Journal of the Operational Research Society (2011) 62, 742–749

www.palgrave-journals.com/jors/

Disruption management of the vehicle routing problem with vehicle breakdown

Q Mu[1], Z Fu[2], J Lysgaard[3] and R Eglese[1]*

[1]Lancaster University Management School, Lancaster, UK; [2]Central South University, Changsha, PR China; and [3]Aarhus School of Business, Aarhus University, Aarhus, Denmark

This paper introduces a new class of problem, the disrupted vehicle routing problem (VRP), which deals with the disruptions that occur at the execution stage of a VRP plan. The paper then focuses on one type of such problem, in which a vehicle breaks down during the delivery and a new routing solution needs to be quickly generated to minimise the costs. Two Tabu Search algorithms are developed to solve the problem and are assessed in relation to an exact algorithm. A set of test problems has been generated and computational results from experiments using the heuristic algorithms are presented.

Journal of the Operational Research Society (2011) **62,** 742–749. doi:10.1057/jors.2010.19
Published online 21 April 2010

Keywords: vehicle breakdown; vehicle routing; heuristics

Introduction

Many procedures have been developed to produce optimal or near optimal solutions to vehicle routing and scheduling problems where demands and travel times are known and fixed. However, no matter how good a plan is, various disruptions may take place at the execution stage, which can make the plan no longer an optimal or even a feasible solution. Disruptions during the execution of a vehicle routing problem (VRP) plan may be caused by vehicle breakdowns, traffic accidents blocking one or more links, delayed departures from the depot or any service point, new orders or cancelled orders, etc. When a disruption occurs, routes should be quickly revised to minimise the negative effect it may cause to the delivery company and their customers. In practice, plans may be revised manually based on people's past experiences or common sense. Dealing with disruptions is a complicated decision-making process; therefore a decision support system with effective algorithms, which can quickly find a new routing plan when disruptions occur, is valuable.

The disrupted VRP problem is different from the classic VRP because:

1. In the original problem, vehicles depart from the depot and end at the depot. The goal is to design a set of minimum cost routes, originating and ending at the central depot. When a disruption happens, vehicles are at different locations. An optimal routing solution has to be found for vehicles that are starting from different locations and end at the depot. Therefore, the sub-routes can be open ones, not closed ones as they were in the original problem. However, a disrupted VRP is not exactly an open VRP problem as both ends of each vehicle route are fixed once disruption occurs.
2. Computing time. When a disruption occurs, it is critical to respond quickly with a new plan. This means while it is possible to spend hours or possibly days to find a high quality solution for the original problem, disruption management requires a quick response and thus a short computing time. As there is a clear trade-off between the computing time and the quality of the solution, an algorithm should be developed with a good balance between the two.
3. The original problem is solved from scratch. To solve the new problem, benefit will be taken from the solution to the original problem. First, this will make it quicker to find the new routing solution. Second, this may help minimise the deviation from the original plan. Deviation from the original plan may cause service delay or drivers' overtime work. Change of plan may also cause issues if drivers are not familiar with new routes assigned.
4. Objectives. The objective of the original problem is usually to minimise the total operations costs involved. When a disruption occurs, there may be additional costs to take into account (eg cost of delay on delivering an order) and the inconvenience to the customers and drivers (eg waiting time, deviation from the original plan).
5. Decision making. In the situation when disruptions happen, it may be desirable to generate multiple solutions for the decision maker to choose from. Sometimes violations of customer requirements or other constraints are unavoidable

*Correspondence: *R Eglese, Department of Management Science, Lancaster University Management School, Lancaster, LA1 4YX, UK. Website: http://www.lums.lancs.ac.uk/profiles/richard-eglese/*

when unexpected disruptions happen. Decisions may need to take account of issues that are not easily quantifiable. Several alternative good solutions may better support a manager's decision making. The original problem does not usually involve multiple solutions.

In this paper, we propose a formulation for one type of the disrupted VRP that involves a vehicle breakdown. Two Tabu Search algorithms are developed to solve the disrupted VRP. The algorithms focus on 1, 2 and 3 in the above list. Issues in 4 and 5 are not explored, that is we will not consider the multi-objective approach to the disrupted VRP in this paper. It should be noted that although we consider making use of the original plan as discussed in point 3 above, minimising deviation from the original plan is not an objective for the problem discussed in this paper. A set of test problems has been generated based on standard VRP benchmark problems and computational results from experiments using the heuristic algorithms are presented.

Literature review

Dealing with unexpected events after a plan has been made is an important issue in managing a business activity. Yu and Qi (2004) have given a detailed discussion about a variety of approaches to uncertainties. They divide those approaches into two stages: in-advance planning and real-time re-planning. Contingency planning, stochastic models and robust optimisation are examples of in-advance planning. Disruption management is an example of the real-time re-planning. Real-time re-planning is necessary in some cases, because many disruptions are rare and unpredictable, thus cannot be embedded in a plan in advance. Disruption management aims to enable the dynamic revision of an operational plan when disruptions occur and provide good feasible solutions to the disruption problem in real time.

Disruption management was first applied to flight scheduling (Teodorović and Guberinić, 1984), which is often referred to as aircraft recovery. The application was then extended to crew recovery (Teodorović and Stojković, 1995). The airline industry has been the area where disruption management is mostly studied because the flight disruption often involves huge cost loss. Literature reviews on disruption management in the airline industry can be found in Filar *et al* (2001), Kohl *et al* (2004), Yu and Qi (2004), and Clausen *et al* (2005). Yet disruption management studies can also be found in machine scheduling (Akturk and Gorgulu, 1999; Hall and Potts, 2004), project scheduling (Eden *et al*, 2002; Zhu *et al*, 2005), production planning (Yang *et al*, 2005) and supply chain coordination (Xia *et al*, 2002; Qi *et al*, 2004; Xiao *et al*, 2005). However, there is little research on dealing with disrupted VRPs.

A disrupted VRP can be approached as a special kind of Dynamic/Real-time Vehicle Routing Problem (DVRP), where new demand information is received as time progresses and must be dynamically incorporated into an evolving schedule. Such problems are found in many different application areas and there exist some problems that must be solved in real time, such as dial-a-ride systems, courier services, taxi cab services and emergency services. Psaraftis (1988) was among the first to study dynamic versions of the VRP. He lists 12 issues for which the DVRP differs from the conventional static problem, reviews generic features that a dynamic vehicle routing procedure should possess and recommends directions for further research in this area. Powell *et al* (1995), Psaraftis (1995), Bertsimas and Simchi-Levi (1996), Gendreau and Potvin (1998), and Ghiani *et al* (2003) have provided detailed surveys on DVRPs and related routing problems. The current literature on DVRP has mostly focused on a single source of uncertainty (ie the dynamic service requests), although some recognition has been given to other sources of uncertainty (Gendreau and Potvin, 1998). Disruption management deals with all unexpected events that are significant enough to require the original routing plan to be changed.

Li *et al* (2004, 2007a, 2007b, 2009a) have done a series of studies on the vehicle rescheduling problem (VRSP). It is based on the Single Depot Vehicle Scheduling Problem (SDVSP), which is the problem of assigning vehicles to a set of predetermined trips with fixed starting and ending times with an objective of minimising capital and operating costs. The VRSP involves producing a new schedule for the previously scheduled trips after a trip has been severely disrupted. After that, Li *et al* (2009b) introduced the Real-time Vehicle Rerouting Problems with Time Windows. The paper addresses a similar problem to the one formulated in our paper in that they both look at vehicle breakdown that disrupts a VRP plan. However, Li *et al* (2009b) deals with a VRP with Time Windows, which involves both delivery and pickup services whereas our paper deals with a Capacitated VRP where Time Windows are not involved. The rerouting problem in Li *et al* (2009b) is formulated as a set-covering problem and the authors try to minimise a weighted sum of operation, service cancellation and route disruption costs. The problem is solved by a Lagrangian relaxation based-heuristic that includes an insertion based-algorithm to obtain a feasible solution for the primal problem. Solomon (1987) benchmark problems are used as test instances. A vehicle breakdown is introduced at an early time in the schedule and the number of backup vehicles is set as 0 or 1. The algorithm is run for a certain number of iterations. Computational results show a considerable cost saving compared to the solution from the naive manual approach.

Problem description

The problem addressed in this paper is based on the Capacitated Vehicle Routing Problem (CVRP). In this case, the disruption involves a vehicle v_i that breaks down at time t after leaving the depot and before delivering to the final customer on its route. We only look at the case where one vehicle breaks down because it is unusual to have more than one vehicle breaking down on the same day. In this situation, a number of unserved customers have to be reallocated to other vehicles. Extra vehicles (EVs) may have to be used to finish serving all the customers. The objective of this disrupted VRP is (1) to minimise the number of vehicles used and (2) to minimise the total distance travelled by all the vehicles after disruption happens to finish the delivery to the unserved customers. Priority is given to the number of vehicles. Therefore, a solution with fewer vehicles will always be better than a solution with more vehicles. For those solutions with the same number of vehicles, the one with the minimum total travel distance is preferred. The following assumptions have been made:

1. One EV is available in the depot. As has already been discussed, we assume that a solution that does not require the use of an EV is always better than the one that requires an EV.
2. A vehicle cannot be diverted until it has visited its current destination. This assumption is made because it is not easy to estimate the time required for finding the new plan and the communication with drivers. Therefore, in the new routing plan, the starting point for each vehicle that has not broken down is its next visiting point in the original plan.
3. The commodity that the vehicles are delivering is a single commodity that is transferable between customers, such as gas or oil. This means any vehicle is able to serve any customer as long as it is carrying enough to satisfy the customer requirement.
4. Each vehicle departs with full capacity.
5. The cost of the distribution is proportional to the distance travelled. No extra costs are involved.
6. All the vehicles depart from the depot at the same time 0 and we assume that all the vehicles are travelling at the same speed, that is one unit of distance per unit of time.
7. There are no time window constraints for any of the orders.
8. When a vehicle breaks down, if another vehicle is serving a customer, that customer is the starting point of its new route.
9. If a vehicle breaks down when it is serving a customer, this customer will not need to be visited by another vehicle.
10. If a vehicle has already returned to the depot when another vehicle breaks down, reuse of that vehicle should be seen as using an EV.

We shall refer to this problem as defined above as the Disrupted Capacitated Vehicle Routing Problem with Vehicle Breakdown (DCVRP-B).

The algorithms

We develop two metaheuristic algorithms to solve the DCVRP-B because the problem is NP-hard, and there is a time constraint on finding a new routing plan. Both algorithms are based on Tabu Search due to its effectiveness in solving VRPs.

Algorithm I

(1) *The initial solution*

When a vehicle breaks down, an easy alternative plan is to send another vehicle from the depot to complete the route that the broken-down vehicle has not finished. The rest of the vehicles will keep following the original plan. As the original plan already gives the best possible routes, we base our initial solution on it to save computing time when finding the new solution to the disrupted problem.

Before developing an initial solution, it is necessary to check whether it is possible to complete serving all the customers without using the EV. This is done by comparing the total demand required for the unserved customers and the total load carried by the unbroken vehicles. We use different approaches to obtain the initial solution for these two different cases.

For those problems where an EV must be used, the initial solution is the same as the easy alternative plan:

1. For the disrupted route, a vehicle is sent from the depot to serve the customer which the broken-down vehicle was going to serve next when it broke down. The vehicle will continue serving the rest of the customers following the same order as was specified in the original plan.
2. For the rest of the routes, the starting point for each route is the next customer that should be served according to the original plan when disruption happens. The vehicles will continue serving the rest of the customers following the same order as was specified in the original plan.

For those problems where the service can be completed without using the EV, we have used an insertion algorithm to find the initial solution. The routes that do not involve a broken-down vehicle are formed in the same way as in the previous case. Assume EVs are not available. The unserved customers in the disrupted route will be inserted into the closest routes according to their distance to the midpoint between the starting point and the depot. If an insertion violates the capacity constraint, the customer should be inserted into the next closest route. If there is no feasible insertion route, the customer will be inserted into its closest route even though it is infeasible. The cost function is defined as:

$$\text{Cost} = \Sigma_i [c_i + p_1 \times E_i] \tag{1}$$

where i represents the routes, c_i is the cost of route i, p_1 is a penalty for violating the capacity constraints. As will be discussed later, p_1 is initially set to a value p_1' and will be changed during the search process depending on the feasibility of the neighbourhood moves that have been made. E_i is the excess of load in route i. If a solution is feasible, E_i is equal to zero for all the routes.

(2) *The Tabu Search algorithm*
We have proposed a Tabu Search algorithm, which we believe is appropriate considering the characteristics of the DCVRP-B. It involves a small number of parameters and does not have any random element. It is simple, easy to implement, stable, flexible and finds a relatively good solution in a reasonable time.

The Tabu Search proposed starts from an initial solution, feasible or infeasible, and moves at each iteration to a best neighbourhood solution, until a stopping criterion is satisfied.

The neighbourhood search is based on a relocation process that involves removing one customer from its current route and inserting it into another route. At each iteration, all the possible moves are tried for all the customers, and the move that gives the least cost will be chosen as the next move as long as it is not in the Tabu list.

The neighbourhood solution may be infeasible and is evaluated by the mechanism proposed by Gendreau *et al* (1994). Potential neighbourhood moves are evaluated using the cost function given in Equation (1) in the last section. The penalty p_1 is initially equal to p'_1. It is multiplied by 2 if during the last r_1 consecutive iterations all the solutions have been infeasible or divided by 2 if all the solutions have been feasible during the last r_1 consecutive iterations.

The best neighbourhood solution is defined as follows: if there is any feasible neighbourhood solution with lower cost compared to the current best feasible solution, the feasible solution with the least cost is the best neighbourhood solution; otherwise, choose the infeasible solution that gives the least cost.

The tabu rule can be violated if a solution is feasible and is better than the best feasible solution, or if it is infeasible and it gives lower cost than the best infeasible solution already known.

The tabu list contains the customers that have been moved and the corresponding routes from which they are removed for the last λ iterations, that is (customer, route), λ represents the tabu list length. Every time a new move is made, the tabu list is updated.

To explore more search space, we penalise those vertices that have been moved frequently. The number of times that customer j has been relocated t_j are kept in the memory. The evaluation function can be rewritten as

$$\text{Cost} = \Sigma_i[c_i + p_1 \times E_i] + p_2 \times \left(\frac{t_j}{c}\right) \qquad (2)$$

where p_2 is the penalty term, c is the total number of relocations that have been performed so far.

To intensify the search space, each route is improved by performing two-opt and relocation every r_2 iterations. The single route improvement is also performed every time a new best feasible solution is found. The single route two-opt and relocation are each repeated for r_3 iterations and for each iteration the best neighbourhood move is performed whether it improves the original solution or not. The best solution found for each individual route is then retained.

The stopping criterion is to stop when the program has run for a certain time period because in a disruption situation, a new routing plan needs to be found within a limited time. If no feasible solution has been found by that time, the program returns the easy alternative plan. For the problems where it is possible to find a feasible solution without using the EV, if no feasible solution can be found within 5/6 time of the restricted time period, the algorithm will restart from the easy plan and run for the rest of the time.

Algorithm II

Algorithm II is based on our TS heuristic presented for an open VRP described in Fu *et al* (2005, 2006). The DCVRP-B can be thought of as a mixed closed and open VRP, where if an EV is used, then its route is a closed one and the other routes are open ones (that is, one end of each route is at the depot and the other end is a customer location); otherwise all routes are open ones. In order to make Algorithm II quicker to find the new routing solution, the following modifications were performed on the previous TS heuristic (Fu *et al*, 2005):

(1) In our previous TS heuristic (Fu *et al*, 2005), the initial solution is generated either randomly or by the farthest first heuristic. To take the benefit from the solution to the original problem, in Algorithm II, the initial solution is simply the same as the easy alternative plan.
(2) The previous TS heuristic (Fu *et al*, 2005) is able to deal with a route length constraint that does not exist in the DCVRP-B. Therefore, we relax this constraint by setting it to infinity.
(3) The stopping criterion is changed to be a time constraint.
(4) Some necessary changes are made for the TS heuristic to be capable of dealing with the mixed closed and open route situation.

Furthermore, compared with Algorithm I, this heuristic uses a different neighbourhood structure and randomly selects neighbours rather than searching a neighbourhood completely.

Results and findings

The program for Algorithm I was written in C# and Algorithm II was coded in Delphi 7.0. Both algorithms have been implemented on an Intel Core 2 Duo laptop running at 2.5 GHz with 4 GB of RAM. The test problems were adapted and selected from the standard CVRP problems provided by Augerat *et al* (1995) (Set A, B and P), Christofides and Eilon (1969) (Set E), Fisher (1994) (Set F) and Christofides *et al* (1979) (Set M). All these problems are Euclidean and it is assumed that the distance between each pair of customers is equal to the travelling cost. All the distances have been rounded to their nearest integer. For the instances in set A, both customer locations and demands are random. The instances in set B and M, however, are clustered instances. In the instances of set E and P, customer locations are uniformly distributed. The instances in set F are taken from real vehicle routing applications. The best known solution for each CVRP problem is used as the original routing plan before disruption. The direction of the vehicle on each route is the same as the best schedule given. For each test problem, different combinations of disrupted vehicle and disruption time were tested in the experiments. Because the average route length of the original routing plan for each of the test problems ranges from 83 to 164, three disruption times are created accordingly, which are either early (20), middle (40) or late (60) in the time required for a typical route. As route length varies in each problem and we choose the same disruption times for all test problems, in some cases, the proposed broken-down vehicle has already served all the customers in the route when the disruption happens. In this case, there is no need to reoptimise the problem, as the original solution will still be valid for the disrupted problem and any improvement found could have been applied to the original problem. Those invalid problems have been deleted from the problem set, which leaves 230 problems in total. For each problem, a time constraint of 60 s has been applied. For Algorithm II, the time constraint is 12 s and five runs are tested for each problem. The best result out of five is presented as the result obtained from Algorithm II within the time limit of 60 s. The results obtained by Algorithm I, II and the easy alternative plan are compared with the optimal solution that is discussed in the next section.

Optimal solution

An exact algorithm has been developed to find the optimal solutions for the problems. This has been modified from the algorithm to solve open VRPs and described in Letchford *et al* (2007). Optimal solutions can be found for most of the problems. Most optimal solutions can be obtained in a very short time. However, nine out of 230 problems require over a minute and in addition there are four cases when the algorithm cannot find a feasible solution even after a long computing time (a limit of 10 min has been used) and only lower bounds can be obtained. Thus the exact algorithm approach cannot reliably find solutions to all disruption problems within the limited time required, so a heuristic approach is still needed. The results from the exact algorithm are used to measure the quality of the heuristic algorithms.

Parameter tuning

Parameters have been tuned for Algorithm I. Algorithm I involves the following six parameters:

- p_1' initial penalty value for violation of capacity constraint.
- p_2 penalty for frequently moved vertices.
- r_1 number of iterations after which p_1 will be doubled or halved without changing feasibility.
- r_2 number of iterations after which a single route improvement should be performed.
- r_3 number of times that a single route improvement should be repeated.
- λ tabu tenure.

The parameters were tested on a sample of problems representative of the whole set. The average results obtained were stable over a wide range of settings for all parameters except p_2. We observed that better results can be obtained if a larger value of p_2 is used for smaller problems. The best average results were found by using the following parameter settings $p_1' = 1$; $p_2 = 10\,000$ (when $n \leqslant 50$) or 200 (when $n > 50$); $r_1=10$; $r_2=20$; $r_3=20$; $\lambda=n/k$ (n is the number of customers; k is the number of vehicles available).

The results presented below are the results obtained by using this parameter value set.

Results

In this section, detailed results are presented for one problem as an example, and then the results are summarised. Table 1 shows the results for problem A-n34-k5. The first column shows the disrupted vehicles and disruption times we tested. (1, 20) means vehicle 1 breaks down at time 20. The next four columns give the results of the disruption problem presented in this paper using different approaches. They are the costs that are generated after the disruption occurs. To illustrate the results, we have calculated the cost caused by the disruption, which is the difference between the distance travelled if no disruption happens and the total distance travelled if disruption happens. To calculate the disruption cost, we will need the result of the original plan (cost if no disruption happens) and the cost generated before the disruption happens, which is presented in the sixth column of Table 1. The cost of the original plan for A-n34-k5 (1, 20), for example, is 778 and so the optimal disruption cost will be $164+642-778=28$. The disruption costs for A-n34-k5 are presented in Table 2. The optimal solution is first shown in the second column followed by the results obtained by using the other three approaches. Comparisons are made and the percentage deviations from the optimal solution for each of the three approaches are presented. The average values are summarised in the last row of the table.

As can be seen, Algorithm I is able to find optimal solutions for all the problems for A-n34-k5. Algorithm II is slightly worse, with only one case (4, 20) where the optimal solution cannot be found. If using an easy alternative plan, the disruption cost for A-n34-n5 is the highest: 7.35% higher than the optimal solution.

Table 3 shows, for each of the 15 test instances, the average percentage deviations from the optimal solution for the disruption costs obtained by each of the three approaches. The bottom line shows the average of all the problems. Those problems that do not require an EV are excluded from this summary, but are presented in Table 4 because the easy plan

Table 1 Cost results for A-n34-k5

A-n34-k5	*Cost after disruption*				*Cost before disruption*
	Optimal	*Easy*	*Algorithm I*	*Algorithm II*	
(1, 20):	642	642	642	642	164
(1, 40):	544	544	544	544	276
(1, 60):	468	482	468	468	371
(2, 20):	648	648	648	648	164
(2, 40):	519	519	519	519	276
(2, 60):	439	439	439	439	371
(3, 20):	646	646	646	646	164
(3, 40):	571	571	571	571	276
(3, 60):	476	476	476	476	371
(4, 20):	639	651	639	641	164
(4, 40):	549	549	549	549	276
(4, 60):	450	462	450	450	371
(5, 20):	646	646	646	646	164
(5, 40):	535	540	535	535	287
(5, 60):	447	447	447	447	371

Table 2 Disruption cost results for A-n34-k5

A-n34-k5	*Optimal*	*Easy*	*Dev (%)*	*Algorithm I*	*Dev (%)*	*Algorithm II*	*Dev (%)*
(1, 20):	28	28	0.00	28	0.00	28	0.00
(1, 40):	42	42	0.00	42	0.00	42	0.00
(1, 60):	61	75	22.95	61	0.00	61	0.00
(2, 20):	34	34	0.00	34	0.00	34	0.00
(2, 40):	17	17	0.00	17	0.00	17	0.00
(2, 60):	32	32	0.00	32	0.00	32	0.00
(3, 20):	32	32	0.00	32	0.00	32	0.00
(3, 40):	69	69	0.00	69	0.00	69	0.00
(3, 60):	69	69	0.00	69	0.00	69	0.00
(4, 20):	25	37	48.00	25	0.00	27	8.00
(4, 40):	47	47	0.00	47	0.00	47	0.00
(4, 60):	43	55	27.91	43	0.00	43	0.00
(5, 20):	32	32	0.00	32	0.00	32	0.00
(5, 40):	44	49	11.36	44	0.00	44	0.00
(5, 60):	40	40	0.00	40	0.00	40	0.00
Average			7.35		0.00		0.53

always uses the EV and thus the results are not comparable with the optimal plan as they use a different number of vehicles. Results for the cases where the optimal algorithm failed and where a different number of vehicles is used in different approaches are also excluded in this table but are presented in Table 5.

As can be observed from Table 3, Algorithm I provides the best results among the three approaches with 1.60% higher disruption cost than the optimal solution. If following the easy plan, the disruption cost will be 13.65% higher than the optimal solution on average. In fact, within a minute, Algorithm I is able to find the optimal solution in all but nine cases out of the problems presented in this table and there are 95 cases where easy plan is not the optimal plan. It can also be observed that for all of the three approaches, the deviation from the optimal solution is bigger for larger problems. It seems that for larger problems more cost can be saved by applying algorithms to solve the disruption problem than simply using easy plan, although larger problems are more difficult to solve to optimality within the time limit. Moreover, it seems that there is more chance for the easy plan to be optimal for clustered problems, which are B-n39-k5, B-n50-k7 and M-n101-k10, where less than 1% of deviation from the optimal solution can be observed.

Table 4 shows the results for those problems that do not require an EV. Both Algorithms I and II perform worse than the cases when an EV is used. This is because the new plan should be different from the easy plan and it takes a longer time to find a good solution. Again, Algorithm I performs better than Algorithm II. In the 25 problems where an optimal solution is available, Algorithm I can find 15 of them, but some high individual percentage deviations result

Table 3 Average percentage deviations from optimal disruption costs (with extra vehicle)

Instances	*Easy* (%)	*Algorithm I* (%)	*Algorithm II* (%)
A-n32-k5	0.00	0.00	0.00
A-n33-k5	4.36	0.00	0.36
A-n34-k5	7.35	0.00	0.53
A-n39-k5	4.21	0.00	0.39
B-n39-k5	0.00	0.00	0.00
B-n50-k7	0.41	0.00	0.00
E-n22-k4	0.96	0.00	0.00
E-n51-k5	8.07	0.00	3.89
E-n76-k10	15.04	0.39	3.38
E-n101-k8	40.08	0.72	4.69
F-n72-k4	9.26	0.00	4.50
M-n101-10	0.88	0.00	0.88
P-n45-k5	20.92	0.00	1.76
P-n76-k4	26.17	5.52	18.87
P-n101-k4	48.32	19.33	33.97
Average	13.65	1.60	4.72

Table 4 Results for test problems without extra vehicles

		Algorithm I (%)	*Algorithm II* (%)
B-n39-k5	(3, 40)	0.00	0.00
	(4, 40)	0.00	0.00
B-n50-k7	(1, 40)	0.00	0.00
	(3, 40)	0.00	14.75
	(4, 40)	0.00	0.00
	(4, 60)	0.00	0.00
	(6, 40)	4.40	25.27
	(7, 60)	0.00	2.82
E-n22-k4	(2, 40)	0.00	0.00
	(2, 60)	0.00	0.00
	(3, 60)	0.00	0.00
M-n101-10	(2, 20)	77.08	327.08
	(2, 40)	0.00	0.00
	(3, 40)	16.98	40.88
	(4, 40)	1.52	13.64
	(5, 20)	60.87	258.70
	(5, 40)	0.00	0.00
	(6, 20)	420.45	502.27
	(8, 20)	57.89	589.47
	(8, 40)	0.00	0.00
	(9, 40)	6.90	18.97
	(10, 20)	117.50	380.00
P-n45-k5	(2, 60)	0.00	0.00
	(3, 60)	0.00	0.00
Average		31.82	90.58

in the average deviation from the optimal solution being over 30%.

Table 5 shows the five cases that cannot be included in the above tables either because no optimal solution has been found (problems from E-n101-k8 and M-n101-10) or because different numbers of vehicles are used (E-n76-k10 (1, 20)). For each of the problems and algorithms, the value of disruption cost is presented as well as whether the EV is used in the solution.

Conclusions

We have described the DCVRP-B. Two heuristic algorithms based on Tabu Search have been presented. One is newly proposed for the problem (Algorithm I). The other is based on previous work using the open VRP formulation (Algorithm II). The algorithms take advantage of the original plan and a new routing plan can be found within a limited time. The newly proposed Tabu Search algorithm (Algorithm I) involves a small number of parameters and does not have a random element. The algorithm is simple, flexible and always gives a feasible solution. This algorithm outperforms Algorithm II and can also save a considerable amount of disruption cost compared to using an easy alternative plan. An exact algorithm has also been developed and it is able to find the optimal solution quickly in most of the cases, but in some cases fails to find a feasible solution within the required time limit. Algorithm I is able to find the optimal solution in all but 19 cases of the 230 problems tested. Although it is relatively harder for our algorithms to find optimal solutions for larger problems because of the time limit, more disruption cost savings can actually be obtained compared with using an easy plan.

Acknowledgements—This research was partly supported by the National Natural Science Foundation of China (NSFC, 70671108).

References

Akturk MS and Gorgulu E (1999). Match-up scheduling under a machine breakdown. *Eur J Opl Res* **112**: 81–97.

Augerat P, Belenguer JM, Benavent E, Corberan A, Naddef D and Rinaldi G (1995). *Computational results with a branch and cut code for the capacitated vehicle routing problem*. Technical Report RR 949-M. Universite Joseph Fourier, Grenoble, France.

Bertsimas DJ and Simchi-Levi D (1996). A new generation of vehicle routing research: Robust algorithms addressing uncertainty. *Opns Res* **44**: 286–304.

Table 5 Results for problems that are not included in Tables 3 and 4

		Optimal	*Use EV*	*Easy*	*Use EV*	*AlgI*	*Use EV*	*AlgII*	*Use EV*
E-n76-k10	(1, 20):	66	No	7	Yes	7	Yes	66	No
E-n101-k8	(1, 20):	—	—	16	Yes	16	Yes	15	Yes
	(6, 20):	—	—	19	Yes	19	Yes	19	Yes
	(7, 20):	—	—	17	Yes	17	Yes	17	Yes
M-n101-10	(1, 20):	—	—	37	Yes	172	No	37	Yes

Christofides N and Eilon S (1969). An algorithm for the vehicle dispatching problem. *Opl Res Quart* **20**: 309–318.

Christofides N, Mingozzi A and Toth P (1979). The vehicle routing problem. In: Christofides N, Mingozzi A, Toth P and Sandi L (eds). *Combinatorial Optimization*. Wiley: Chichester, UK, pp 315–338.

Clausen J, Larsen A and Larsen J (2005). *Disruption management in the airline industry—concepts, models and methods*. Technical Report. Informatics and Mathematical Modelling, Technical University of Denmark. DTU.

Eden C, Williams T, Ackermann F and Howick S (2002). On the nature of disruption and delay in major projects. *J Opl Res Soc* **51**: 291–300.

Filar JA, Manyem P and White K (2001). How airlines and airports recover from schedule perturbations: A survey. *Ann Opns Res* **108**: 315–333.

Fisher ML (1994). Optimal solution of vehicle routing problems using minimum K-trees. *Opns Res* **42**: 626–642.

Fu Z, Eglese R and Li LYO (2005). A new tabu search heuristic for the open vehicle routing problem. *J Opl Res Soc* **56**: 267–274.

Fu Z, Eglese R and Li LYO (2006). Corrigendum: A new tabu search heuristic for the open vehicle routing problem. *J Opl Res Soc* **57**: 1018.

Gendreau M, Hertz A and Laporte G (1994). A tabu search heuristic for the vehicle routing problem. *Mngt Sci* **40**: 1276–1290.

Gendreau M and Potvin JY (1998). Dynamic vehicle routing and dispatching. In: Crainic T and Laporte G (eds). *Fleet Management and Logistics 1998*. Kluwer: Boston, pp 115–126.

Ghiani G, Guerriero F, Laporte G and Musmanno R (2003). Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *Eur J Opl Res* **151**: 1–11.

Hall NG and Potts C (2004). Rescheduling for new orders. *Opns Res* **52**: 440–453.

Kohl K, Larsen A, Larsen J, Ross A and Tiourine S (2004). *Airline disruption management—perspectives, experiences and outlook*. Carmen Research and Technology Report. CRTR-0407, September 2004.

Letchford AN, Lysgaard J and Eglese RW (2007). A branch-and-cut algorithm for the capacitated open vehicle routing problem. *J Opl Res Soc* **58**: 1642–1651.

Li JQ, Mirchandani PB and Borenstein D (2004). Parallel auction algorithm for bus rescheduling. *Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport*, San Diego, California, USA.

Li JQ, Borenstein D and Mirchandani PB (2007a). A decision support system for the single-depot vehicle rescheduling problem. *Comput Opns Res* **34**: 1008–1032.

Li JQ, Mirchandani PB and Borenstein D (2007b). The vehicle rescheduling problem: Model and algorithms. *Networks* **50**: 211–229.

Li JQ, Mirchandani PB and Borenstein D (2009a). A Lagrangian heuristic for the real-time vehicle rescheduling problem. *Transport Res E-Log* **45**: 419–433.

Li JQ, Mirchandani PB and Borenstein D (2009b). Real-time vehicle rerouting problems with time windows. *Eur J Opl Res* **194**: 711–727.

Powell WB, Jaillet P and Odoni A (1995). Stochastic and dynamic networks and routing. In: Ball MO, Magnanti TL, Monma CL and Nemhauser GL (eds). *Handbooks in Operations Research and Management Science 1995*. Network Routing Elsevier: Amsterdam, pp 141–296.

Psaraftis HN (1988). *Vehicle Routing: Methods and Studies, Chapter Dynamic Vehicle Routing Problems*. Elsevier Science Publishers B.V: North Holland.

Psaraftis HN (1995). Dynamic vehicle routing: Status and prospects. *Ann Opns Res* **61**: 143–164.

Qi X, Bard JF and Yu G (2004). Supply chain coordination with demand disruptions. *Omega* **32**: 301–312.

Solomon MM (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Opns Res* **35**: 254–265.

Teodorović D and Guberinić S (1984). Optimal dispatching strategy on an airline network after a schedule perturbation. *Eur J Opl Res* **15**: 178–182.

Teodorović D and Stojković G (1995). Model to reduce airline schedule disturbances. *J Trans Eng* **121**: 324–331.

Xia YS, Qi XT and Yu G (2002). *Real-time production and inventory disruption management under the continuous rate economic production quantity model* Working paper. Red McCombs School of Business: The University of Texas at Austin.

Xiao T, Yu G, Sheng Z and Xia Y (2005). Coordination of a supply chain with one-manufacturer and two-retailers under demand promotion and disruption management decisions. *Ann Opns Res* **135**: 87–109.

Yang J, Qi X and Yu G (2005). Disruption management in production planning. *Nav Res Log* **52**: 420–442.

Yu G and Qi X (2004). *Disruption Management: Framework, Models and Applications*. World Scientific Publishing Co. Pte Ltd: Singapore.

Zhu G, Bard JF and Yu G (2005). Disruption management for resource-constrained project scheduling. *J Opl Res Soc* **56**: 365–381.

Received November 2008;
accepted December 2009 after one revision.