



**HAL**  
open science

# A heuristic algorithm for a vehicle routing problem with pickup & delivery and synchronization constraints

Seddik Hadjadj, Hamamache Kheddouci

## ► To cite this version:

Seddik Hadjadj, Hamamache Kheddouci. A heuristic algorithm for a vehicle routing problem with pickup & delivery and synchronization constraints. 9th International Network Optimization Conference (INOC), Jun 2019, Avignon, France. 10.5441/002/inoc.2019.21 . hal-02155536

**HAL Id: hal-02155536**

**<https://hal.science/hal-02155536>**

Submitted on 13 Jun 2019

**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.

# A heuristic algorithm for a vehicle routing problem with pickup & delivery and synchronization constraints

Seddik Hadjadj

Laboratoire d'informatique en image et systèmes  
d'information  
Villeurbanne, France  
mohamed-seddik.hadjadj@liris.cnrs.fr

Hamamache Kheddouci

Laboratoire d'informatique en image et systèmes  
d'information  
Villeurbanne, France  
hamamache.kheddouci@univ-lyon1.fr

## ABSTRACT

In this paper, we consider a vehicle routing problem with pickup & delivery and synchronization constraint. One vehicle with a known and finite capacity has to visit  $n$  customers to pickup or deliver empty containers. At the same time, another vehicle has to deliver ready-mixed concrete by pouring it into the previously delivered containers. This implies dealing with capacity and temporal precedence constraints.

We propose a heuristic to tackle this problem. A two-step approach including a local search and a constructive algorithm. We provide some experiments that show positive results.

## 1 INTRODUCTION

This work is carried out in collaboration with a company which specializes in the sale of ready-mixed concrete.

Today, each ready-mixed concrete order requires the mobilization of one or more *mixer trucks*, even for very small quantities of concrete. However, such trucks are cumbersome, expensive, and disproportionate in some cases, especially when delivering small quantities of concrete.

Therefore, the company proposes a new delivery method to deal more effectively with such orders. The idea is to share a single mixer truck by several customers with small quantities ( $\leq 500$  liters), which implies organizing optimized mixer truck tours.

On the other hand, to ensure the profitability of such truck tours, waiting times at each customer's location have to be reduced. Today, a mixer truck has to wait until the concrete is poured on site to leave a customer's location, and this causes a huge waste of time. To tackle this problem, the company proposes to pour the concrete from the truck into special containers instead of pouring it directly on site which is more difficult and takes more time. The truck can then leave faster, and the customer can use the concrete in the containers all day long. Waiting times are then drastically reduced.

However, since the containers which are supposed to contain the concrete are special, they also must be delivered to the customer. This implies organizing another tour to deliver the containers and pick them up after they have been used.

In brief, this new method is a three-step process :

- (1) A vehicle delivers a number of empty containers to the customer ;
- (2) Thereafter, a mixer truck delivers a certain quantity of ready-mixed concrete by pouring it into the containers delivered ;
- (3) The next day, the vehicle returns to the customer to pick up the containers after they have been used.

To ensure the profitability of this method, the company needs a decision support system that can generate two efficient and synchronized vehicle tours : a pickup & delivery tour for the containers, and a mixer truck tour to deliver the concrete, knowing that each customer has to receive the containers before the concrete (temporal precedence constraint), and that the vehicle carrying the containers has a maximal capacity (capacity constraint).

This paper aims to provide an efficient heuristic to build such synchronized vehicle tours minimizing the total travel times.

The paper is structured as follows. Section 2 provides a literature review of vehicle routing problems with pickup & delivery. Section 3 gives a formulation for the problem tackled in this work. In section 4, we present our heuristic. Section 5 is devoted to some experimental results, and section 6 concludes the paper.

## 2 LITERATURE REVIEW

We present a brief review of pickup & delivery problems.

### 2.1 Pickup & Delivery Problems

There are three main classes of pickup & delivery problem in the literature :

**2.1.1 One-to-One Problems.** One or more vehicles have to carry  $n$  commodities, where each commodity has an origin and a destination. One of the best known examples of this class is the *Dial-a-ride problem* which consists of transporting people from an origin to a destination. The problem has been studied for both single [13] and multiple [4] vehicle cases, with various types of constraints related to ride times, time windows [5, 14]...

**2.1.2 One-to-Many-to-One Problems.** Commodities are divided into "delivery commodities" and "pickup commodities". One or more vehicles have to carry the delivery commodities from the depot to the customers and the pickup commodities from the customers to the depot. Assuming that  $n_p$  is a set of pickup customers, and  $n_d$  a set of delivery customers, two cases have been distinguished for these problems : single demands, where  $n_p \cap n_d = \emptyset$ , and combined demands, where  $n_p \cap n_d \neq \emptyset$ . For the latter case, [7] consider various possible path types such as the *Hamiltonian* path, where each customer is visited once such that pickup and delivery are performed simultaneously, as well as the *double path*, where each customer that has a combined demand (pickup and delivery) is visited twice, the first time for pickup, the second for delivery. Several heuristics have been proposed for both path types for the single and the multi-vehicle cases [3, 12]...

**2.1.3 Many-to-Many Problems.** One or more vehicles have to transport goods between customers knowing that each customer can be a source or a destination of any type of good. Among the problems of this class, the *One-Commodity pickup and delivery*

*travelling salesman problem* is the variant that we consider in this work. It was introduced in [10]. A single vehicle with a known and finite capacity has to carry a single commodity between pickup customers and delivery customers. Picked up commodities can be supplied to delivery customers. This problem is known to be *NP-Hard*. Moreover, checking the existence of a feasible solution is an *NP-Complete* problem [8]. Studies on such problems are relatively scarce. A branch and cut algorithm has been proposed in [10] for small instances, and two heuristics have been developed in [11] to tackle larger instances, in particular by defining "the infeasibility of a path", and adapting the nearest neighbourhood heuristic to increase the chance of obtaining a feasible solution. Furthermore, [9] have proposed a hybrid method combining GRASP (greedy randomized adaptive search procedure) and VND (variable neighbourhood descent) metaheuristics. This method gave better results than the previously proposed ones. For a detailed survey on pickup and delivery problems, we refer the reader to [1].

### 3 PROBLEM FORMULATION

Given two vehicles  $V_1$  and  $V_2$  such that :

- $V_1$  is in charge of delivering (or picking up) empty containers, and has a known and finite maximum capacity  $Q$  ( $Q$  is the maximum number of containers that can be carried by the vehicle) ;
- $V_2$  is a mixer truck carrying a sufficient quantity of concrete.

And considering :

- $D_1$  the depot of  $V_1$  ;
- $D_2$  the depot of  $V_2$  ;
- $N = \{1, \dots, n\}$  a set of  $n$  customers who require a visit of  $V_1$  and/or  $V_2$  ;
- $N = N_p \cup N_d$ , where :
  - $N_d$  is the set of customers who require delivering containers + concrete (who require a visit of both  $V_1$  and  $V_2$ ) ;
  - $N_p$  is the set of customers who require picking up containers (who require a visit of  $V_1$  only) ;
  - $N_p \cap N_d = \emptyset$ .

The problem can be defined on a complete graph  $G = (V, E)$  as follows (see Fig 1) :

- $V = \{D_1\} \cup \{D_2\} \cup N$  is a set of  $n + 2$  nodes ;
- $E = \{(i, j), i, j \in V, i \neq j\}$  is a set of edges representing connections between nodes ;
- $C = \{c_{i,j}, (i, j) \in E\}$  represents the travel time between  $i$  and  $j$  ( $c_{i,j} = c_{j,i}, \forall (i, j) \in E$ ) ;
- $D = \{d_i, i \in N\}$  is a set of customers' demands ( $|d_i|$  is the number of containers to deliver to / pick up from customer  $i$ ,  $d_i < 0 \quad \forall i \in N_d$  and  $d_i > 0 \quad \forall i \in N_p$ ) ;

Assuming that :

- $x_{i,j}$  is a boolean variable such that :
    - $x_{i,j} = 1$  if  $j$  is visited immediately after  $i$  by  $V_1$ ,
    - 0 otherwise.
  - $y_{i,j}$  is a boolean variable such that :
    - $y_{i,j} = 1$  if  $j$  is visited immediately after  $i$  by  $V_2$ ,
    - $y_{i,j} = 0$  otherwise.
- (Note that  $y_{i,j} = 0 \quad \forall i \in N_p, \forall j \in N_p$ ).
- $q_i$  is the number of containers in  $V_1$  after his visit to customer  $i$  (the initial number of containers in  $V_1$  when leaving the depot  $D_1$  is  $q_{D_1} = Q_{init}$ ) ;

- $t_{1,i}$  represents the departure time of  $V_1$  from customer  $i$  location ( $t_{1,0}$  represents de departure time of  $V_1$  from the depot  $D_1$ ) ;
- $t_{2,i}$  represents the departure time of  $V_2$  from customer  $i$  location ( $t_{2,0}$  represents de departure time of  $V_2$  from the depot  $D_2$ ).

The objective is to find two optimized vehicle tours  $T_{V_1}$  and  $T_{V_2}$  minimizing the total travel times, such that  $T_{V_1}$  is a pickup & delivery tour through  $n$  customers, and  $T_{V_2}$  is a concrete delivery tour through the  $n_d$  customers who have received containers. Thus, we consider the following objective function :

$$\min \sum_{i=0}^n \sum_{j=0}^n x_{i,j} c_{i,j} + \sum_{i=0}^n \sum_{j=0}^n y_{i,j} c_{i,j} \quad (1)$$

Subject to :

$$\sum_{j \in N} x_{i,j} = 1 \quad \forall i \in \{D_1\} \cup N \quad (2)$$

$$\sum_{i \in N} x_{i,j} = 1 \quad \forall j \in \{D_1\} \cup N \quad (3)$$

$$\sum_{j \in N_d} y_{i,j} = 1 \quad \forall i \in \{D_2\} \cup N_d \quad (4)$$

$$\sum_{i \in N_d} y_{i,j} = 1 \quad \forall j \in \{D_2\} \cup N_d \quad (5)$$

$$x_{i,D_2} = 0 \quad \forall i \in \{D_1\} \cup N \quad (6)$$

$$x_{D_2,i} = 0 \quad \forall i \in \{D_1\} \cup N \quad (7)$$

$$y_{i,j} = 0 \quad \forall i, j \in \{D_1\} \cup N_p \quad (8)$$

$$x_{i,j}(q_i + d_j - q_j) = 0 \quad \forall i, j \in \{D_1\} \cup N \quad (9)$$

$$q_i \leq Q \quad \forall i \in \{D_1\} \cup N \quad (10)$$

$$q_i \geq 0 \quad \forall i \in \{D_1\} \cup N \quad (11)$$

$$q_{D_1} = Q_{init} \quad (12)$$

$$x_{i,j}(t_{1,i} + c_{i,j} - t_{1,j}) = 0 \quad \forall i, j \in \{D_1\} \cup N \quad (13)$$

$$y_{i,j}(t_{2,i} + c_{i,j} - t_{2,j}) = 0 \quad \forall i, j \in \{D_2\} \cup N_d \quad (14)$$

$$t_{2,i} \geq t_{1,i} \quad \forall i \in \{D_2\} \cup N_d \quad (15)$$

$$t_{1,0} = 0 \quad (16)$$

Where :

- Constraints (2) and (3) ensure that each customer is visited exactly once by vehicle  $V_1$ , while constraints (4) and (5) ensure that each "delivery customer" is visited exactly once by vehicle  $V_2$  ;
- Constraints (6) and (7) relate to the fact that  $V_1$  cannot visit the depot of  $V_2$ , while (8) ensures that  $V_2$  cannot visit neither the depot of  $V_1$  nor the "pickup customers" ;
- Constraints (9) to (12) are related to vehicle capacity. If customer  $j$  is visited immediately after customer  $i$  ( $x_{i,j} = 1$ ), then, the condition  $q_j = q_i + d_j$  must be satisfied. Furthermore,  $q_i$  must be smaller than  $Q$  and greater than 0 ;
- Constraint (13) and (14) concern the computing of departure times of  $V_1$  and  $V_2$  from each customer's location. Thus, if customer  $j$  is visited by vehicle  $m$  immediately after customer  $i$ , then  $t_{m,j} = t_{m,i} + c_{i,j}$  ;
- Constraint (15) concern the temporal precedence between  $V_1$  and  $V_2$ . The vehicle  $V_2$  cannot arrives at a customer's location before  $V_1$ . In other words,  $t_{2,i} \geq t_{1,i}$ .

Note that Picked up containers can be supplied to a delivery customer if necessary.

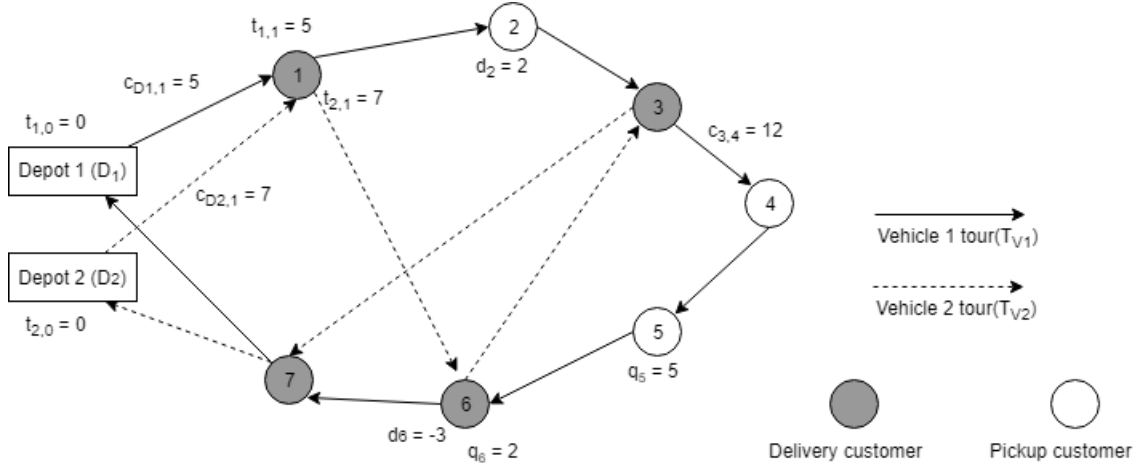


Figure 1: Synchronized vehicle tours.

## 4 PROPOSED HEURISTIC

To tackle the problem described above, we propose a two-step heuristic :

- (1) We generate a feasible pickup & delivery tour for the vehicle  $V_1$  ( $T_{V_1}$ ) using the local search approach described below ;
- (2) Then, we build a tour for  $V_2$  ( $T_{V_2}$ ) taking  $T_{V_1}$  as a strong constraint.

### 4.1 Generating the pickup & delivery tour

The pickup & delivery problem tackled here is the one-commodity pickup & delivery traveling salesman problem. We have a single vehicle ( $V_1$ ) picking up or delivering a single type of commodity (empty containers). A picked up container can be supplied to another customer during a tour, and the vehicle has a maximum capacity that cannot be exceeded during a tour.

We propose a local search method which starts from an initial feasible solution  $S$ , and tries to improve it by moving to  $S'$ , a feasible neighbouring solution of  $S$ , such that  $f(S') < f(S)$ , where  $f(S)$  is the total travel time of  $V_1$ . The process is repeated until no improvement can be found.

**4.1.1 Neighbourhood Structure.** We use the 1-shift algorithm introduced in [2] to generate the neighbourhood of a given solution  $S$ . This method consists in changing the position of a customer in a tour from  $i$  to  $j$ . Customers who are in positions  $i + 1, i + 2, \dots, j$  of the tour are then shifted backwards (see Fig. 2).

**4.1.2 Feasibility Checking.** For each generated solution, we ensure that capacity constraints are respected. A feasible solution is a tour in which the number of containers loaded on the vehicle  $V_1$  never exceeds its maximum capacity  $Q$ , and is never negative. Fig.2 presents an example of a feasible and an infeasible solution. Given a feasible solution  $S$  and a 1-shift neighbouring solution  $S'$  of  $S$  obtained by shifting a customer from position  $i$  to  $j$ . It can easily be shown that  $S'$  is feasible if and only if the partial tour from customer  $i$  to customer  $j$  is feasible. Indeed, to check the feasibility of a neighbouring solution, we only check the feasibility of the tour between position  $i$  and position  $j$ .

### 4.2 Generating the mixer truck tour

Once the pickup & delivery tour for the vehicle  $V_1$  is generated, we build a second tour for  $V_2$  considering the first one as a strong constraint. Thus, starting from  $D_2$ , the idea is to choose, at each iteration of the procedure, the next customer to be visited. So, as it appears in Fig 3, among all customers who require a visit of  $V_2$  :

- We identify those who can be visited by  $V_2$  after the departure of  $V_1$ . In other words, when  $V_2$  is at customer  $i$  location, we calculate  $t_{2,i} + c_{i,j}$  for each customer  $j$  who requires a visit. We choose the next customer from those for whom  $t_{2,i} + c_{i,j} \geq t_{1,j}$  (temporal precedence constraint) ;
- Among all the customers for whom the temporal precedence constraint is respected, we choose the nearest one (in terms of travel time) from the current position of the vehicle ;
- This procedure is repeated until all the customers are visited.

## 5 COMPUTATIONAL RESULTS

The approach described above was implemented in Java, and executed on AMD A10-7700K Radeon R7, 3.40 GHz With 8 GB RAM.

To the best of our knowledge, there is no benchmark instances for simultaneous vehicle routing problems with pickup & delivery. Therefore, we tested our algorithm on the Euclidian PDTSP instances generated by [6], which consider a single depot for each instance. The number of customers varies between 25 and 200. We adapted the instances to fit our constraints by considering the depot and the first customer of each instance as the depots of the two vehicles considered in our problem.

Table 1 shows the average results obtained by the pickup & delivery tour and the mixer truck tour. Pickup & delivery tours are more costly because they involve more customers. In the other hand, they are more flexible since they are not subject to temporal precedence constraint, contrary to mixer truck tours. Therefore, we can hope to obtain better results when focusing on the optimization of the pickup & delivery tours.

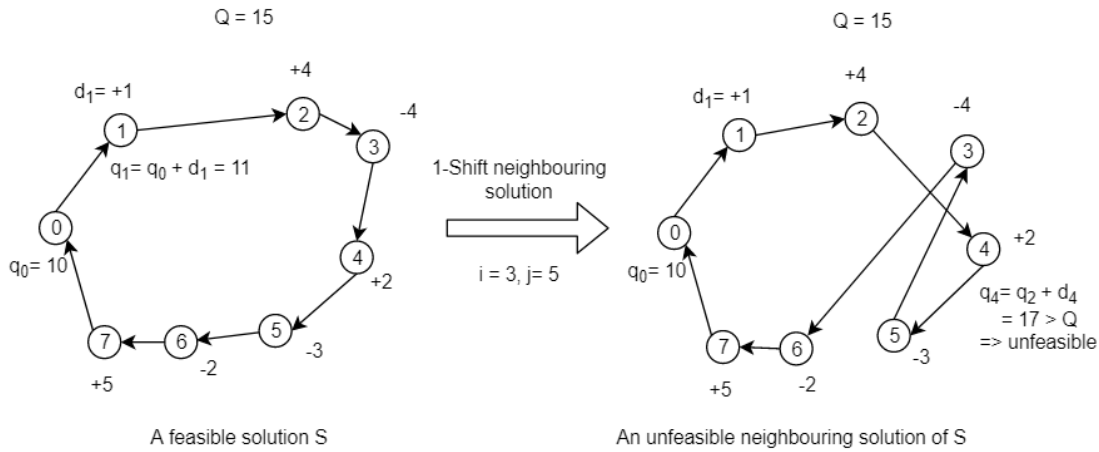


Figure 2: 1-Shift algorithm

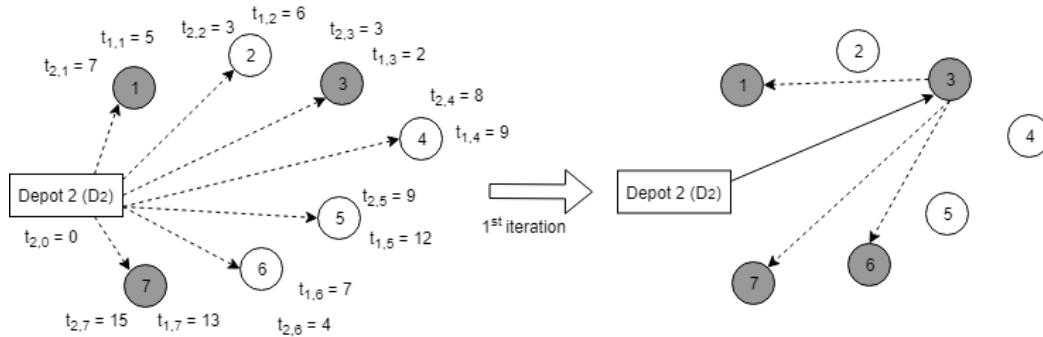


Figure 3: Building a mixer truck tour

Table 1: Average results on the Euclidian PDTSP instances

Number of customers	Pickup & delivery tour	Mixer truck tour
25	564.36	363.19
50	871.33	573
75	1143.1	779.24
100	1429.71	1001.41
150	2019.18	1613.86
200	2704.91	1813.22

## 6 CONCLUSIONS

We presented an approach to tackle a vehicle routing problem with pickup & delivery and synchronization constraint. This approach is a two-step heuristic. We start by generating a pickup & delivery tour for a first vehicle respecting vehicle capacity constraint. Then, we construct a second tour according to the first one for another vehicle, respecting temporal precedence constraint. The objective function considered is the minimization of the total travel times.

We tested our algorithms on the Euclidian PDTSP instances proposed in [6]. We adapted the instances to fit our constraints and collected the results, which were positive.

Future works will be devoted to the implementation of the ILP model proposed in this paper and the development of other approaches exploiting other types of heuristics, and including other constraints such as time windows, multiple vehicles...

## REFERENCES

- [1] Gerardo Berbeglia, Jean-François Cordeau, Irina Gribkovskaia, and Gilbert Laporte. 2007. Static pickup and delivery problems: a classification scheme and survey. *Top* 15, 1 (2007), 1–31.
- [2] Dimitris Bertsimas. 1988. *Probabilistic combinatorial optimization problems*. Ph.D. Dissertation. Massachusetts Institute of Technology.
- [3] Jeng-Fung Chen and Tai-Hsi Wu. 2006. Vehicle routing problem with simultaneous deliveries and pickups. *Journal of the Operational Research Society* 57, 5 (2006), 579–587.
- [4] Jean-François Cordeau. 2006. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research* 54, 3 (2006), 573–586.
- [5] Jean-François Cordeau and Gilbert Laporte. 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological* 37, 6 (2003), 579–594.
- [6] Michel Gendreau, Gilbert Laporte, and Daniele Vigo. 1999. Heuristics for the traveling salesman problem with pickup and delivery. *Computers & Operations Research* 26, 7 (1999), 699–714.
- [7] Irina Gribkovskaia, Øyvind Halskau sr, Gilbert Laporte, and Martin Vlček. 2007. General solutions to the single vehicle routing problem with pickups and deliveries. *European Journal of Operational Research* 180, 2 (2007), 568–584.
- [8] H Hernández-Pérez. 2004. Traveling salesman problems with pickups and deliveries. *Dissertation, University of La Laguna, Spain* (2004).

- [9] Hipólito Hernández-Pérez, Inmaculada Rodríguez-Martín, and Juan José Salazar-González. 2009. A hybrid GRASP/VND heuristic for the one-commodity pickup-and-delivery traveling salesman problem. *Computers & Operations Research* 36, 5 (2009), 1639–1645.
- [10] Hipólito Hernández-Pérez and Juan-José Salazar-González. 2004. A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics* 145, 1 (2004), 126–139.
- [11] Hipólito Hernández-Pérez and Juan-José Salazar-González. 2004. Heuristics for the one-commodity pickup-and-delivery traveling salesman problem. *Transportation Science* 38, 2 (2004), 245–255.
- [12] Arild Hoff and Arne Løkketangen. 2006. Creating lasso-solutions for the traveling salesman problem with pickup and delivery by tabu search. *Central European Journal of Operations Research* 14, 2 (2006), 125–140.
- [13] Harilaos N Psaraftis. 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science* 14, 2 (1980), 130–154.
- [14] Paolo Toth and Daniele Vigo. 1996. Fast local search algorithms for the handicapped persons transportation problem. In *Meta-Heuristics*. Springer, 677–690.