Lancaster University Management School
Working Paper
2011/013

Disrupted Capacitated Vehicle Routing Problem with Order Release Delay (DCVRP-ORD)

Qianxin Mu and Richard William Eglese

The Department of Management Science
Lancaster University Management School
Lancaster LA1 4YX
UK

The LUMS Working Papers series can be accessed at http://www.lums.lancs.ac.uk/publications/
LUMS home page: http://www.lums.lancs.ac.uk/

Disrupted Capacitated Vehicle Routing Problem with Order Release Delay (DCVRP-ORD)

Qianxin Mu[a] and Richard W Eglese[a,]*

[a] *Department of Management Science, Lancaster University Management School, Lancaster, LA1 4YX,UK*

** Corresponding Author.*

Phone: +44 (0) 1524 593869

Fax: + 44 (0)1524 844885

Email: r.eglese@lancaster.ac.uk

Abstract

With the popularity of the just-in-time system, more and more companies are operating with little or no inventories, which make them highly vulnerable to delays on supply. This paper discusses a situation when the supply of the commodity does not arrive at the depot on time, so that not enough of the commodity is available to be loaded on all vehicles at the start of the delivery period. New routing plans need to be developed in such a case to reduce the impact the delay of supply may have on the distribution company. The resulting vehicle routing problem is different from other types of vehicle routing problems as it involves waiting and multiple trips. Two approaches have been developed to solve the order release delay problem, both of which involve a Tabu Search algorithm. Computational results show the proposed approaches can largely reduce the disruption costs that are caused by the delayed supply and they are especially effective when the length of delay is long.

Key words: Vehicle routing, disruption management, heuristics, multi-objective optimisation

Introduction

The classical vehicle routing problem (VRP) is concerned with designing a set of vehicle routes so that a number of customers can be visited at the minimum cost. Each route starts and ends at the depot and each vehicle is involved with only one route. The load that each vehicle carries must not exceed the vehicle's capacity and sometimes there is a preset limit on the total length of each route. Each customer must be visited exactly once. The objective is to minimise the travel distance and the number of vehicles used. In practice, the vehicle routing plans are sometimes disrupted by unexpected events such as vehicle breakdowns, traffic congestions, delayed departure from the depot or any service point, new orders or cancelled orders, etc. Disruptions often make the original routing plan no longer optimal or even feasible. Therefore, it is often necessary to revise the original routing plan to minimise the negative effect it may cause to the delivery company and their customers. The complexity of the resulting problems increases because there are additional issues to be taken into account. A number of potentially conflicting objectives need to be considered and decision makers need to be actively involved in managing the situation and deciding the best course of action. Disruption management in practice involves making judgements about relative priorities as well as replanning and rescheduling.

A number of studies have been conducted during the past few years on dealing with disruptions that occur during the execution stage of a vehicle routing plan. Zhang and Tang (2007) presented a rescheduling model of a vehicle routing problem with time windows when a vehicle disruption, such as vehicle failure or a traffic incident, occurs at a particular time and lasts for a predefined period of time. The objective is to find a new schedule that minimizes the weighted sum of total distance and deviations from the time windows. A hybrid algorithm which combines ant colony optimization (ACO) with scatter search was proposed to solve the problem. Computational results show that the new schedule produced by the hybrid algorithm has saved a considerable amount of cost compared with following the original schedule. Li et al. (2009) introduced the Real-time Vehicle Rerouting Problems with Time Windows which deals with vehicle breakdown that disrupts a VRP plan. The rerouting problem 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 and computational results show a considerable cost saving compared to the solution from the naive manual approach. Mu et al. (2010) also look at the situation when a vehicle breaks down during the delivery and a new routing solution needs to be quickly generated to minimise the costs. The problem is based on the capacitated VRP (CVRP) and the objective is to minimise the number of vehicles used and the total travel distance. The problem is defined as the Disrupted Capacitated Vehicle Routing Problem with Vehicle Breakdown (DCVRP-B). Two Tabu Search algorithms are developed to solve the problem.

One is newly proposed for the problem and the other is based on previous work using the open VRP formulation. They are assessed in relation to an exact algorithm. The algorithms are tested on a number of test instances that are designed based on standard CVRP test problems. The proposed algorithms are run under a time constraint of 1 minute to reflect the need to respond quickly with a working alternative plan when disruption occurs. The newly proposed Tabu Search algorithm returns the best results and can also save a considerable amount of disruption cost compared to using an easy alternative plan.

This paper discusses a situation when the supply of the commodity does not arrive at the distribution depot on time, so that not enough of the commodity is available to load all vehicles for their delivery schedule. New vehicle routing plans need to be developed quickly in such a case to reduce the impact the delay of supply may have on the distribution company. The newly proposed Disrupted Capacitated Vehicle Routing Problem with Order Release Delay (DCVRP-ORD) is different from the other disruption problems discussed before because the disruption happens to the commodity supply and it happens before the vehicles leave the depot. In addition, the new plan involves waiting and multiple trips. Characteristics of this order release delay problem are examined in this paper. Different approaches are applied to find the optimal solution to the problem and the results are analysed.

Problem Description

DCVRP-ORD is a problem which finds an alternative routing plan when not all the goods are ready to be delivered at the start of the delivery period but delayed items will be available later during the period. Therefore, either certain vehicles have to wait at the depot until the goods become available again, or some of them have to depart with only part of the orders and come back for the delayed items later. The delay could cause both delay of the services to customers and driver's overtime working, neither of which is desirable. To minimise the impact the disruption has caused, rerouting is necessary. The new routing plan may involve multiple trips as waiting until the items become available does not always (or often does not) produce the optimal plan. Moreover, as some orders only become available after a certain time, certain drivers may have to wait at the depot till the delayed orders arrive. Even though the driver may have already completed a trip, the delayed orders may still not have arrived by the time the driver is back at the depot. Therefore, waiting time has to be taken into account when calculating the driver's time cost.

DCVRP-ORD is based on CVRP, which means the original VRP is formulated as a CVRP. By this, we assume that the original problem is only concerned with vehicle capacity, travel distances and number of vehicles used. In DCVRP-ORD, to minimise the negative effect the supply delay has caused, some additional objectives need to be taken into account apart from minimising total travel distance. They are:

1. Minimise delay to the services. Although the original problem does not involve delivery time windows, a deviation from the original delivery time may cause problems for the customers if they had been expecting a delivery to arrive by a certain time. Early delivery may also be a problem if customers need to make special arrangements to receive the delivery. We assume only a service which is delivered later than originally planned will cause disruption cost, early deliveries will not. We also assume that delay cost is proportional to the length of the delay.
2. Minimise overtime for drivers. Drivers may have already been assigned to another job after the current one. A delay on the current job may affect the following jobs. This can cause extra costs for employing drivers or disruption costs on the following jobs if they are for the same company. This is different from minimising total travel distance, instead the objective is to limit the time to complete each route. We assume all the drivers are paid for the fixed period that was planned in the original routes. Any additional time will be paid. The wage rate for overtime is proportional to the extra time that the driver works compared to the fixed period. Drivers will also get paid for waiting at the same rate.
3. Minimise the deviation from the original plan. Minimising drivers' overtime and delay to services can both be considered as minimising the deviation from the original plan. Drivers' unfamiliarity with the new routes due to the change of plan could also be included but will not be counted in the model studied. This objective will not be explicitly included as it is already covered by objectives 1 and 2.

This paper presents a weighted sum approach to the multi-objective DCVRP-ORD problem. This approach is easy to implement and computationally efficient. It can also provide an insight into the characteristics of the DCVRP-ORD problem by adjusting weights according to the decision maker's preferences. The aggregated objective function is shown below:

$$Min\sum_i D_i C_1 + \sum_i \max(F_i, (L_i + W_i))C_2 + \sum_j \max(0, (T_j^A - T_j^P))C_3 \qquad \text{Eq. (1)}$$

C_1 - variable vehicle cost per distance unit travelled

C_2 - labour cost per time unit

C_3 - delay cost per time unit

D_i - travel distance for vehicle i

F_i - fixed time for driver i

L_i - actual time used for driver i

W_i - total waiting time at the depot until delayed supply is available for vehicle i

T_j^A - actual service time for customer j

T_j^P - planned service time for customer j

Vehicle capacity and number of vehicles are the same as that in the original plan which means the delivery can still be finished in one trip. There is no route length limit for each vehicle, although if a new route becomes longer than originally planned, the driver has to be paid for the extra time used. This problem studies the situation when only one delay has happened and all the delayed orders will arrive at the same time. We also assume that the amount of the delayed orders and the length of delay are known and will be taken into account when planning the new routing schedule. A new constraint that has to be applied to this order release delay problem is the total amount that can be carried in the trips that depart before the delayed orders have arrived. The routing plan in the order release delay problem can be divided into two stages. The first stage involves the trips that depart before the delayed items arrived. The second stage involves the trips that deliver the delayed items. The total demand of all the customers that are served in the first stage has to be less than or equal to the total available supply at the start

of the day. It should be noted that to minimise the total costs, there is no need for any vehicle to do more than one trip in the first stage or in the second stage.
The following assumptions have also been made when modeling the problem:

1. The commodity that the vehicles are delivering is a single commodity which is transferable between customers and there is no requirement that certain items have to be delivered by specific vehicles.
2. A service can be immediately delivered when a vehicle arrives at a customer, i.e. the service time is 0.
3. Split delivery is not allowed which means once a customer is served it has to receive all the orders at one time.
4. A distance unit is assumed to be 1 km. A time unit is assumed to be 1 minute. The vehicle speed is assumed to be 1 km/minute all the time which corresponds to an average speed of 60 km/h.

Approaches

This section describes an easy alternative plan, which can be easily obtained manually, and two other approaches each of which involves a different initial solution and improved by a Tabu Search algorithm.

Easy plan

When supply is delayed, a common response is to still follow the original routing plan, but some vehicles will have to wait at the depot until the delayed supply becomes available. Assume that a vehicle will not depart with partially loaded goods. Therefore, the easy plan is simply to delay a necessary number of vehicles in the departure and wait until all the delayed orders arrive. As the routes follow the original plan, total travel distance $\sum_i D_i C_1$ will keep the same as before. For those delayed vehicles, the driver's working time is $L_i + W_i$. As routes follow the original plan, L_i is fixed and driver's time actually depends on the waiting time W_i and how many vehicles have been delayed. If a vehicle departs at the planned time, the driver's time is the same as the route length which stayed the same as planned. The services for all the customers on the delayed routes will definitely be delayed. Therefore, to minimise the original objective function is equivalent to minimising the following:

$$Min\, DVC_2 + DSC_3 \qquad \text{Eq. (2)}$$

where D is the length of the delay, V is the number of delayed vehicles, and S is the number of delayed customer services. Therefore, the easy plan tries to find a suitable set of vehicles to be delayed so that $VC_2 + SC_3$ is minimised.

Approach 1

The easy plan is not always a good idea because not all available orders are delivered at the earliest possible time. This will cause increased service delay to customers when the delay is long. To improve the result, our first approach will apply a Tabu Search algorithm to this easy plan. Improvement can be obtained by moving customers into different routes or by delivering to them at different times.

Approach 2

Another approach is to deliver all the current available orders at the start of the day to reduce the delay to customers. There are a number of ways of doing this. A common response to an order release delay situation is still to follow the original plan but operate so that some vehicles leave with only part of the load required for the route. One can also send all the vehicles out for delivery at the start of the day. Who will carry full loads and who will carry partial loads are decided by certain rules. Approach 2 is described as follows:

1. The delivery is divided into two stages. The available supply will be delivered at the first stage. The delayed supply will be delivered at the second stage. The first stage will deliver as many available orders as possible.
2. Choose the customers which are the closest to the depot to be served in the second stage. Reorder the customers according to their distances to the depot in ascending order. Start from the top of the list, take out as many customers as possible from the list as long as the total demand of the remaining customers is smaller than or equal to the supply that is available for the first stage. Put the customers that are taken out on the second stage list. Ideally, this will reduce service delay because it takes a shorter time for the delayed items to arrive at the customers’ locations. This could also reduce the distance travelled at the second stage as customers are closer to the depot.

3. The initial routes in the first stage are the same as the original plan but with those customers on the second stage list deleted. The initial solution for the second stage is to use a least cost insertion method to insert the customers on the list into the most promising routes.

This initial solution is followed by the same Tabu Search algorithm as Approach 1. Although the Tabu Search algorithm is the same, the quality of initial solution can make a big difference to the final results for disruption problems due to the limited time for finding the new routing plan.

Tabu Search

The Tabu Search algorithm applied is similar to that used for DCVRP-B proposed in Mu et al. (2010), though some changes have been made.

As computing time is limited for the disruption problem, the neighbourhood structure is kept simple. Vertex relocation is still the only neighbourhood operator used which involves relocating a vertex into another route. The relocation route could be a route in the first stage or in the second stage. It can also be the route served by the same vehicle, i.e. a vertex can be removed from the first stage of a vehicle into the second stage of the same vehicle. Because not all the vehicles are assigned multiple trips in the initial solution, those that are assigned only one trip will be artificially added another trip 0-0 so that the possibilities are open for customers to be inserted and to have multiple trips. For example, suppose a vehicle has to wait at the depot until the delayed orders arrive before it can do the route 0-1-2-3-0. To do the Tabu Search, the new initial routing plan for the vehicle will become:

0-0	*0-1-2-3-0*
Stage 1	Stage 2

Suppose a vehicle departs at the start of the day and all the customer demand can be met by this delivery. A second stage will be added as follows:

0-1-2-3-0	*0-0*
Stage 1	Stage 2

Therefore, in Tabu Search, the number of routes are 2K (K is the number of vehicles) because every vehicle has two routes, each in different stages. The starting time for all the trips in stage 1 is 0, and the starting time for the trips in stage 2 varies. If the first trip of a vehicle finishes (at time t) before the delayed items become available (at time *t'*), the starting time for the second trip of the vehicle will be the time when the delayed items arrive, i.e. *t'*. There will be a waiting time of (*t'*- *t*) for the vehicle. If by the time a vehicle finishes its first trip and comes back to the depot the delayed items have already arrived, i.e. *t'*< *t*, it can start the second trip straight away and the starting time for the second stage is *t*.

At each iteration, all the possible moves are tried for all the customers, and the move that gives the least cost according to the evaluation function will be chosen as the next move as long as it is not in the tabu list The tabu list in the proposed algorithm contains the customers that have been moved and the corresponding routes from which they are removed, i.e. (customer, route). Only those moves that are performed in the last θ iterations are recorded and every time a new move is made, the tabu list is updated by adding the attributes of the new move and deleting the attributes of the oldest move. θ represents the length of the tabu list, also called tabu tenure. The evaluation function is defined as follows:

$$Eq.(1) + p_1 E_1 + p_2 E_2 + p_3 ({t_j}/{C}) \qquad \text{Eq.(3)}$$

As mentioned before, two constraints should be met: one is that the total load carried by a vehicle must not exceed the vehicle capacity; the other is that the amount delivered at the first stage should be less than or equal to the amount available at the beginning. Violation of the constraints is allowed during the search process but should be penalised in the objective function with self adjusting penalty parameters p_1 (for capacity constraint) and p_2 (for supply constraint). The penalty parameters are initially set to p_1' and p_2' respectively. They are multiplied by 2 if during the last r_1 and r_2 consecutive iterations all the solutions have been infeasible, or divided by 2 if all the solutions have been feasible during the last r_1 and r_2 consecutive iterations. E_1 represents the total load that exceeds vehicle's capacity in the proposed solution and E_2 represents the shortage of supply for the first stage in the proposed solution.

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

The vertices that have been moved frequently are also penalised. The number of times that customer j has been relocated t_j are kept in the memory. The final term in the evaluation function Eq.(3) represents the long term memory cost of the proposed move. p_3 is the penalty parameter. C is the total number of relocations that have been performed so far.

To intensify the search in promising regions, periodic route improvements are performed. Single route improvement is performed every time a new feasible best solution is found. It is also performed for the current best infeasible solution if for the past r_3 iterations no better feasible solution has been discovered. The single route improvement procedure involves 2-Opt and vertex relocation applied sequentially to each route. The two methods are each repeated for r_4 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.

RESULTS AND FINDINGS

This section presents the experimental testing results of the approaches described in the previous section. 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). For most of these problems, optimal solutions have been found using exact algorithms and thus can be used as the solution of the original problem before disruption occurs. 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 each instance, a set of 6 problems will be tested which are defined by different choices of delayed amount (small, large, extra large) and different choices of length of delay (short, long). The choice of the amount of delayed supply depends on the maximum single route demand as well as the total demand of all the routes. Let X = the maximum route demand, TD = total demand, the three classes of delayed amount can be represented as follows:

Small: only one vehicle is delayed. Delayed amount = $0.5X$

Large: more than one vehicle delayed. Delayed amount = $1.5X$

Extra Large (XL): about half of the vehicles are delayed. Delayed amount = 0.5*TD*

Length of delay could be short or long, which depends on the average route lengths in the original plan. Let Y = average route length in the original plan, we use the following to represent different classes of length of delay:

Short delay: Delayed time = $0.5Y$

Long delay: Delayed time = $1.5Y$

Table 1 shows the maximum route demand for each test problem, as well as the total demand and the average route length.

Table 1 Route demand and length information

Instances	Maximum route demand	Total demand	Average Route Length
A-n32-k5	98	410	157
A-n33-k5	98	446	132
A-n34-k5	96	460	156
A-n39-k5	100	475	164
B-n39-k5	100	440	110
B-n50-k7	100	609	106
E-n22-k4	5900	22500	94
E-n51-k5	160	777	104
E-n76-k10	140	1368	83
E-n101-k8	199	692	102
F-n72-k4	29978	114840	59
M-n101-k10	200	1810	82
P-n45-k5	149	692	102
P-n76-k4	350	1364	148
P-n101-k4	392	1458	170

Table 2 shows the choices of the delayed amount and the delayed time for each test problem, rounded to the nearest integer.

Table 2 Choices of delayed amount and length of delay

	Delayed amount			Length of delay	
Instances	small	large	xl	short	long
A-n32-k5	49	147	205	78	235
A-n33-k5	49	147	223	66	198
A-n34-k5	48	144	230	78	233
A-n39-k5	50	150	238	82	247
B-n39-k5	50	150	220	55	165
B-n50-k7	50	150	305	53	159
E-n22-k4	2950	8850	11250	47	141
E-n51-k5	80	240	389	52	156
E-n76-k10	70	210	684	42	125
E-n101-k8	100	299	346	51	153
F-n72-k4	14989	44967	57420	30	89
M-n101-k10	100	300	905	41	123
P-n45-k5	75	224	346	51	153
P-n76-k4	175	525	682	74	222
P-n101-k4	196	588	729	85	255

For each test instance, 6 different combinations of delayed amount and length of delay have been tested:

SS - Small amount, short delay

SL - Small amount, long delay

LS - Large amount, short delay

LL - Large amount, long delay

XLS - Extra large amount, short delay

XLL - Extra large amount, long delay

Therefore, there are 90 problems to be tested in total. 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 problem, a time constraint of 60 seconds has been applied. The program has been written in C# and implemented on an Intel Core 2 Duo laptop running at 2.5GHZ with 4GB of RAM.

Parameter tuning

Although the algorithm uses standard approaches and many components and mechanisms of the algorithm have been applied by many authors to solve different problems, their choices of values of parameters are not necessarily suitable for the DCVRP-ORD presented in this paper. As the performance of a heuristic may vary quite significantly when parameter values are modified, it is wise to find the appropriate parameter values for the algorithm that are applied for the order release delay problem. The purpose of the tuning procedure is not finding the best solutions to the test problems used, but to define the suitable parameter settings, whether or not dependent on the problem characteristics, that could provide a high quality solution for any kind of DCVRP-ORD. The following parameters for the proposed Tabu Search algorithm have been tested with different values:

p_1' - initial penalty value for violation of capacity constraint.

p_2' - initial penalty value for violation of supply constraint.

p_3 - 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 p_2 will be doubled or halved without changing feasibility

r_3 - number of iterations after which the single route optimisation procedure should be performed

r_4 - number of iterations that a single route optimisation algorithm should be repeated

θ - Tabu tenure

Parameter values will be tested on the whole set of 90 problems. Weight values used for parameter tuning are: $C_1 = 0.3, C_2 = 0.1, C_3 = 0.5$. These are estimated values of the real life costs related to each objective and next section explains the reason why these values are chosen. Parameter tuning is performed for Approach 1 only and then used for Approach 2. Results are compared to the easy plan and percentage deviations are calculated. The parameter values that provide the best results are:

$p_1' = 1, p_2' = 1, p_3 = 200, r_1 = 10, r_2 = 10, r_3 = n/k, r_4 = 1$ (n is the number of customers; k is the number of vehicles available).

The results presented in the next section are obtained by using this parameter set for the Tabu Search algorithm.

Results

This section summarises the results for different approaches to solve the test problems using the parameter values that have been chosen in the last section. The algorithm is executed on the test problems using 7 different configurations of weights. In practice, these weights could be set by the decision makers according to their preferences for each objective. Here, the weights for each objective are chosen inside the estimated cost range for each unit of the objective. The travel cost is estimated as 30p~40p per kilometre. The average pay to drivers is estimated as within the range of 10p/min and 20p/min. The cost to the delay of service is hard to quantify. However, it is reasonable to assume that the cost of delay depends on the length of delay and a cost penalty of at least £30 per hour i.e. 50p per minute represents the right order of magnitude for some situations. The following values of weights have been considered:

Table 4 Combination of weights tested

Test	C_1	C_2	C_3
1	1	0	0
2	0	1	0
3	0	0	1
4	0.3	0.1	0.5
5	0.4	0.1	0.5
6	0.3	0.2	0.5
7	0.3	0.1	2

The first three tests optimise the three objective functions separately. In these cases, only one objective will be optimised without considering the other objectives. These three tests define the extreme values of the objective solution space. However, the result for test 1 will not be presented in this paper because if the objective is to minimise the total distance travelled, the easy plan always produces the optimal solution as the routing plan is exactly the same as the original plan. The other four tests apply different combinations of weight values according to the estimation of the actual cost caused. Test 4 gives a base value set and tests 5, 6 and 7 contain the combination of weights with only one weight being different from test 4. Although these tests do not generate all the Pareto-optimal solutions, the comparison of results given by different weight combinations will help to provide a deeper understanding of the DCVRP-ORD problem. In addition to the total cost generated by the aggregated objective function, the value of each single objective function that is produced by optimising the weighted sum of all the objectives has also been recorded.

An example

Take problem A-n32-k5 as an example. Figure 5 shows the results for the problem under different choices of delayed amount and length of delay when $C_1 = 0.3, C_2 = 0.1, C_3 = 0.5$.

Table 5 Results for A-n32-k5 at $C_1 = 0.3, C_2 = 0.1, C_3 = 0.5$

		Total Cost			Distance			Driver time			Delayed service		
Instance	**Delay**	**Easy**	**A1**	**A2**	**Easy**	**A1**	**A2**	**Easy**	**A1**	**A2**	**Easy**	**A1**	**A2**
A-n32-k5	SS	477.4	445.3	397.7	784	800	876	862	878	969	312	235	76
	SL	807.1	696.5	599.2	784	800	949	1019	1035	1105	940	706	408
	LS	758.2	632.2	534.8	784	886	944	940	974	1116	858	538	280
	LL	1653.1	1367.4	1225.9	784	1030	934	1254	1439	1317	2585	1829	1628
	XLS	844	746.8	738.8	784	836	924	1018	1070	1106	1014	778	702
	XLL	1911.6	1558.4	1564.6	784	1136	1050	1489	1706	1616	3055	2094	2176

The second column in Table 5 shows the different combinations of delayed amount and length of delay tested. The third, fourth and fifth columns contain the total costs generated by optimising the weighted sum of all the objectives by using the easy plan, Approach 1 and Approach 2 respectively. The next group of 3 columns shows, for each approach, the distance travelled in the optimal solution if the aggregated objective function is to be optimised. The next group of 3 columns shows the total driver time spent and the last group of columns is the total time of delay to the customers.

For (SS-49/78), i.e. small delay amount 49 and a short delay of 78 minutes, the easy plan is:

Table 6 Result for A-n32-k5 (SS) Easy plan

A-n32-k5 (SS) Easy Plan												
Route #1:	0	21	31	19	17	13	7	26	0			
Event Times:	0	64	73	78	80	104	118	134	155			
Route #2:	0	12	1	16	30	0						
Event Times:	78	107	115	126	135	151						
Route #3:	0	27	24	0								
Event Times:	0	26	34	59								
Route #4:	0	29	18	8	9	22	15	10	25	5	20	0
Event Times:	0	62	100	109	128	132	155	172	188	210	231	267
Route #5:	0	14	28	11	4	23	3	2	6	0		
Event Times:	0	27	85	104	113	142	149	152	178	230		

For each vehicle, the table shows the time that the vehicle reaches each customer. If a vehicle's starting time is not 0, it means the vehicle is delayed. For example, in this problem, Vehicle 2 is delayed for 78 minutes. In this case, the easy plan is the same as the original plan, except that Vehicle 2 has to wait at the depot for 78 minutes until it can start the delivery and all the customers on that route are delayed for 78 minutes. Figure 1 shows the routes of the original plan in a graph.

Figure 1 Graph for A-n32-k5 original plan

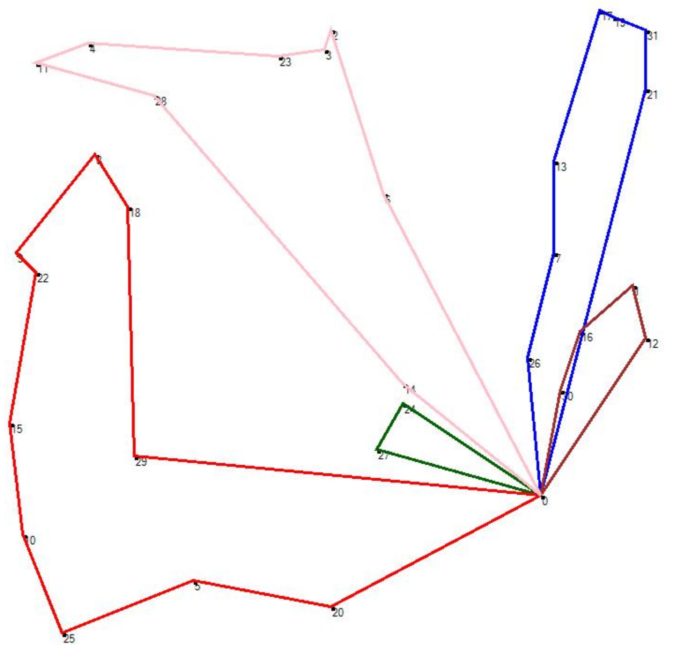


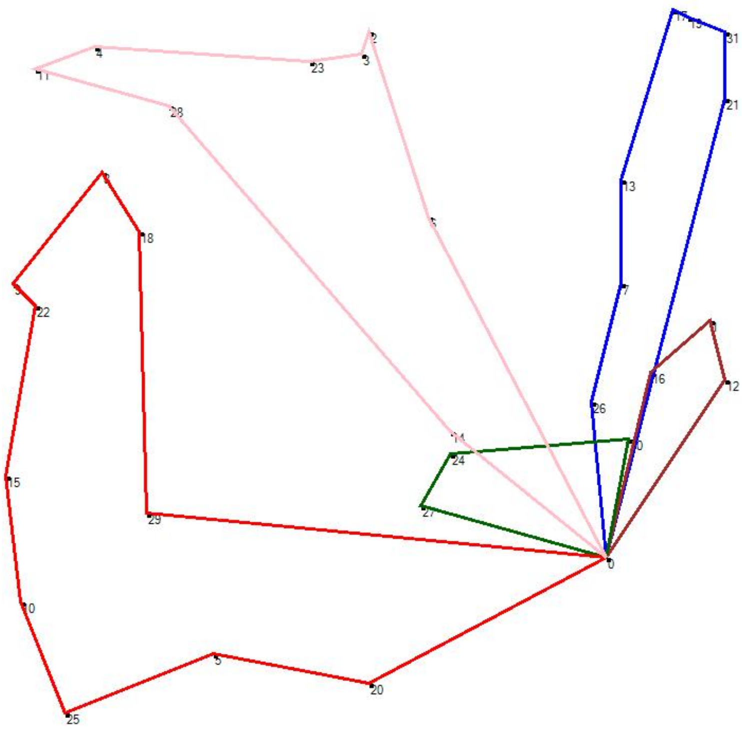
Table 7 demonstrates the routes and times for the solution found by Approach 1. It looks similar to the easy plan with only customer 30 being moved from route 2 to route 3. This move saves £32.1 of cost mainly because of the reduction of delay in serving customer 30.

Table 7 Result for A-n32-k5 (SS) Approach 1

A-n32-k5 (SS) Approach 1												
Route #1:	0	21	31	19	17	13	7	26	0			
Event Times:	0	64	73	78	80	104	118	134	155			
Route #2:	0	12	1	16	0							
Event Times:	78	107	115	126	152							
Route #3:	0	27	24	30	0							
Event Times:	0	26	34	58	74							
Route #4:	0	29	18	8	9	22	15	10	25	5	20	0
Event Times:	0	62	100	109	128	132	155	172	188	210	231	267
Route #5:	0	14	28	11	4	23	3	2	6	0		
Event Times:	0	27	85	104	113	142	149	152	178	230		

Figure 2 illustrates it in a graph.

Figure 2 Graph for A-n32-k5 (SS) Approach 1



Approach 2 produces a very different solution from the other approaches. As can be seen from Table 8, vehicle 3 is delayed instead of vehicle 2. Some customers in route 3 and 4 are swapped so that service delay is reduced. This has caused a total cost reduction of £56.6.

Table 8 Result for A-n32-k5 (SS) Approach 2

A-n32-k5 (SS) Approach 2										
Route #1:	0	21	31	19	17	13	7	26	0	
Event Times:	0	64	73	78	80	104	118	134	155	
Route #2:	0	12	1	16	30	0				
Event Times:	0	29	37	48	57	73				
Route #3:	0	24	5	25	20	0				
Event Times:	78	103	145	167	208	244				
Route #4:	0	27	29	18	8	9	22	15	10	0
Event Times:	0	26	63	101	110	129	133	156	173	252
Route #5:	0	14	28	11	4	23	3	2	6	0
Event Times:	0	27	85	104	113	142	149	152	178	230

The graph for the solution is presented in Figure 3.

Figure 3 Graph for A-n32-k5 (SS) Approach 2

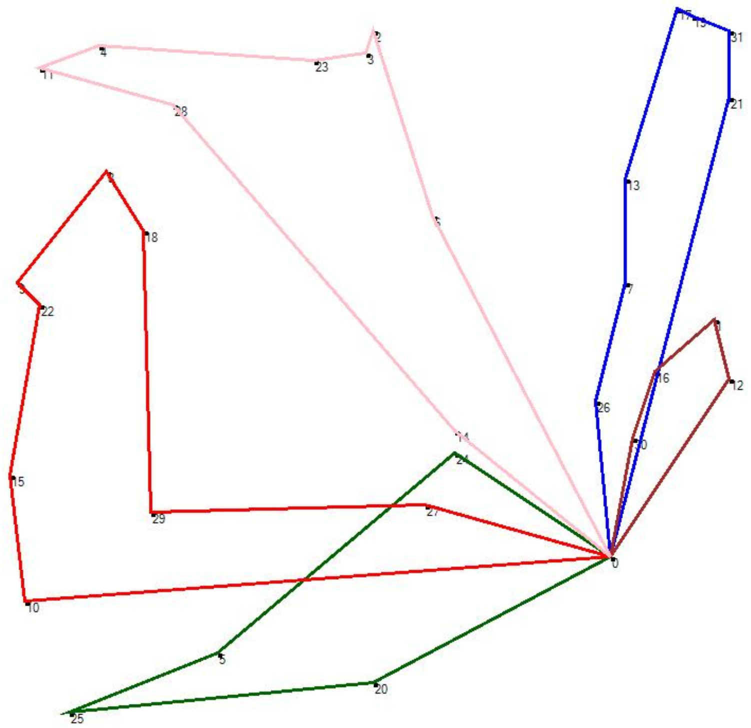


Table 9 shows the routing information obtained by the easy plan for A-n32-k5 (LL-147/235), i.e. large amount of delayed supply (147) with long delay of 235 minutes. Two vehicles are delayed, which are Vehicle 1 and Vehicle 2.

Table 9 Result for A-n32-k5 (LL) Easy plan

A-n32-k5 (LL) Easy Plan												
Route #1:	0	21	31	19	17	13	7	26	0			
Event Times:	235	299	308	313	315	339	353	369	390			
Route #2:	0	12	1	16	30	0						
Event Times:	235	264	272	283	292	308						
Route #3:	0	27	24	0								
Event Times:	0	26	34	59								
Route #4:	0	29	18	8	9	22	15	10	25	5	20	0
Event Times:	0	62	100	109	128	132	155	172	188	210	231	267
Route #5:	0	14	28	11	4	23	3	2	6	0		
Event Times:	0	27	85	104	113	142	149	152	178	230		

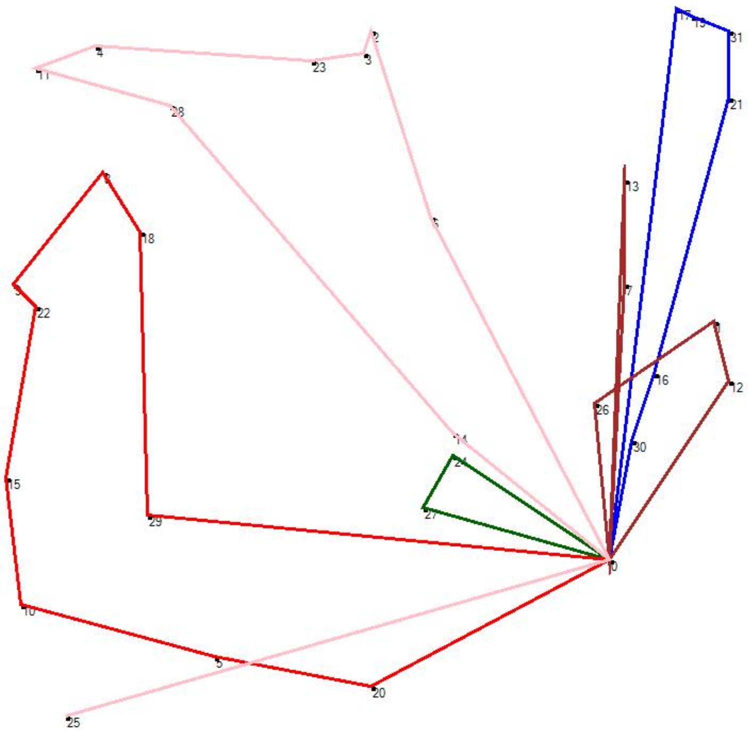
The solution obtained by Approach 1, as demonstrated in Table 10, only delays one vehicle at the start of the day. However, two vehicles (2 and 5) have to do multiple trips and both of them have to wait at the depot for some time until the delayed supply arrives at time 235 so that they can depart for the second trip. A large cost (£1653.1-£1367.4 = £285.7) can be saved by this action.

Table 10 Result for A-n32-k5 (LL) Approach 1

A-n32-k5 (LL) Approach 1													
Route #1:	0	30	16	21	31	19	17	0					
Event Times:	235	251	260	298	307	312	314	389					
Route #2:	0	12	1	26	0	0	13	7	0				
Event Times:	0	29	37	56	77	235	286	300	337				
Route #3:	0	27	24	0									
Event Times:	0	26	34	59									
Route #4:	0	29	18	8	9	22	15	10	5	20	0		
Event Times:	0	62	100	109	128	132	155	172	199	220	256		
Route #5:	0	14	28	11	4	23	3	2	6	0	0	25	0
Event Times:	0	27	85	104	113	142	149	152	178	230	235	311	387

The routes are shown in a graph (Figure 4).

Figure 4 Graph for A-n32-k5 (LL) Approach 1



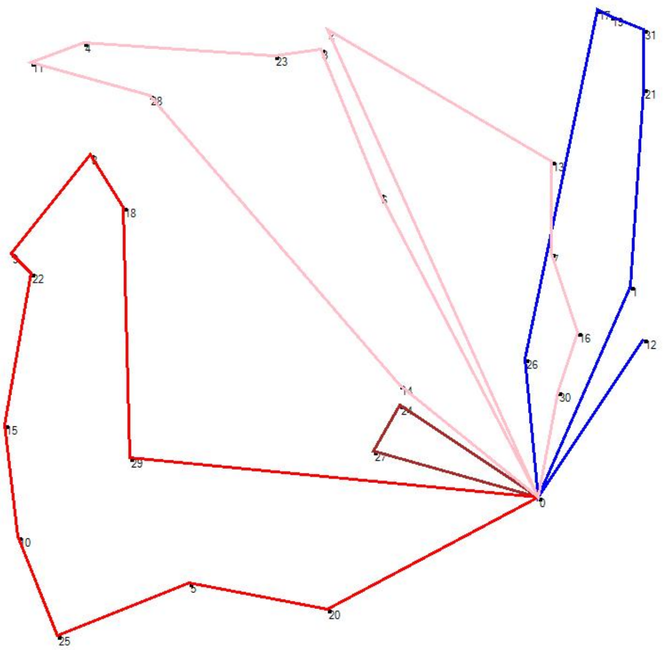
Routes with the same colour means they are served by the same vehicle. Table 11 shows the routing information obtained by using Approach 2.

Table 11 Result for A-n32-k5 (LL) Approach 2

A-n32-k5 (LL) Approach 2																
Route #1:	0	1	21	31	19	17	26	0	0	12	0					
Event Times:	0	35	65	74	79	81	135	156	235	264	293					
Route #2:	0	27	24	0												
Event Times:	235	261	269	294												
Route #3:																
Event Times:																
Route #4:	0	29	18	8	9	22	15	10	25	5	20	0				
Event Times:	0	62	100	109	128	132	155	172	188	210	231	267				
Route #5:	0	14	28	11	4	23	3	6	0	0	30	16	7	13	2	0
Event Times:	0	27	85	104	113	142	149	173	225	235	251	260	273	287	326	404

Again Approach 2 produces a very different solution from the other two. Vehicle 3 does not need to do any delivery at all in this case because of the large amount of delayed supply and the long delay. Only Vehicle 2 is delayed at the start of the day and Vehicle 1 and Vehicle 2 have to do multiple trips. This solution makes a lot less cost than both the easy plan and Approach 1 (£1225.9). The graph is shown in Figure 5.

Figure 5 Graph for A-n32-k5 (LL) Approach 2



Summarised results

Summarised results for all the test problems are shown in Table 12 and Table 13. Table 12 shows the average costs for all the test problems for different choices of cost weights. It also shows the average distance, driver time and delayed service, which are the solution of each single objective, when the aggregated function is optimised. The distance, driver time and delayed service values for easy plan are not affected by the changes of weights as the easy plan is fixed for each test problem. Table 13 summarises the results of the comparison of each approach with the easy plan. Figures in the table represent the average percentage deviations of each approach to the easy plan. It can be observed that on average both Approach 1 and Approach 2 give better results than the easy plan and Approach 1 is also better than Approach 2 on average for every choice of objective weights. It can also be observed that both approaches perform better when the weight of delay (C_3) increases but perform worse when the weight of distance and driver time increase (C_1 and C_2). The reduction of costs is largely caused by the reduction of delayed service and in sacrifice of distance and driver time.

Table 12 Average Costs

Weights	Total Cost			Distance			Driver time			Delayed service		
	Easy	A1	A2	Easy	A1	A2	Easy	A1	A2	Easy	A1	A2
0,1,0	880.70	855.44	869.36	647.80	745.57	789.99	880.70	855.44	869.36	2009.22	1975.63	2136.31
0,0,1	2009.22	1412.23	1702.57	647.80	841.59	894.61	880.70	997.86	1033.38	2009.22	1412.23	1702.57
0.3,0.1,0.5	1287.02	1051.54	1177.01	647.80	738.53	794.11	880.70	896.20	915.72	2009.22	1480.71	1694.40
0.4,0.1,0.5	1351.80	1129.91	1251.12	647.80	733.88	789.58	880.70	895.01	912.04	2009.22	1493.72	1688.18
0.3,0.2,0.5	1375.09	1146.88	1257.44	647.80	733.00	793.91	880.70	886.44	905.84	2009.22	1499.39	1676.20
0.3,0.1,2	4300.85	3194.89	3597.36	647.80	787.36	836.89	880.70	942.87	970.30	2009.22	1432.20	1624.63

Table 13 Average deviation from easy plan

Weights	Total Cost		Distance		Driver time		Delayed service	
	Approach 1	Approach 2	Approach 1	Approach 2	Approach 1	Approach 2	Approach 1	Approach 2
0,1,0	-2.75%	-0.99%	16.07%	22.81%	-2.75%	-0.99%	-21.26%	-8.47%
0,0,1	-42.38%	-24.23%	30.34%	38.34%	12.13%	16.32%	-42.38%	-24.23%
0.3,0.1,0.5	-20.13%	-8.27%	14.87%	23.38%	1.40%	4.30%	-36.38%	-19.55%
0.4,0.1,0.5	-17.67%	-6.82%	14.21%	22.55%	1.54%	3.77%	-35.59%	-19.59%
0.3,0.2,0.5	-18.10%	-8.13%	14.11%	23.27%	0.45%	3.20%	-35.68%	-20.98%
0.3,0.1,2	-32.80%	-19.89%	22.37%	29.63%	6.32%	9.70%	-40.74%	-26.07%

Table 14 and Table 15 compare each approach by the number of times they outperform each other. Approach 1 can almost always find a better solution than the easy plan, whereas there are on average 25 out of 90 cases where Approach 2 failed to produce better results than the easy plan. Although Approach 2 cannot match Approach 1's performance on average, there are still 28 out of 90 cases when it finds better solutions than Approach 1. It can also be observed from the tables that Approach 2 can find a larger number of better results compared to the easy plan if the weight of delayed service increases. However, no obvious patterns can be observed if compared to Approach 1.

Table 14 Number of better results

Weights	A1 Vs Easy	A2 Vs Easy	A1 Vs A2
0,1,0	82	49	52
0,0,1	90	70	56
0.3,0.1,0.5	90	66	58
0.4,0.1,0.5	89	61	57
0.3,0.2,0.5	89	65	54
0.3,0.1,2	90	75	54

Table 15 Number of worse results

Weights	A1 Vs Easy	A2 Vs Easy	A1 Vs A2
0,1,0	0	39	37
0,0,1	0	20	22
0.3,0.1,0.5	0	24	24
0.4,0.1,0.5	0	29	27
0.3,0.2,0.5	0	25	29
0.3,0.1,2	0	15	28

Finally, Table 16 compares the performance of the two approaches when different choices of delayed amount and length of delay are chosen. Each figure in the table represents the average percentage deviation from the easy plan for all the testing problems, if the particular method approach, weights for objectives, and choice of the combination of delayed amount and length of delay are chosen. For example, “-1.14%” in the top left corner represents the average of percentage deviations from the easy plan when applying Approach 1 with small amount of delay, short delay length, and a weights combination of (0, 1, 0).

Table 16 Results for different choices of delay amount and length of delay

Weights	Approaches	SS	SL	LS	LL	XLS	XLL
0,1,0	1	-1.14%	-3.98%	-1.89%	-4.30%	-1.96%	-3.21%
	2	-1.95%	-8.38%	4.63%	-7.08%	9.88%	-3.04%
0,0,1	1	-68.37%	-55.74%	-39.24%	-34.24%	-29.36%	-27.33%
	2	-54.44%	-54.29%	-3.27%	-27.92%	13.17%	-18.62%
0.3,0.1,0.5	1	-18.73%	-31.77%	-15.72%	-23.28%	-13.70%	-17.56%
	2	-13.74%	-31.41%	8.45%	-18.57%	17.06%	-11.43%
0.4,0.1,0.5	1	-16.28%	-26.23%	-14.33%	-20.10%	-13.05%	-16.02%
	2	-11.78%	-28.38%	8.38%	-16.47%	17.27%	-9.92%
0.3,0.2,0.5	1	-16.81%	-26.42%	-14.70%	-21.32%	-11.91%	-17.46%
	2	-12.46%	-28.95%	6.64%	-16.87%	15.30%	-12.41%
0.3,0.1,2	1	-43.81%	-46.64%	-29.38%	-30.85%	-21.80%	-24.34%
	2	-35.35%	-45.51%	-2.66%	-25.50%	8.57%	-18.90%

The first thing that can be observed from this table is that in all weight combination choices except (0, 0, 1), longer delay means larger improvement can be produced by the proposed approaches compared to the easy plan. This is because in the easy plan, not all the available orders are delivered at the earliest possible time which means the longer delay will result in a bigger delay cost. It can also be observed that the larger the delayed amount is, the less improvement can be made. Approach 2 performs badly when a large or extra large amount of orders are delayed for a short time.

Conclusion

This paper presents a Disrupted Capacitated Vehicle Routing Problem with Order Release Delay (DCVRP-ORD), which aims to find an alternative routing plan when the supply of the commodity does not arrive at the depot on time. This problem is different from other types of VRPs because multiple trips may be required for some vehicles and waiting time is also involved. It is formulated as a multiple objective problem and a weighted sum approach has been applied.

Two approaches have been developed. Approach 1 starts with a solution that exactly follows the original routes but has some vehicles waiting at the depot until the full supply amount becomes available. Approach 2 starts with a solution that delivers as much supply as possible at the beginning and leaves those customers that are closest to the depot being served later. A Tabu Search algorithm is then applied to improve the initial solutions for both approaches. Both of the approaches have been compared with an easy plan. A set of 15 standard CVRP test instances have been selected and for each instance, a set of 6 problems have been tested which are defined by different choices of delayed amount (small, large, extra large) and different choices of length of delay (short, long). A set of six weights combinations for objectives has been used. Both approaches perform better than the easy plan especially when the weight of delay increases. The reduction of cost is largely caused by the reduction of delayed service and in sacrifice of distance and driver time. Although Approach 2 cannot match Approach 1's performance on average, it finds better solutions than Approach 1 for approximately one third of the problems.

References

Augerat, P., Bewlenguer, J.M., 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, 1995.

Christofides, N. and Eilon, S. (1969). An algorithm for the vehicle dispatching problem. Operational Research Quarterly, 20: 309–318.

Christofides, N., Mingozzi, A. and Toth, P. (1979). The vehicle routing problem. In N. Christofides, A. Mingozzi, I. Toth, and C. Sandi, eds. Combinatorial Optimization, pp315-338. Wiley: Chichester, UK.

Fisher, M.L. (1994). Optimal solution of vehicle routing problems using minimum k-trees. Operations Research, 42:626-642.

Li, J.Q., Mirchandani, P.B. and Borenstein, D. (2009b). Real-time vehicle rerouting problems with time windows. European Journal of Operational Research, 194: 711-727.

Mu, Q., Fu, Z., Lysgaard, J. and Eglese, R. (2010). Disruption Management of the Vehicle Routing Problem with Vehicle Breakdown. Journal of the Operational Research Society. Published online: 21 April, 2010. doi:10.1057/jors.2010.19

Zhang, X. and Tang, L. (2007). Disruption management for the vehicle routing problem with time windows. In D.S. Huang, L. Heutte and M. Loog, eds. Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques, 2(5): 225-

234. Proceedings of the Third International Conference on Intelligent Computing, ICIC 2007, Qingdao, China, August 21-24, 2007.