ICIC Express Letters
Volume **10**, Number **4**, April **2016**

ICIC International ©2016 ISSN 1881-803X
pp. 793–798

A HEURISTIC RESCUE METHOD FOR COLD CHAIN DELIVERY PROBLEM WITH AN ACCIDENT OF VEHICLE BREAKDOWN

Zheng Wang and Lei Bao

School of Software Technology
Dalian University of Technology
No. 2, Linggong Road, Ganjingzi District, Dalian 116023, P. R. China
drwz@dlut.edu.cn

Received October 2015; accepted January 2016

Abstract. *The accident of vehicle breakdown may easily break off the continuous low temperature state of a cold chain in food delivery. Once a vehicle is broken down, it is crucial to rescue cold-chain foods in it and get them back into low temperature as soon as possible. This paper presents the cold chain delivery problem with an accident of vehicle breakdown and sums up three possible situations of the breakdown of a refrigerated vehicle. To solve the problem, a heuristic method for rescuing the broken vehicle under the three possible situations is presented. And a number of experiments are conducted on benchmark instances. The experiments demonstrate the efficiency and effectiveness of our heuristic rescue method.*
Keywords: Heuristic rescue method, Cold chain delivery problem, Accident of vehicle breakdown

1. **Introduction.** It is very important to keep foods in continuous low temperature state throughout the whole cold chain delivery. Once exposed to high-temperature environments, foods may be very perishable and cause irreparable harm to the health of consumers. However, vehicle breakdown is a significant accident which not only is difficult to predict and avoid but easily breaks off the continuous low temperature state of a cold chain and leaves foods in a big risk of deterioration. Therefore, when a vehicle is broken down, it is crucial to rescue the foods, get them back into low temperature state as soon as possible in order to prevent them from deteriorating, and consequently guarantee food safety for consumers.

This paper focuses on the rescue method for cold chain delivery problem with an accident of vehicle breakdown (CDP-VB). The difficulty of solving the problem lies in that vehicle rescheduling, vehicle breakdown, and cold chain delivery should be considered simultaneously, while most of related researches focused on only one aspect of the problem.

On the vehicle rescheduling problem (VRSP), Li et al. gave a clear definition and presented models and fast algorithms to solve the problem [1]. Afterwards, Spliet et al. presented a mathematical programming formulation as well as a heuristic solution method referred to as the two-phase heuristic for VRSP [2]; Lorini et al. presented a methodology to quickly modify vehicle routes in order to account for the occurrence of new customer requests, which might imply diverting a vehicle away from its current destination [3]; Mu et al. developed two tabu search algorithms to deal with the vehicle breakdown disruption that occurred at the execution stage of an original vehicle routing plan [4].

Regarding the cold chain delivery, Hsu et al. constructed a model to obtain optimal delivery routes, loads, fleet dispatching and departure times for delivering perishable food from a distribution center [5]; Osvald and Stirn developed an algorithm for the distribution of fresh vegetables [6]; and Chen et al. proposed a nonlinear mathematical model to consider production scheduling and vehicle routing with time windows for perishable food products in the same framework [7].

Despite a number of studies on the respective aspects of vehicle rescheduling and cold chain delivery, the extant solution methodologies cannot be applied to cold chain delivery problem with the accident of vehicle breakdown. Therefore, a new method to deal with the problem is required.

The remainder of this paper is organized as follows. The problem description is presented in Section 2. Then Section 3 gives a heuristic rescue method. And experimental results of the method are reported and analyzed in Section 4. Finally, conclusions are provided in the end.

2. **Problem Description.**

The CDP-VB can be defined on a complete graph $G = (V, E)$, where $V = \{v_0, v_1, \dots, v_n\}$ is the set of customer vertices with the depot v_0 and $E = \{(v_i, v_j) : v_i, v_j \in V,\ i \neq j\}$ is the set of edges. Each customer v_i $(1 \leq i \leq n)$ is associated with a positive demand q_i, a service time sv_i, and a time window $[e_i, l_i]$ with $0 \leq e_i \leq l_i$ and can be visited only once by exactly one refrigerated vehicle. Each edge $(v_i, v_j) \in E$ $(i \neq j,\ 0 \leq i \leq n,\ 0 \leq j \leq n)$ is associated with a positive cost c_{ij}. Define K as a set of nk identical refrigerated vehicles available to deliver goods from the depot to the customers. Each vehicle has the same capacity Q and can perform only one route beginning and ending at the depot node v_0.

During the execution process of an initial schedule, a vehicle k is broken down and thus cannot complete its subsequent delivery tasks as scheduled. Therefore, vehicles currently in service or a backup vehicle from depot should be employed and rescheduled to rescue the foods in the broken vehicle and deliver them to customers. There are the following three possible situations of the breakdown accident of a refrigerated vehicle since a refrigerated vehicle mainly has two systems: motion system and cooling system.

(1) Vehicle k cannot move but its cooling system can work.
(2) Vehicle k cannot move and its cooling system cannot work either.
(3) Vehicle k can move but its cooling system cannot work.

The objective of our problem, defined in Formula (1), is to minimize total rescheduling cost including transportation cost, $f(x)+\alpha g(x)$, and penalty cost, $\beta h(x)$. In Formula (1), x indicates a vehicle rescheduling scheme which is a set of routes for all vehicles; $f(x)$, $g(x)$, and $h(x)$ represent the total travel cost of all vehicles, the number of backup vehicles rescheduled for rescue, and the total delay time for all customers respectively; and α and β stand for the fixed vehicle cost of a rescheduled backup vehicle and the penalty cost of one unit of delay time respectively. The calculations of $f(x)$ and $g(x)$ are the same as those defined in [8] and the calculation of $h(x)$ is the same as that in [9].

$$c(x) = f(x) + \alpha g(x) + \beta h(x) \tag{1}$$

3. **A Heuristic Rescue Method.**

CDP-VB is a difficult problem in that even very small instances of such a problem cannot be easily solved in very limited time by exact and even meta-heuristic approaches. As a result, a fast heuristic, not an exact approach or a meta-heuristic, is studied in this section to tackle the problem in practice.

3.1. **A rescue method for the first situation.**

In the first situation, vehicle k has to remain at its breakdown point and wait for rescue vehicle(s) to pick its foods. And it can wait for a longer time than the other two situations because of the effectiveness of its cooling system.

In our rescue method, we assign the unserved customers of vehicle k to other vehicles according to their original schedule sequence which has been fully optimized according to the order of the preferred time windows and the locations of customers. The sequential assignment of the unserved customers is essentially a greedy strategy, which has been proved in our experiments to be effective in that it can quickly generate good-quality solutions to CDP-VB. The rescue algorithm is detailed in the following two steps.

Step 1: Remove the first unserved customer from the unserved customer sequence of vehicle k, try inserting it into the routes of the other running vehicles or backup vehicles, and then take the insertion scheme with the minimal $c(x)$ as the final insertion solution.

Step 2: If the unserved customer sequence is not empty, go to Step 1; otherwise the rescue method stops.

Note that when an unserved customer is inserted to a route in Step 1, the breakdown point should be also assigned to the same route and visited before the unserved customer because the goods in the broken vehicle should be picked before delivered. Suppose vehicle k is broken down at the point BP when it travels from customer A2 to A3 on the route "Depot → A1 → A2 → A3 → A4 → A5 → Depot". At the moment of breakdown, another vehicle B is travelling from B2 to B3 on the route "Depot → B1 → B2 → B3 → B4 → B5 → Depot". If vehicle B has been assigned an unserved customer, e.g., A3, and the insertion scheme for another customer, e.g., A4, will be inserted into the same route of B because of its minimal $c(x)$, there will be multiple BP points in the route. For example, the route "Depot → B1 → B2 → BP → A3 → B3 → BP → B4 → A4 → B5 → depot" with two assigned customers A3 and A4 has two BP points. In such a situation, merging these BP points should be tried. Regarding the route in the above example, if the capacity of vehicle B is enough for picking goods of A4 at the first BP point, the vehicle can pick the goods of A3 and A4 together at the first BP point and hence the second BP point can be removed; otherwise, keep the two BP points in the route unchanged. The same merging process can be applied to any pair of BP points if there are more than two BP points in a route.

3.2. **A rescue method for the second situation.** In the second situation, vehicle k has to remain at its breakdown point and only one rescue vehicle and one-time goods transfer are preferred since the cooling system is not effective and thus the extant cold air in vehicle k will be released quickly if its refrigerator door is opened repeatedly for multiple rescue vehicles. Furthermore, a rescue operation must be completed in a limited period of time pt, which is regarded as a rigid constraint that should not be violated in this situation. This is because even though the refrigerator door of vehicle k is kept closed, the cold air in vehicle k will still be gradually released and consequently goods in the vehicle will begin to deteriorate after a period of time.

To deal with this situation, we first select from running vehicles and backup vehicles a feasible rescue vehicle, which not only has enough capacity to load the goods of the unserved customers of vehicle k but can arrive at the breakdown point of vehicle k in the period pt. And then we design rescue scheme for each feasible rescue vehicle by a similar procedure of inserting iteratively an unserved customer into a route detailed in Section 3.1. The difference only lies in that unserved customers can be inserted into only one route in each iteration of the procedure of the 2nd situation. Finally, we choose the rescue vehicle whose rescue scheme has the minimal total cost.

Note that in the procedure of selecting feasible rescue vehicles, if the capacity of a running vehicle is not enough for loading the goods of the unserved customers of vehicle k, we continue to check the time feasibility and the capacity feasibility of the running vehicle after it delivers its next customer. The checking procedure of a vehicle's feasibility continues as long as the time constraint is not violated and the capacity constraint is violated and stops when both the two constraints are satisfied, the time constraint is violated, or the last customer of the running vehicle is checked.

3.3. **A rescue method for the third situation.** In the third situation, vehicle k can go to meet its rescue vehicle at any point as long as its goods are transferred before they begin to deteriorate. For the same reason as the second situation, only one rescue vehicle and one-time goods transfer are preferred in this situation and a rescue operation must be completed in a limited period of time pt.

The rescue method for the third situation can be completed in the same three steps as that of the second situation: selection of feasible rescue vehicles, design of rescue scheme for each feasible vehicle, and determination of the best feasible rescue vehicle. However, because vehicle k can move in the third situation, it does not need to stay at its breakdown point waiting for rescue but can go to meet its rescue vehicle at any point in the period pt. This makes its rescue method different from that of the second situation in that the junction point of a rescue vehicle and the broken vehicle k is not necessarily the breakdown point of vehicle k but can be one of the following 7 cases which may be the potential circumstances to get the minimal total cost: (1) the midpoint of the breakdown point of vehicle k and the point in which a rescue vehicle locates when vehicle k is broken down; (2) the breakdown point of vehicle k; (3) the point in which a rescue vehicle locates when vehicle k is broken down; (4) the point of an unserved customer of a rescue vehicle; (5) the point of the next unserved customer of the broken vehicle k; (6) the midpoint of the breakdown point of vehicle k and the depot; and (7) the depot.

In the above cases (1) and (6), it is assumed that the two vehicles travel towards each other at the same speed; while in the other cases, one of the broken vehicles and its rescue vehicle remain where they are and the other travels towards it.

In the step of selecting feasible rescue vehicles, the feasibilities of both the time pt and vehicle capacity are checked on all possible junction points. If there is at least one junction point at which the two constraints are satisfied, the vehicle can be used as a feasible rescue vehicle. In the step of designing rescue scheme for each feasible vehicle, the total costs are calculated for all feasible insertions of unserved customers of vehicle k under all feasible junction points and the one with the minimal total cost is taken as the rescue scheme for the rescue vehicle.

4. **Computational Results.** The heuristic rescue method for CDP-VB has been coded in C# and run on a computer with an Intel® Core^{TM} i3 CPU and 2 GB RAM. And Professor Cordeau's VRP benchmark problems are chosen as the test instances, each of which has 100 customers and one depot. The instance information of customer locations and requirements, vehicle numbers and capacities can be found from the original instance files (http://neo.lcc.uma.es/vrp/wp-content/data/instances/cordeau/C-vrptw.zip) and the optimal solutions can be found from their solution files (http://neo.lcc. uma.es/vrp/wp-content/data/instances/cordeau/C-vrptw-sol.zip). In each instance, the first vehicle scheduled in the initial plan is supposed to break down once for all the three situations and thus there are totally 168 ($56 * 3 = 168$) rescheduling schemes that are generated by our heuristic for all the instances. The vehicle breakdown time is a random value that is uniformly distributed between 0 and the time that the first vehicle in the initial plan returns to depot. We set $\alpha = 100$, $\beta = 3$, and $pt = 800$ in our experiments and suppose the waiting time for a broken vehicle under the first situation is unlimited. Then the average vehicle breakdown time (VBT), the average total costs of rescheduling schemes under three situations (TC1, TC2, and TC3) calculated by our heuristic rescue method, the average total rescue cost (Avg. TC = (TC1 + TC2 + TC3)/3), the average total cost of the initial plan after vehicle breakdown accident (TC of IP), the average gap of values in the former two columns $\frac{\text{Avg. TC}-\text{TC of IP}}{\text{TC of IP}} * 100\%$ (Gap), the average running times in seconds (Avg. CPU), and the average number of vehicles used in initial plan (NV of IP) are reported in Table 1 for each instance set.

From Table 1, the following four points can be observed.

(1) Our rescue method can generate good solutions, the total costs of which are about 31.6% averagely higher than the total costs of initial plans for three vehicle breakdown

Table 1. Average results for each instance set under different situations

Ins. #	VBT	TC1	TC2	TC3	Avg. TC	TC of IP	Gap	Avg. CPU	NV of IP
C1	405.89	571.51	627.05	560.75	586.43	460.46	27.36%	0.102	10
C2	1777	464.23	487.63	387.14	446.33	288.94	54.47%	0.125	3
R1	97.67	801.77	815.84	711.97	776.53	632.50	22.77%	0.088	12.08
R2	692.64	374.34	376.48	319.97	356.93	257.06	38.85%	0.119	2.73
RC1	104.5	880.11	936.90	798.62	871.88	722.51	20.67%	0.134	11.5
RC2	562.75	550.03	550.03	491.95	530.67	422.23	25.68%	0.304	3.25
Avg.	606.74	607.00	632.32	545.07	594.80	463.95	31.63%	0.15	7.09

situations. It is always acceptable in practice to spend more cost than originally scheduled in rescuing broken vehicle as long as the rescue cost is not too higher than the initial cost.

(2) Furthermore, our method can be completed in a very short time of less than 1 second for one accident of vehicle breakdown. Therefore, our method can well handle the difficult CDP-VB and is able to be put into practice.

(3) It can be found that values in column TC2 are no less than values in TC1 and TC3. This is because when the time period pt is long enough for rescue vehicles in the second situation, its rescue costs will be the same as that in the first situation; and when the vehicle breakdown point is chosen as the junction point of the broken vehicle and its rescue vehicle, the rescue costs of the second and the third situations are also the same. Therefore, the second situation can be deemed as a special case of the other two situations.

(4) Another interesting result can be seen from Table 1 that generally the more the number of vehicles scheduled in initial plans, the smaller the gap between rescue costs and initial costs. The result can be explained that more vehicles scheduled can lead to more chances and ways to rescue the broken vehicle. This finding also reveals that the traditional idea about minimizing the vehicle number in the initial schedule is not beneficial to rescue once an accident of vehicle breakdown occurs.

5. **Conclusions.** This paper studies the cold chain delivery problem with an accident of vehicle breakdown, sums up three possible situations of the breakdown of a refrigerated vehicle, and develops a heuristic method for rescuing the broken vehicle under the three possible situations. The experiments made on well-known benchmark instances with vehicle breakdown accidents generated randomly for the possible three situations show that our heuristic rescue method can generate good solutions, the total costs of which are about 31.6% averagely higher than the total costs of initial plans, in a very short time of less than 1 second for one accident of vehicle breakdown. It is also revealed from our experiments that the traditional idea about minimizing the vehicle number in the initial schedule is not beneficial to rescue once an accident of vehicle breakdown occurs.

In the future study, the presented heuristic rescue method can be further improved by considering different kinds of deterioration of foods because different foods may differ in the time periods of deteriorating trends.

Acknowledgment. This work is partially supported by the grants from the National Natural Science Foundation of China (Nos. 71271037, 71272093, 71201014, 71201055) and the Fundamental Research Funds for the Central Universities (No. DUT15QY38). Those supports are gratefully acknowledged.

REFERENCES

[1] J. Q. Li, P. B. Mirchandani and D. Borenstein, The vehicle rescheduling problem: Model and algorithms, *Networks*, vol.50, no.3, pp.211-229, 2007.
[2] R. Spliet, A. F. Gabor and R. Dekker, The vehicle rescheduling problem, *Computers & Operations Research*, vol.43, no.1, pp.129-136, 2014.
[3] S. Lorini, J. Y. Potvin and N. Zufferey, Online vehicle routing and scheduling with dynamic travel times, *Computers & Operations Research*, vol.38, no.7, pp.1086-1090, 2011.
[4] Q. Mu, Z. Fu, J. Lysgaard and R. Eglese, Disruption management of the vehicle routing problem with vehicle breakdown, *Journal of the Operational Research Society*, vol.62, no.4, pp.742-749, 2011.
[5] C. I. Hsu, S. F. Hung and H. C. Li, Vehicle routing problem with time-windows for perishable food delivery, *Journal of Food Engineering*, vol.80, no.2, pp.465-475, 2007.
[6] A. Osvald and L. Z. Stirn, A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food, *Journal of Food Engineering*, vol.85, no.2, pp.285-295, 2008.
[7] H. K. Chen, C. F. Hsueh and M. S. Chang, Production scheduling and vehicle routing with time windows for perishable food products, *Computers & Operations Research*, vol.36, no.7, pp.2311-2319, 2009.
[8] Z. Wang, Y. Li and X. Hu, A heuristic approach and a tabu search for the heterogeneous multi-type fleet vehicle routing problem with time windows and an incompatible loading constraint, *Computers & Industrial Engineering*, 2014.
[9] E. Taillard, P. Badeau, M. Gendreau, F. Guertin and J. Y. Potvin, A tabu search heuristic for the vehicle routing problem with soft time windows, *Transportation Science*, vol.31, no.2, pp.170-186, 1997.