A Cumulative Service State Representation for the Pickup and Delivery Problem with Transfers

Monireh Mahmoudi
School of Packaging, College of Agriculture and Natural Resources, Michigan State University, East Lansing, Michigan, 48824, mahmou18@msu.edu

Junhua Chen
School of Traffic and Transportation, Beijing Jiaotong University, Beijing, P. R. China 100044, cjh@bjtu.edu.cn

Tie Shi
School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, P. R. China 610031, tshi2005@my.swjtu.edu.cn

Yongxiang Zhang
School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, P. R. China 610031, bk20100249@my.swjtu.edu.cn

Xuesong Zhou
School of Sustainable Engineering and the Built Environment, Arizona State University, Tempe, Arizona 85281, xzhou74@asu.edu

Extended Abstract

Transfers are used to provide more efficient transportation networks by reducing the operational costs, as well as making more flexible routes available for passengers and freights. A large number of daily trips are classified in this category. An example of this type of transportation service is the first mile/last mile transport of the commuters who go from an origin to a transit station or from a station to a final destination.

Despite the extensive prior research on the pickup and delivery problem without transfers, few studies have focused on the pickup and delivery problem with transfers. The latter problem is observed in the literature as the pickup and delivery problem with transshipments/transfers (PDPT). Mues and Pickl (2005) developed a path-based mixed integer programing (MIP) model for the PDPT and applied a column generation to solve the model. Cortés et al. (2010) proposed an MIP model for the PDPT in which passengers have different options for transfer from one vehicle to another at particular transfer nodes. In three major papers on this topic by Drexl (2012a, 2012b, 2013), different types of synchronization (i.e., task synchronization, operation synchronization, movement synchronization, load synchronization, and resource synchronization) have been extensively discussed. Recently, Rais et al. (2014) proposed an MIP formulation for the PDPT with/without time windows for services in which heterogeneous vehicles and flexible fleet size are allowed. The MIP model proposed by Rais et al. (2014) solves the problem on passengers' origin/destination-based network. Their model does not work directly with transportation networks in which change in travel time is subject to the time of the day (e.g., high-occupancy toll lanes) or the load of the vehicle (e.g., high-occupancy vehicles lanes). Dealing with several constraints, especially those related to the validity of the time and load variables has prompted us to look at this challenging problem from a different angle.

In this research, we add time dimension to the space graph to not only track the location of vehicles at any time but also impose parcels' pickup and delivery time windows, synchronization time points, and precedence constraints to the problem. Defining time as an explicit dimension and physical transportation networks as the base of our proposed networks help us to handle the foregoing instances of high-occupancy vehicles and high-occupancy toll lanes. We also add another dimension, called the "parcels' cumulative service state" to the constructed space-time graph to track the service status of parcels at any time and impose the coupling and precedence constraints to the model.

To handle a large-scale set of parcels, we develop the traditional cluster-first, route-second approach. We reach optimality for the clusters derived from the original set of parcels. We also propose an integer programming model to improve the vehicles' efficiency. Based on this premise, our contribution is three-fold:

(i) In terms of methodology, we propose a new mathematical model for the PDPT, in which heterogeneous vehicles and flexible fleet size are allowed. Based on our space-time-state network representation, we apply a dynamic programing (DP) to include the vehicle-to-task assignment constraints and provide exact solutions for small-scale problems. Meanwhile, our space-time-state network representation prevents sub-tours which is a common issue in the family of vehicle routing problems.

(ii) More importantly, to address the curse of dimensionality, we demonstrate a consistent transition from the cumulative service states to cumulative flow count diagrams to effectively estimate the overall dynamic system performance. It should be noted that the concept of cumulative flow count diagram is widely applied as a representative of dynamic activities in traffic science literature. The concept of cumulative flow count diagram accounts for the cumulative flow count of vehicles passing through a transportation system, in which vehicle concentrations, queue sizes, travel times, and delays are the main measures of the system performance evaluation (see the papers by Newell, 1982 and Hall, 1991 for a summary of corresponding methodologies).

(iii) With the consistent microscopic and macroscopic system representation, our model and algorithm can effectively handle large-scale real-world instances and generate a good initial solution to be applied for our Lagrangian heuristic method. Then, by our Lagrangian heuristic, we evaluate the marginal cost of each transfer and guide a fast search for real-world test cases with about 10,000 delivery orders.

Computational experiments

We performed extensive numerical experiments over the standard data set used by Ropke and Pisinger (2006) and real-world large-scale data set proposed by Cainiao Network (with about 10,000 delivery orders) to examine the computational efficiency of our developed algorithm. Due to a limit on pages, we just bring the results of the data set by Ropke and Pisinger (2006).

The data set used by Ropke and Pisinger (2006) is the modified version of instances initially introduced by Savelsbergh and Sol (1998). In this data set, parcel p's origin and destination are denoted by node p and node $n + p$, respectively, where n is the total number of parcels in the system. In addition, the coordinates (x and y) of parcels' origin and destination are randomly generated and uniformly distributed over a $[0{,}50] \times [0{,}50]$ square. A single depot is located in $[25{,}25]$. Moreover, $[0{,}1000]$ is considered as the vehicles' time horizon (vehicles' time horizon is assumed to be identical). Feasible departure/arrival time windows are also randomly generated for each parcel.

Each instance in this data set is named by two alphabetical letters (i.e., AA, BB, CC, DD, XX, or YY) and a double-digit number after the alphabetical letters. The two alphabetical letters are representative of the class of the instance, while the double-digit number demonstrates the total number of parcels in the instance. Each class presents the characteristics of its instances including vehicles' capacity, length of parcels' departure time windows, and load of parcels (Table 1). The load of each parcel is randomly generated from a given set (Table 1). We note that the load of each parcel does not exceed the maximum capacity of the vehicles. The capacity of vehicles for instances of a class is the same.

Table 1
The characteristics of the instances of the data set used by Ropke and Pisinger (2006)

Class	Vehicles' capacity	Length of parcels' departure time windows	The set from which the loads are randomly generated
AA	15	60	[5,15]
BB	20	60	[5,20]
CC	15	120	[5,15]
DD	20	120	[5,20]
XX	15	60	[1,15]
YY	15	120	[1,15]

According to Table 1, we expect more levels of complexity in instances XX and YY in comparison to instances AA, BB, CC, and DD due to the wide range of parcel's load. To the best of our knowledge, very few papers have published the results of instances XX and YY. We note that, in this data set, the objective is to serve all parcels with

fewer vehicles. Table 2 presents the results obtained from our algorithm on the instances of the data set used by Ropke and Pisinger (2006). The third and fourth columns of Table 2 from left, denoted by "A" and "B", present the total number of vehicles used to serve all parcels in our algorithm and the algorithm proposed by Ropke and Pisinger (2006), respectively.

To compare the performance of two algorithms in terms of number of vehicles, we define the fifth column of Table 2 from left as the ratio of A to B. The sixth and seventh columns of Table 2 from left, denoted by "C" and "D", present the upper bound of total routing cost obtained from solving the problem by our algorithm and the algorithm proposed by Ropke and Pisinger (2006), respectively. To compare the performance of two algorithms in terms of routing cost, we define the eighth column of Table 2 from left as the ratio of C to D. The ninth column of Table 2 from left, denoted by "E", presents the lower bound of total routing cost obtained from our algorithm. The tenth column of Table 2 from left, calculated by $(\frac{C-E}{C} \times 100)$, presents the gap percentage between lower and upper bounds of our solution. According to Table 2, for the four largest AA instances, for one of the medium-scale CC instances and for all of the DD instances, our heuristic performed substantially better than that of Ropke and Pisinger (2006) (in terms of both number of vehicles and routing cost). The situation was reversed for the four largest BB instances and four of the six largest CC instances, with the method of Ropke and Pisinger (2006) performing substantially better in both respects. On all other instances, the results were similar.

Table 2
A comparison of the results obtained from our algorithm and the algorithm proposed by Ropke and Pisinger (2006).

Name	# of clusters	A	B	$\frac{A}{B}$	C	D	$\frac{C}{D}$	E	Gap %	Computation time (sec)	Computation time (sec) (Ropke and Pisinger (2006))
AA30	5	4	5	0.80	41,316	51,317.40	0.81	41288	0.07	25.6	27
AA35	5	5	5	1.00	51,506	51,343.53	1.00	51244	0.51	44.2	33
AA40	6	6	6	1.00	61,759	61,609.44	1.00	61432	0.53	44.67	41.4
AA45	7	6	6	1.00	62,029	61,693.01	1.01	61542	0.79	52.57	49.8
AA50	8	6	7	0.86	62,213	71,932.03	0.86	61687	0.85	54.25	58.8
AA55	8	8	8	1.00	82,405	82,185.31	1.00	81765	0.78	95.63	64.2
AA60	9	8	9	0.89	82,629	92,366.70	0.89	81936	0.84	94.44	76.8
AA65	10	7	8	0.88	72,783	82,331.12	0.88	72245	0.74	98.3	87.6
AA70	10	8	11	0.73	82,950	112,458.28	0.74	82347	0.73	143.7	98.4
AA75	11	8	9	0.89	83,044	92,529.42	0.90	82545	0.60	141.82	112.8
Average of AA	7.9	6.6	7.4	0.90	68263.4	75976.624	0.91	67803.1	0.64	79.518	64.98
BB30	5	4	5	0.80	41,267	51,193.62	0.81	41065	0.49	28.4	28.2
BB35	5	5	6	0.83	51,580	61,400.07	0.84	51206	0.73	38.2	32.4
BB40	6	5	5	1.00	51,785	51,421.35	1.01	51142	1.24	60.83	44.4
BB45	7	6	6	1.00	61,950	61,787.28	1.00	61632	0.51	71.86	49.2
BB50	8	7	7	1.00	72,164	71,889.75	1.00	71593	0.79	89.5	58.8
BB55	8	8	8	1.00	82,483	82,080.73	1.00	81889	0.72	122.75	64.2
BB60	9	11	10	1.10	112,988	102,323.77	1.10	101998	9.73	122.22	73.8
BB65	10	10	8	1.25	103,211	82,623.98	1.25	82454	20.11	122.9	85.2
BB70	10	11	9	1.22	113,534	92,647.75	1.23	92465	18.56	160.4	100.8
BB75	11	11	9	1.22	113,558	92,476.30	1.23	91897	19.07	154.18	112.8
Average of BB	7.9	7.8	7.3	1.04	80452	74984.46	1.05	72734.1	7.20	97.124	64.98
CC30	5	5	5	1.00	51,358	51,145.18	1.00	50987	0.72	44.2	28.2
CC35	5	5	5	1.00	51,578	51,235.64	1.01	51147	0.84	51.8	34.2
CC40	6	5	6	0.83	51,695	61,473.91	0.84	51213	0.93	49.5	43.2
CC45	7	5	8	0.63	51,955	81,408.89	0.64	51473	0.93	53.57	49.8
CC50	8	7	6	1.17	72,154	61,936.27	1.16	61759	14.41	51.38	63.6
CC55	8	10	6	1.67	102,460	61,930.55	1.65	61770	39.71	90.88	71.4
CC60	9	8	7	1.14	82,546	72,104.00	1.14	71876	12.93	84.89	82.8
CC65	10	9	8	1.13	92,803	82,326.62	1.13	81896	11.75	102.9	90
CC70	10	9	9	1.00	92,963	92,613.68	1.00	92229	0.79	149.3	102
CC75	11	9	9	1.00	93,220	92,711.74	1.01	92231	1.06	149	112.8
Average of CC	7.9	7.2	6.9	1.06	74273.2	70888.648	1.06	66658.1	8.41	82.742	67.8
DD30	5	4	6	0.67	41,426	61,040.10	0.68	41137	0.70	41.4	27.6
DD35	5	5	7	0.71	51,614	71,308.04	0.72	51545	0.13	68.2	33.6
DD40	6	5	6	0.83	51,851	61,531.68	0.84	51290	1.08	68	43.2
DD45	7	5	8	0.63	51,960	81,601.63	0.64	51238	1.39	72.86	48
DD50	8	6	7	0.86	62,131	71,761.23	0.87	62065	0.11	70.25	60
DD55	8	6	7	0.86	62,358	72,051.95	0.87	61789	0.91	121.38	69
DD60	9	7	8	0.88	72,521	82,308.08	0.88	72120	0.55	113.78	78.6
DD65	10	7	8	0.88	72,825	82,200.77	0.89	72385	0.60	120.6	90
DD70	10	8	8	1.00	83,034	82,631.56	1.00	82496	0.65	173.6	102
DD75	11	9	9	1.00	93,255	92,970.84	1.00	92748	0.54	165.45	109.8
Average of DD	7.9	6.2	7.4	0.83	64297.5	75940.588	0.84	63881.3	0.67	101.552	66.18

XX30	5	4	-	-	41,093	-	-	40923	0.41	101.4	-
XX35	5	5	-	-	51,313	-	-	51075	0.46	174.6	-
XX40	6	5	-	-	51,540	-	-	51120	0.81	166.33	-
XX45	7	7	-	-	71,719	-	-	61156	14.73	156.29	-
XX50	8	6	-	-	61,707	-	-	61438	0.44	126.88	-
XX55	8	6	-	-	61,839	-	-	61486	0.57	254.25	-
XX60	9	6	-	-	62,033	-	-	61945	0.14	299.44	-
XX65	10	6	-	-	62,531	-	-	62344	0.30	286.4	-
XX70	10	7	-	-	72,775	-	-	72251	0.72	400.1	-
XX75	11	8	-	-	82,960	-	-	82645	0.38	388.73	-
Average of XX	7.9	6	-	-	61951	-	-	60638.3	1.90	235.442	-
YY30	5	4	-	-	41,195	-	-	41023	0.42	76.2	-
YY35	5	5	-	-	51,363	-	-	51045	0.62	187.6	-
YY40	6	6	-	-	61,608	-	-	61242	0.59	161.33	-
YY45	7	6	-	-	61,806	-	-	61502	0.49	176.14	-
YY50	8	6	-	-	61,966	-	-	61706	0.42	203.88	-
YY55	8	7	-	-	72,121	-	-	61840	14.26	274.13	-
YY60	9	6	-	-	62,321	-	-	61972	0.56	276.56	-
YY65	10	7	-	-	72,464	-	-	72132	0.46	327.3	-
YY70	10	7	-	-	72,586	-	-	72211	0.52	419.6	-
YY75	11	9	-	-	92,679	-	-	82325	11.17	397.64	-
Average of YY	7.9	6.3	-	-	65010.9	-	-	62699.8	2.95	250.038	-

Conclusions

In this research, by extending the work pioneered by Bellman (1962), Held and Karp (1962) and Psaraftis (1980) on using the dynamic programing method to solve the vehicle routing problem, we embed many complex constraints of the pickup and delivery problem with transfers on a three-dimensional space-time-state network. In this network construction process, elements of time and load are explicitly added as new dimensions to the physical transportation network. To address the curse of dimensionality, we demonstrate a consistent transition from the microscopic cumulative service states to macroscopic cumulative flow count diagrams, which can be used to effectively estimate the overall dynamic system performance. We also split the large-scale primary problem into a number of small-scale sub-problems. We use a dynamic programming algorithm to solve a least-cost path problem for the local clusters derived from the original set of parcels. At the end, extensive computational results over the data set used by Ropke and Pisinger (2006) and real-world data set proposed by Cainiao Network were performed to examine the effectiveness of our developed algorithm.

References

Bellman, R. (1962) Dynamic programming treatment of the traveling salesman problem. *Journal of the Association for Computing Machinery*, 9: 61-63.

Cortés, C.E., Matamala, M., Contardo, C. (2010) The pickup and delivery problem with transfers: formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200, 711–724.

Drexl, M. (2012a) Synchronization in vehicle routing – A survey of VRPs with multiple synchronization constraints. *Transportation Science*, 297–316.

Drexl, M. (2012b) Rich vehicle routing in theory and practice. *Logistics Research*, 5, 47–63.

Drexl, M. (2013) Applications of the vehicle routing problem with trailers and transshipments. *European Journal of Operational Research*, 227, 275–283.

Hall, R.W. (1991) Queueing Methods for Services and Manufacturing, Prentice Hall, Engelwood Cliffs, New Jersey.

Held, M., Karp, R.M. (1962) A dynamic-programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1): 196-210.

Large-scale network and test data set from Cainiao's Last Mile Rush competition, Retrieved on August 18, 2017 from https://tianchi.shuju.aliyun.com/competition/information.htm?spm=5176.100067.5678.2.LRPZpR&raceId=231581.

Mues, C. and Pickl, S. (2005) Transshipment and time windows in vehicle routing. In *Proceedings of the 8th international symposium on parallel architectures. Algorithms and networks* (pp. 113–119). Piscataway, NJ: IEEE.

Newell, G.F. (1982). *Applications of Queueing Theory,* Chapman and Hall, London.

Psaraftis, H.N. (1980) A dynamic programming approach to the single-vehicle, many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14(2): 130–154.

Rais, A., Alvelos, F., Carvalho, M.S. (2014) New mixed integer programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235(3):530–539.

Ropke, S. and Pisinger, D. (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4) 455–472.

Savelsbergh, M.W.P and Sol, M. (1998) Drive: Dynamic routing of independent vehicles. *Operations Research*, 46(4): 474-490.