us suppose that we have an instance of the VRP with unit demands: a distance matrix D (where $d_{i,j}$ is the distance from i to j), a vehicle capacity C , an initial feasible solution S (represented as a graph $G(V, E)$) and a set of mobile customers. Let k denotes the number of tours in S . In S , each fixed vertex u has a successor $s(u)$ which is the fixed vertex which follows it in the tour (we have $s(u) = u$ if u is the depot and it is the only fixed vertex of a tour). For example, if S is the solution represented in Figure 3 *Left*, the successor $s(1)$ of vertex 1, is 3. It is the direct successor of 1 once we removed the mobile edges (see Figure 3 *Right*).

Let us model the search of the best solution in the neighborhood of S by a matching problem. Let $G'(V', E')$ be a complete bipartite graph such that $V' = V_1 \cup V_2$, where the vertices of V_1 are the mobile customers and the vertices of V_2 are the fixed customers plus k copies of the depot. We have then $E' = \{[u, v] \mid u \in V_1, v \in V_2\}$. The weights of the edges are defined as follows: let $u \in V_1$ and $v \in V_2$, the weight of the edge $[u, v]$ is equal to $d_{v,u} + d_{u,s(v)} - d_{v,s(v)}$. If an edge $[u, v]$ belongs to the matching we obtained, it means that the vertex u is inserted between the vertices v and $s(v)$. Let $\{T_1, T_2, \dots, T_k\}$ be a partition of V_2 such that for each $i \in \{1, \dots, k\}$, the vertices of T_i are the fixed customers of the i^{th} tour of S and one copy of the depot. Let $r_i = C - |T_i| + 1$. Our aim is to find in G' a minimum-weight complete matching

M such that the number of edges in $M \cap E_i$ is smaller than or equal to r_i , for each $i \in \{1, \dots, k\}$.

Example 4. Figure 2 shows the matching instance corresponding to the VRP instance whose initial solution is showed in Figure 3. The distance matrix (of G') is shown in Figure 2 *Right*. In Figure 2 *Left*, we did not draw all the edges but only the four edges which belong to the minimum-weight particular restricted complete matching (of cost -4). This optimal solution corresponds to the solution of Figure 4 for the VRP (the best neighbor solution of the instance of Figure 3): we start with the initial solution of Figure 3 and insert vertex 2 between 0 and 6, and 7 between 6 and 0, in the first tour (the tour whose fixed vertices are those of T_1) and we insert 4 between 1 and 3, and 5 between the depot (0' is a copy of the depot) and 1 in the second tour.

Theorem 3.1 *Let S be a feasible solution of the VRP with unit demands and let ξ be a subset of the vertices of S defined as mobile. Let G' be the bipartite graph which corresponds to the multi-insertion neighborhood of S (G' is constructed as stated above). The solution M of the MIN WPRCM problem on G' gives us a best neighboring solution S' of S , and S' is the solution which corresponds to S in which we have removed the mobile vertices and then inserted each mobile edge as follows: if $[u, v] \in M$, vertex u is inserted between the vertices v and $s(v)$.*

Proof: If we choose $[u, v]$ in M , it means that we insert the mobile vertex u between the fixed vertices v and $s(v)$: the cost of $[u, v]$ is equal to $d_{v,u} + d_{u,s(v)} - d_{v,s(v)}$, which is the cost to go from v to u , plus the cost to go from u to $s(v)$ ($s(v)$ is the fixed successor of v), minus the cost to go from v to $s(v)$. Thus, the sum of the weights of the edges in M is the sum that we have to add to the total distance of the tours in the graph of the fixed vertices of G to obtain the total distance of the tours of S' . Therefore, the smaller this sum is, the smaller the total distance of S' will be. Let us now show that all the neighbors of S can be represented by a particular restricted complete matching in G' and that only the neighbors of S are represented by such a matching.

We know that the number of mobile vertices is smaller than or equal to the number of arcs in the graph of fixed vertices (otherwise we could not insert each mobile vertex between two fixed vertices and S would not have any neighbor). This number is smaller than or equal to the number of fixed vertices (not taking into account the depot), plus the number of tours in the graph. Indeed, for each fixed vertex except the depot, there is one and only one outgoing arc, and the number of outgoing arcs of the depot is equal to the number of tours, k . Thus, the number of arcs in the graph of fixed vertices is equal to $|V_2|$, and we deduce from it that $|V_1| \leq |V_2|$. Therefore, since M is a complete matching of G' , all the vertices of V_1 belong to M . This means that all the mobile vertices will be inserted in S' . Moreover, since M is a matching, each vertex is inserted at most once. So each mobile vertex u is inserted once and only once in S' and it is inserted between two fixed vertices (v and $s(v)$, which are both fixed vertices, by construction).

Let us now show that the solution obtained is feasible: each tour must contain the depot and there must be at most $C + 1$ vertices in a tour (C clients and the depot). We know that, for each $i \in \{1, \dots, k\}$, the number of edges in $M \cap E_i$ is smaller than or equal to $r_i = C - |T_i| + 1$. $|T_i|$ is the number of fixed vertices of the i^{th} tour and it contains the depot. The number of

edges in $M \cap E_i$ is the number of mobile vertices added to this tour. The number of vertices in the i^{th} tour of S' is then $|T_i| + |M \cap E_i| \leq C + 1$ and the i^{th} tour contains the depot.

Thus, each solution of the particular restricted complete matching problem corresponds to a neighbor of S and each neighbor of S corresponds to a solution of the particular restricted complete matching problem. Since S' is the solution of minimum cost, S' is the best neighbor of S . \square

4 Resolution of the MIN WPRCM problem

The MIN WPRCM problem can be expressed by an integer linear program as follows: We have a bipartite graph $G = (V, E)$ with vertices $V = (V_1, V_2)$, and a partition $\{T_1, T_2, \dots, T_k\}$ of vertices of V_2 . Let $\{E_1, E_2, \dots, E_k\}$ be a partition of the edges E such that $[u, v] \in E_i$ if and only if $v \in T_i$.

$$\text{Minimize } \sum_{[u,v] \in E} c_{uv} x_{uv} \quad (1)$$

subject to:

$$\sum_{v \in V_2, [u,v] \in E} x_{uv} = 1 \quad \forall u \in V_1 \quad (2)$$

$$\sum_{u \in V_1, [u,v] \in E} x_{uv} \leq 1 \quad \forall v \in V_2 \quad (3)$$

$$\sum_{[u,v] \in E_i} x_{uv} \leq r_i \quad \forall i \in \{1, \dots, k\} \quad (4)$$

$$x_{uv} \in \{0, 1\} \quad \forall [u, v] \in E. \quad (5)$$

We have $x_{vw} = 1$ if the edge $[v, w]$ belongs to the matching we look for, and $x_{vw} = 0$ otherwise. Equation (1) means that our goal is to minimize the weight of the chosen edges (this means for the VRP that the distance to connect the mobile vertices to tours of the fixed vertices has to be as small as possible). Since $|V_1| \leq |V_2|$, Equations (2) et (3) mean that we are looking for a matching (for the VRP, Equation (2) means that each mobile vertex must be inserted once and only once, and Equation (3) means that at most one mobile vertex is inserted between two fixed vertices). Equation (4) represents the constraint capacities (for the VRP, the number of mobile vertices inserted in a same tour has to be smaller than or equal to the remaining capacity of the vehicle).

Since the constraint matrix of the linear program is totally unimodular (see [ABPW06]), we can polynomially solve it by using linear programming techniques. However it is more convenient to use flow techniques to solve this problem.

We can express the MIN WPRCM problem by the following flow problem: we have an oriented graph $G' = (V', E')$ where each arc (u, v) has a cost $w_{u,v}$ and a capacity $Cap_{u,v}$, we have a source and a sink and we are looking for a maximum flow of minimum cost. The vertices V' are

the source, vertices V_1 and V_2 (also belonging to G), k vertices $\{a_1, a_2, \dots, a_k\}$, and the sink. The source has an outgoing arc towards each vertex of V_1 . Each vertex $u \in V_1$ has an outgoing arc towards vertex $v \in V_2$ if and only if there is an edge $[u, v] \in E$. Each vertex $u \in T_i$ has an outgoing arc towards vertex a_i , and vertices $\{a_1, a_2, \dots, a_k\}$ have an outgoing arc towards the sink. The cost of all the arcs are 0, except for the arcs between the vertices of V_1 and the vertices of V_2 : the cost of arc (u, v) is equal to the cost of this edge in the corresponding MIN WPRCM problem: $w_{u,v} = c_{u,v}$. The capacity of all the arcs is 1, except for the arcs between vertices $\{a_1, a_2, \dots, a_k\}$ and the sink: the capacity of the arc between a_i and the sink is equal to r_i . $|V_1|$ units of flow are released from the source. For example, Figure 5 shows the flow instance corresponding to the MIN WPRCM problem instance shown on Figure 2.

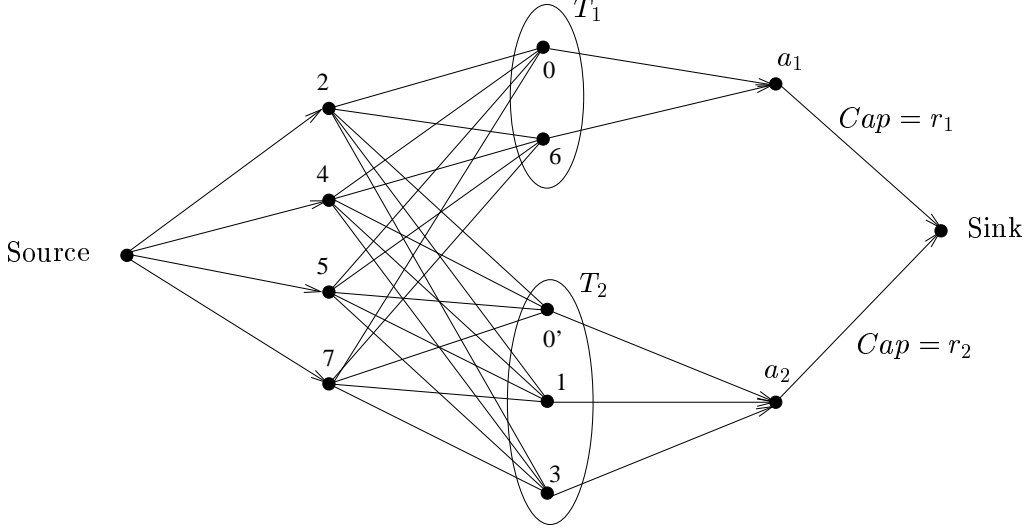


Figure 5: Flow instance corresponding to the matching instance of Figure 2. Arcs are oriented left to right. The capacity of each arc is equal to Cap , or 1 when Cap is not written. The cost of each arc is equal to 0 except for the arcs between the vertices of V_1 and the vertices of V_2 .

We can polynomially find the maximum flow of minimum cost using a flow algorithm [AMO93]. Knowing this solution, we can deduce a solution to the corresponding MIN WPRCM problem: an edge between vertices of V_1 and V_2 belongs to the minimum weight matching if and only if there is one unit of flow which goes through this edge in G' . Indeed, each vertex of V_1 releases one unit of flow, there is at most one unit of flow which goes through each vertex of V_2 , and no more than r_i flows units come from vertices of T_i .

Given the reduction presented in the previous section, we obtain the following result.

Corollary 4.1 *The problem of finding the best solution in the exponential multi-insertion neighborhood, for the VRP problem with unit demands, can be solved in polynomial time.*

One can notice that to use the multi-insertion neighborhood within a tabu search framework, one needs to identify a move with minimum positive cost. This can be easily done, by solving a set of instances of the maximum flow of minimum cost. In each instance the cost of an edge is fixed to a large value, in order to prevent a mobile vertex to be inserted in the same position than the initial solution.

5 Case where the demands of the customers are arbitrary

We have considered so far a restricted version of the VRP in which the demand of each customer is exactly one. We may ask whether the exponential neighborhood that we have introduced can also be explored in polynomial time if the demands of the customers are arbitrary, i.e. if each customer asks for a number (integer) of units of goods, which is not necessarily one. In this case, the multi-insertion neighborhood is the one explained in Section 2, and in which a solution is feasible if and only if the sum of the demands of the customers in the same tour is smaller than or equal to the capacity of the vehicle.

Theorem 5.1 *It is \mathcal{NP} -hard to find the best solution in the multi-insertion neighborhood if the demands of the customers are arbitrary, and this even if the distances (in the matrix of the distances) are all equal to 1.*

Proof: We give a reduction from the PARTITION problem [GJ79]: Given $n \geq 3$ integer numbers x_1, \dots, x_n with $\sum_{i=1}^n x_i = 2L$ for some integer L , the task is to decide whether there is a subset $B \subseteq \{1, \dots, n\}$ such that $\sum_{i \in B} x_i = L$. Given an instance of PARTITION, we construct the following instance of our problem:

We have a depot and $3n - 4$ customers. In the initial solution there are $n + 2$ tours: n tours of length 2, in which there is only one customer to be served, and 2 tours in which there are $n - 2$ customers per tour. The mobile vertices are the customers of the n first tours (the tours of length 2); the other customers and the depot are fixed vertices. Figure 6 shows such an instance when $n = 5$. Each fixed customer has a demand of 1, each mobile customer i has a demand equal to x_i , and the vehicle capacity is $n - 2 + \frac{\sum_{i=1}^n x_i}{2}$. The distance between every couple of vertices is 1.

We now have to insert the mobile vertices between the fixed ones, and we wish to minimize the total distance of the tours. The best solution is obtained when the cost of adding a mobile vertex to the tours of fixed vertices is 1: the total distance is then equal to $3n - 2$. This is possible if and only if the mobile vertices are not inserted in a tour where the only fixed vertex is the depot, that is if they are inserted in the large tours. Since the vehicle capacity is $n - 2 + \frac{\sum_{i=1}^n x_i}{2}$, and since each fixed customer has a demand of 1, the sum of the demands of the mobile customers inserted in a tour has to be smaller than or equal to $\frac{\sum_{i=1}^n x_i}{2}$. The total sum of the demands of the mobile vertices is equal to $\sum_{i=1}^n x_i$, so this is possible if and only if there exists a partition (M_1, M_2) of the mobile customers such that the sum of the demands of customers of M_1 is equal to the sum of the demands of customers of M_2 , that is if there is a solution of the PARTITION problem. \square

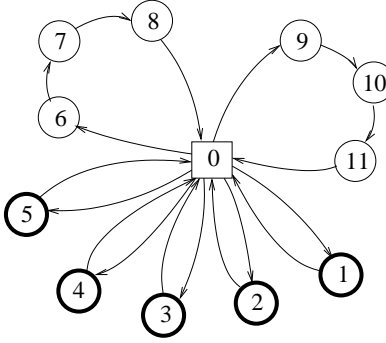


Figure 6: Mobile vertices are vertices 1 to 5. Mobile vertex i has a demand of x_i . The best neighbor solution has a cost of 13 if and only if there is a partition of $\{x_1, x_2, x_3, x_4, x_5\}$.

6 Conclusion

We have introduced an exponential size neighborhood for the VRP which can be explored in polynomial time using flows techniques. The neighborhood that we considered is an extension of the one proposed by Gutin [Gut99] for the traveling salesman problem. Given the relation between the two problems it was natural to investigate from a theoretical point of view whether its exploration can be done in polynomial time in the case of the VRP.

We showed that the exploration of the multi-insertion neighborhood is polynomial if each customer asks for the same quantity of goods, no matter the distance matrix between customers is (distance matrix symmetric or not, which fulfills or not the triangular inequality). In the case where customers ask for different quantities of goods, the problem becomes \mathcal{NP} -hard, even if all the distances (between each couple of customers, and between the depot and each customer) are equals.

We can also notice that, whereas the restricted complete matching problem in a bipartite graph (in which the matching can contain at most r_i edges of the set of edges E_i) is \mathcal{NP} -hard in the general case [IRT78], and that it is polynomial when there is only one constraint (i.e. there is only one set E_1 in which we can choose at most r_1 edges), this problem can be solved in polynomial time if the set of edges $\{E_i\}$ corresponds to a partition of V_2 (and this, even if each edge has a weight and that we look for a minimum-weight matching).

References

- [ABPW06] E. Angel, E. Bampis, F. Pascual, and J. Warnier. Restricted matchings and efficient exploration of an exponential neighborhood for the vehicle routing problem. Technical Report IBISC RR 2006-01, IBISC, Université d'Evry Val d'Essonne, France, 2006.
- [AEOP02] R.K. Ahuja, Ö. Ergun, J.B. Orlin, and A.P. Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, 123:75–102, 2002.

- [AL97] E. Aarts and J.K. Lenstra. *Local Search in Combinatorial Optimization*. Wiley, 1997.
- [AMO93] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
- [CCPS97] W.J. Cook, W.H. Cunningham, W.R. Pulleybank, and A. Schrijver. *Combinatorial optimization*, chapter 5. Wiley, 1997.
- [CPvdV02] R.K. Congram, C.N. Potts, and S.L. van de Velde. An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. *INFORMS Journal on Computing*, 14(1):52–67, 2002.
- [DW00] V.G. Deineko and G.J. Woeginger. A study of exponential neighborhoods for the travelling salesman problem and for the quadratic assignment problem. *Mathematical Programming Ser. A* 87, pages 519–542, 2000.
- [EOAF02] O. Ergun, J.B. Orlin, and A.S-Feldman. Creating very large scale neighborhoods out of smaller ones by compounding moves: A study on the vehicle routing problem. *MIT Sloan Working Paper No 4393-02*, October 2002.
- [GJ79] M.R. Garey and D.S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness*. W. H. Freeman, San Francisco, 1979.
- [GLS93] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1993.
- [Gut99] G. Gutin. Exponential neighbourhood local search for the traveling salesman problem. *Special issue of Computers and Operations Research on the traveling salesman problem*, 26:313–320, 1999.
- [GYZ02] G. Gutin, A. Yeo, and A. Zverovitch. Exponential neighborhoods and domination analysis for the TSP. In G. Gutin and A.P. Punnen, editors, *Traveling salesman problem and its variations*, pages 223–256. Kluwer, 2002.
- [Hur99] J. Hurink. An exponential neighborhood for a one-machine batching problem. *OR Spectrum*, 21(4):461–476, 1999.
- [IRT78] A. Itai, M. Rodeh, and S.L. Tanimoto. Some matching problems for bipartite graphs. *Journal of the ACM*, 25:517–525, 1978.
- [TP93] P.M. Thompson and H.N. Psaraftis. Cyclic transfer algorithms for multivehicle routing and scheduling problems. *Operations Research*, 41:935–946, 1993.
- [TV01] P. Toth and D. Vigo. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematic Press, 2001.
- [XK96] J. Xu and J.P. Kelly. A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science*, 30:379–393, 1996.