A Heuristic and Split Procedure for Dial-a-Ride Problems

Philippe Lacomme, Alain Quilliot, Xiagang Zhao

Laboratoire d'Informatique (LIMOS, UMR CNRS 6158)
Campus des Cézeaux, 63177 Aubière CEDEX

placomme@sp.isima.fr, quilliot@sp.isima.fr, zhao@sp.isima.fr

Abstract

This paper addresses the dial-a-ride problem which consists in pickup nodes by designing vehicles trips to satisfy the delivery nodes of the serviced customers. Time-Windows are provided and represent user requirement in both pickup nodes and delivery nodes. A fleet of homogeneous vehicles is assumed to be available at a depot node. The work presented focuses only on DARP with homogenous fleet of vehicles and permits to address Cordeau and Laporte's instances which are homogenous ones.

Keywords: DARP, routing, split

1. Introduction

The DARP is a well known NP-hard problem. Earliest relevant publications can be cited in 1980 including [8] which focus in the single vehicle problem. Heuristic approaches have been proposed including [1], [3], [6], [9], [11]. But few publications focus on real-life instances. One can note the contribution of [10] who tackles dial-a-ride problem in Bologna and [2] which addresses the transportation of handicapped in Berlin. The DARP which is addressed here is defined on a complete graph $G = (V, A)$, weighted and directed network with a capacitated homogeneous fleet of vehicles and a number of requests. This study focuses on the static DARP.

The following notations are used for data:

$V = \{v_0, v_1, ..., v_{2n}\}$ Set of node including the depot v_0, the pickup nodes $\{v_1, ..., v_n\}$ and the delivery nodes $\{v_{n+1}, ..., v_{2n}\}$

Q Vehicles capacity (homogenous fleet)

t_{ij} Transportation time from node i to j

c_{ij} Transportation cost from node v_i to node v_j

e_i Earliest starting time of service at node v_i

l_i Latest starting time of service at node v_i

q_i The number of customers to service at node v_i

s_i Service duration

In the model the following variables are used:

A_i The arrival time of a vehicle at the node v_i

B_i The beginning of service at the node v_i

D_i The departure time of a vehicle at the node v_i with $D_i = B_i + s_i$

W_i The waiting time before beginning of service at node v_i, $W_i = B_i - A_i$

R_i The riding time is the time between the end of service at origin v_i and the beginning of service at delivery v_{n+i} as $R_i = B_{n+i} - D_i$

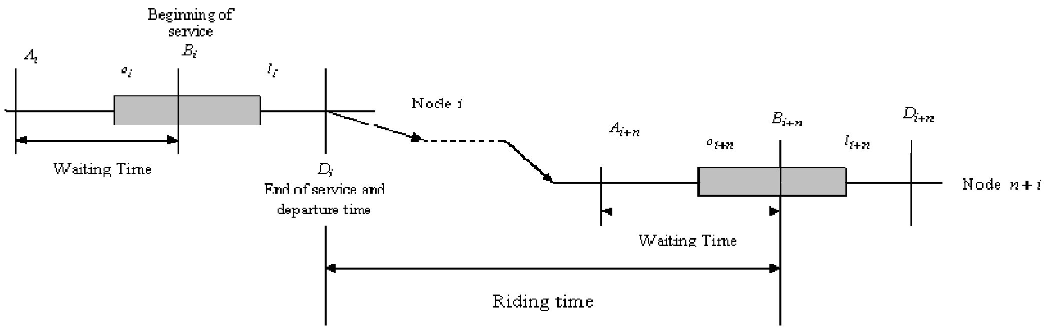

Figure 1 Description and notations

Each customer must be picked up at the pickup node and dropped off at the delivery node. Each trip start and end at the same depot. At any time and for any node v_i, the number of customers q_i is less than the vehicle capacity and the beginning of service must satisfy the time windows i.e. $B_i \in [e_i, l_i]$. A waiting time is allowed before beginning of service at any node v_i, but there is no waiting time after end of service. The end time of service is equal to the departure time. The objective consists in routing the K vehicles by a careful definition of arrival time A_i at each node and departure time D_i and by assigning a permutation node list λ to each vehicle k ($\lambda(i)$ is the i^{th} customer to service, $\lambda(n_k)$ the last customer to service). For each vehicle k, $B(k)$ is the departure time of depot and $E(k)$ is the return time of depot.

The following criteria are linked to a solution:

The total riding time: $TRT = \sum_{i=1}^{n} (B_{i+n} - D_i)$;

The total trip duration: $TD = \sum_{k=1}^{|K|} E(k) - B(k)$;

The total waiting time: $TWT = \sum_{i=1}^{2n} (B_i - A_i) = \sum_{i=1}^{2n} W_i$ where $W_i = (B_i - A_i)$. A solution must satisfy the following constraints:

(C1) the number of customers in the vehicle is upper bounded by the vehicle capacity Q

(C2) the beginning of service at node v_i satisfies the customer time windows i.e. $e_i \le B_i \le l_i$

(C3) the total riding time (TRT) is less than Max_{TRT} (the maximal riding time)

(C4) the trip duration (TD) is less than Max_{TD} (the maximal trip duration)

The objective consists in minimizing (satisfying (C1), (C2), (C3) and (C4)): the total duration (TD); the total riding time (TRT) and the total waiting time (TWT). If there is no feasible solution (no solution can comply with (C1), (C2), (C3) and (C4)), a solution minimizing the violation of constraints is required. The DARP is a multi-objective optimization problem which extends the VRP by extra constraints including for instance time-windows.

2. A Split procedure for a permutation node list evaluation

2.1. Cordeau and Laporte's evaluation for trip

Cordeau and Laporte lastly introduced a powerful evaluation procedure [4] to assign $z = (TRT, TD, TWT, VTW, VRT, VC, VTD)$ (where TRT is the total riding time, TD the total route duration, TWT the total waiting time, VTW the total violation of time windows, VRT the total violation of riding time, VC the total violation of vehicle capacity and VTD the violation of route duration) at one permutation node list $\lambda = (i_1, \ldots, i_k)$ where $i_1 = i_k$ is the depot node. The key features of the evaluation [4] consist in reducing trip duration and waiting time by delaying the departure time of depot and the beginning time of service at nodes.

2.2. Node permutation list decomposition in trips

The decomposition of a permutation node list $\lambda = (i_1, \ldots, i_{2n})$ depends on the succession of pickup and delivery nodes in the sequence. The decomposition of $\lambda = (i_1, \ldots, i_{2n})$ can be addressed by a function $f(\lambda) = (f(i_1), \ldots, f(i_{2n}))$ defined as follows: $f(i_k) = \sum_{p=1}^{k} g(i_p)$ where $g(i_p) = q_{\lambda_{i_p}}$ if $\lambda_{i_p} \leq n$ and $g(i_p) = -q_{\lambda_{n-i_p}}$ if $\lambda_{i_p} > n$. $f(i_p) = 0$ represents a trip delimiter in the permutation node list $\lambda = (i_1, \ldots, i_{2n})$ i.e. a null load of vehicle. It is possible to define $\lambda^* = (i_1^*, \ldots, i_p^*)$ a permutation node list of trip delimiters linked to $\lambda = (i_1, \ldots, i_{2n})$.

2.3. Split proposal

Split procedures have been successfully applied to numerous routing problems and have greatly contributed to definition of efficient algorithms. These successful application domains include the memetic algorithm of [5] for the CARP and the genetic algorithm of [7] for the VRP. A permutation node list $\lambda = (i_1, \ldots, i_{2n})$ where λ_i is the ith node to service can be efficiently splitted in sub-trips using $\lambda^* = (i_1^*, \ldots, i_p^*)$. The Split procedure works on an auxiliary graph $H = (X; A; Z)$. H is a set of $2n+1$ nodes indexed from 0 to $2n$. An arc from node i to node j represents a trip from node λ_i to node λ_j. The weight z_{ij} of (i, j) is equal to the trip cost and z_{ij} is obtained using Cordeau and Laporte's heuristic [4]. A node label $L_i^p = (K_i^p, TRT_i^p, TD_i^p, TWT_i^p, VTW_i^p, VRT_i^p, VC_i^p, VTD_i^p, k, j)$ is the Pth label assign to the node i and it is composed of: K the number of vehicle available; the criteria computed using Cordeau and Laporte's procedure and (k, j) the father label of L_i^p we mean L_j^k.

The initial label of node 0 is $L_0^1 = (K, 0,0,0,0,0,0,0, -1, -1)$ which represent a solution where $K_0 = K$ vehicle are available, and all the cost and constraint violation are stated to 0. (-1;-1) means this initial label has no predecessor in the graph. Each label $L_i^p = (K_i^p, z_i^p, k, j)$ with $z_i^p = (TRT_i^p, TD_i^p, TWT_i^p, VTW_i^p, VRT_i^p, VC_i^p, VTD_i^p)$ generates $L_j^q = (K_j^q, z_j^q, k, j)$ using arc (i, j) and the weight z_{ij} is computed by Cordeau and Laporte's algorithm. The new label is obtained using the following formulae: $K_j = K_i - 1$, $z_j^q(k) = z_i^p(k) + z_{ij}(k)$. Trying to avoid excessive label generation, dominated feasible trips are discarded thanks to the following domination rule. A label $L_i^p = (K_i^p, z_i^p, k, j)$ is state as **dominant** as regards $L_j^q = (K_j^q, z_j^q, k, j)$ if one of the following condition holds: $K_i > K_j$ and $\forall k = 1..7, z_i^p(k) \le z_j^q(k)$ or $\exists k \in 1..7, z_i^p(k) < z_i^q(k)$ and $\forall v = 1..p, v \ne q, z_i^p(v) < z_i^q(v)$.

Thanks to the previous definition, an optimal split of the permutation customer list $\lambda = (i_1, \ldots, i_n)$ can be obtained by saving only non dominated label on each node of the graph $H = (X; A; Z)$. Note also, that due to trips constraints only arc $(i; j)$ where $f(\lambda_i) = f(\lambda_j) = 0$ have to be addressed. A DARP solution for $\lambda = (i_1, \ldots, i_{2n})$ corresponds to a min-cost path from 0 to $2n$ in H. Due to the dominance rule, a permutation node list $\lambda = (i_1, \ldots, i_{2n})$ can induce several labels each representing one feasible split of the list with combination of the 7 criteria.

3. Heuristics for the DARP

The heuristic we provide does not tackle the limitation on the number of vehicles (C1). Trips are generated by insertion technique. A new trip is created when a new pickup node can not be inserted into the current one due to constraints C1, C2, C3, and C4. The heuristic provides a set of trips and a permutation node list $\lambda = (i_1, \ldots, i_{2n})$ by concatenation of the trips.

Nodes of V are used to generated a permutation node list $L = (i_1, \ldots, i_{2n})$ which is an array of nodes sorted in l_i increasing order. Let us define:

$p(i)$ the position in L of the node i which imply.

$p(L(i)+n)$ = position of the delivery node of customer $L(i)$ if $L(i) \le n$.

$p(L(i)-n)$ = position of the pickup node of customer $L(i)$ if $L(i) > n$.

$L^*(i)$ / $L^*(i) = L(p(L(i)+n))$ if $L(i) \le n$ and $L^*(i) = L(p(L(i)-n))$ if $L(i) > n$.

$\delta_i = (L(i); L^*(i))$ if $L(i) \le n$ and $\delta(i) = (L^*(i); L(i))$ if $L(i) > n$.

At each iteration k, the node $\delta_i(1)$ and the node $\delta_i(2)$ are simultaneously added in the current trip t. All insertion positions in the current trip t are investigated firstly for $\delta_i(1)$ and secondly for $\delta_i(2)$. If t is a trip with q nodes (including depot node in first position and depot node in the last position), there is $q-1$ insertion positions for $\delta_i(1)$. The insertion of $\delta_i(1)$ and $\delta_i(2)$ in position p_1 and p_2 in trip t is achieved using the procedure: t'=Insert(t, $\delta_i(1)$, $\delta_i(2)$, p_1, p_2). For all insertion positions investigated, the Cordeau and Laporte's evaluation [4] is used to compute $z = (TRT, TD, TWT, VTW, VRT, VC, VTD)$. The trip t' is feasible if $TD \le \max_{TD}$ and $VTW = VRT = VC = 0$. If there are many insertion positions leading in feasible trip t', the best trip t' is the trip minimizing the trip duration (TD). If all insertion positions lead to not feasible trip t', $\delta_i = (\lambda(i); \lambda^*(i))$ is skipped and the next node in the sorted permutation node list $L = (i_1, \ldots, i_{2n})$ is addressed. If all nodes leave to no feasible trip then a new trip is created.

4. Numerical experiments

4.1 Implementation and benchmark uses

All procedures are implemented under Borland C++ 6.0 package and experiments were carried out on a 2.4 GHz computer under Windows XP with 2 GO of memory. All test instances are available from the Internet site http://neumann.hec.ca/chairedistributique/data/darp/.

4.2 Numerical experiments

The results obtained by the heuristic are compared to the results obtained by [4] which are represented in Table 1. The results introduce for each instance: the number of vehicles used (K'), the total trip duration (TD), the total waiting time (TWT) and the total riding time (TRT). The heuristic does not provide solutions for all the instances: 5 solutions use a number of vehicles (K') exceeding the number of vehicles available. For 15 instances, the solutions reported on Table 1 outperform the previous published solutions of Cordeau and Laporte [4]. For instance pr01, heuristic provide a total duration (TD), a total time windows (TWT) and a total riding time (TRT) lower than Cordeau and Laporte's results.

Table 1. Heuristic evaluation on Cordeau and Laporte's instances

		Cordeau and Laporte's results						Heuristic					
instance	K	K'	TD	TWT	Avg.	TRT	Avg.	K'	TD	TWT	Avg.	TRT	Avg.
pr01	3	3	881.16	211.15	4.40	1094.99	45.62	2	812.28	44.14	0.92	957.41	39.89
pr02	5	5	1985.94	723.87	7.54	1976.73	41.18	5	1781.51	301.15	3.14	2109.42	43.95
pr03	7	7	2579.35	607.27	4.22	3586.68	49.82	7	2418.13	28.33	0.20	2448.29	34.00
pr04	9	9	3583.22	1090.44	5.68	5021.32	52.31	8	3032.76	4.58	0.02	4028.02	41.96
pr05	11	11	3869.95	832.98	3.47	6156.48	51.30	9	3787.05	127.52	0.53	5097.55	42.48
pr06	13	13	5056.83	1375.42	4.78	7273.49	50.51	13	4666.50	22.01	0.76	5981.90	41.54
pr07	4	4	1452.05	440.34	6.12	1508.93	41.91	4	1224.54	30.20	0.42	1204.08	33.45
pr08	6	6	2345.22	410.33	2.85	3691.53	51.27	6	2371.98	13.21	0.09	3159.43	43.88
pr09	8	8	3155.49	323.05	1.50	5621.77	52.05	*10	3518.83	44.38	0.21	4148.45	38.41
pr10	10	10	4480.10	721.33	2.50	7163.71	49.75	*12	4638.06	47.49	0.16	5803.22	40.30
pr11	3	3	965.06	320.60	6.68	1041.50	43.40	2	784.66	30.70	0.64	926.11	38.59
pr12	5	5	1564.74	308.68	3.22	2393.78	49.86	4	1441.43	0.00	0.00	1737.65	36.20
pr13	7	7	2263.68	330.38	2.29	3788.78	52.62	6	2370.38	20.06	0.14	3147.36	43.71
pr14	9	9	2882.17	426.27	2.22	4632.80	48.26	8	3027.58	0.00	0.00	3945.28	41.10
pr15	11	11	3595.63	605.89	2.52	6104.72	50.87	9	3597.23	0.00	0.00	5060.44	42.17
pr16	13	13	4072.47	448.88	1.56	7347.39	51.02	11	4454.48	0.00	0.00	5974.04	41.49
pr17	4	4	1097.25	129.03	1.79	1761.99	48.94	4	1128.36	2.47	0.03	1579.87	43.89
pr18	6	6	2446.51	523.44	3.64	3659.04	50.82	*7	2363.32	38.06	0.26	2762.09	38.36
pr19	8	8	3249.29	487.33	2.26	5581.02	51.68	*10	3428.98	0.00	0.00	4299.99	39.81
pr20	10	10	4040.99	362.37	1.26	7072.29	49.11	*12	4600.50	1.58	0.01	6088.83	42.28

4.3 Test of insertion heuristic with Split

The heuristic gives both trips and the permutation node list $\lambda = (i_1, \ldots, i_{2n})$ linked to the solution. The permutation list generated by the heuristic is optimally splitted to investigate all non dominated solutions linked to $\lambda = (i_1, \ldots, i_{2n})$. All the non dominated splits of the permutation node list $\lambda = (i_1, \ldots, i_{2n})$ are reported on Table 2 for the instances Pr03 and Pr05. For instance Pr03, there is 1 decomposition without any violations among 51 decompositions. For instance Pr05, there are 7 decompositions without any violations among 132 decompositions. Split procedure, gives new decomposition of the permutation node list $\lambda = (i_1, \ldots, i_{2n})$ with one decomposition complying with the heuristic's solution. Lines in bold in Table 2 are the decomposition obtained thanks to the heuristic.

Table 2. Example of split execution on the node list $\lambda = (i_1, \ldots, i_{2n})$ for 3 instances

	K	K'	TD	TWT	TRT	VTW	VRT	VC	VTD
Pr03	**7**	**7**	**2418.13**	**28.33**	**2448.29**	**0**	**0**	**0**	**0**
	7	7	2426.25	36.29	2444.56	9.82	0	0	0
	7	7	2407.52	17.26	2448.3	9.82	0	0	0
	7	6	2427.53	38.15	2448.29	9.82	0	0	0
	…								
	7	2	4367.46	1978.39	2448.29	77651.48	0	0	3625.87
	7	1	4354.37	1978.39	2448.29	77651.48	0	0	3652.87
pr05	**11**	**9**	**3787.05**	**127.52**	**5097.55**	**0**	**0**	**0**	**0**
	11	10	3780.07	120.27	5092.21	0	0	0	0
	11	10	3783.57	121.21	5091.24	0	0	0	0
	11	10	3678.02	16.33	5097.24	0	0	0	0
	11	11	3776.60	113.96	5085.89	0	0	0	0
	11	11	3671.05	9.08	5092.21	0	0	0	0
	11	11	3674.55	10.01	5091.24	0	0	0	0
	…								
	11	2	6431.26	2779.57	5085.89	127797.4	0	0	5471.26
	11	1	6928.51	3278.19	5085.89	260523.5	0	0	6448.51

5. Concluding remarks

This article addresses the Dial-a-Ride problem using: a heuristic based approach to generation of node sequences and a split algorithm to decomposition of any node permutation list into sub-trips. The trip evaluation is managed thanks to the Cordeau and Laporte's algorithm to provide an easy and powerful computation of vehicle arrival and departure time. The multi-objective approach provided by the split algorithm proves it is possible to obtain several non dominated solutions for one permutation node list. Overall, the heuristic with Split procedure provide promising results and this work is a step forward resolution of the DARP. Our work is now directed into: (i) extensions of the approach to non homogenous fleet of vehicles; (ii) a careful definition of a local search taking advantages of the shortest path in the split auxiliary graph; (iii) a memetic algorithm for permutations node list generation.

References

[1] Bodin, L.D., Sexton, T., The multi-vehicle subscriber dial-a-ride problem. *TIMS Studies in Management Science*, vol. 26, pp. 73–86, 1986.

[2] Borndorfer, R., Grotschel, M., Klostermeier F., Kuttner, C., Telebus Berlin: Vehicle scheduling in a dial-a-ride system. Technical Report SC 97-23, Konrad-Zuse-Zentrum fur Informationstechnik Berlin, 1997.

[3] Dumas, Y., Desrosiers, J., Soumis, F., 1991. The pickup and delivery problem with time windows. *European Journal of Operational Research*. vol. 54, pp. 7–22, 1991.

[4] Cordeau J.F. and G. Laporte. A tabu search heuristic for static multi-vehicle dial-a-ride problem. *Transportation Research*, Part B, vol. 37, pp. 579 – 594, 2003

[5] Lacomme P., C. Prins and W. Ramdane-Cherif, "Competitive Memetic Algorithms for Arc Routing Problems". *Annals of Operations Research*. vol. 131, pp. 159-185. 2004.

[6] Madsen, O.B.G., Ravn, H.F., and Rygaard, J.M. A heuristic algorithm for the dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research*, vol. 60, pp.193–208, 1995.

[7] Prins C. A simple and effective evolutionary algorithm for the vehicle routing problem, *Computers & Operations Research*, vol. 31, pp. 1985-2002, 2004.

[8] Psaraftis H. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, vol. 14 pp. 130-54, 1980.

[9] Rekiek, B., Delchambre, A., and Saleh, H.A. Handicapped person transportation problem: An application of the grouping genetic algorithm. *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 511–520, 2006.

[10 Toth, P., Vigo, D., Fast local search algorithms for the handicapped persons transportation problem. In: Osman, I.H., Kelly, J.P. (Eds.), *Meta-Heuristics: Theory and Applications*. Kluwer, Boston, pp. 677–690, 1996.

[11] Zhihai Xiang, Chengbin Chu, Haoxun Chen. A Fast Heuristic For The Static Dial-A-Ride Problem. *European journal of operational research*. vol. 174, pp. 1117-1139, 2006.