Districting for Routing with Stochastic Customers

Hongtao Lei
Gilbert Laporte
Bo Guo

October 2011

CIRRELT-2011-60

Districting for Routing with Stochastic Customers

Hongtao Lei[1,*], **Gilbert Laporte**[2,3], **Bo Guo**[1]

[1] College of Information Systems and Management, National University of Defense Technology, Changsha, Hunan, China P.R., 410073

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[3] Department of Management Sciences, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. We introduce the Vehicle Routing and Districting Problem with Stochastic Customers (VRDPSC). This problem is modeled and solved as a two-stage stochastic program during which the districting decisions are made in the first stage and the Beardwood-Halton-Hammersley formula is used to approximate the expected routing cost of each district in the second stage. District compactness is also considered as part of the objective function. We have developed a large neighbourhood search heuristic for VRDPSC. The heuristic was tested on modified Solomon instances and on modified Gehring and Homberger instances. Extensive computational results confirm the effectiveness of the proposed heuristic.

Keywords. Districting, stochastic vehicle routing, large neighbourhood search metaheuristic.

Acknowledgements. This work was partly supported by the Natural Sciences and Engineering Council of Canada (NSERC) under grants 39682-10, by the Elite Plan Program of the Chinese National University of Defense Technology and by the Chinese National Natural Science Formulation under grand 70971132. This support is gratefully acknowledged.

---

* Corresponding author: hongtaolei@yahoo.cn

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2011

1 Introduction

The problem considered in this paper is the Vehicle Routing and Districting Problem with Stochastic Customers (VRDPSC). It consists of designing the districts for a vehicle routing problem with stochastic customers with the aim of minimizing the expected cost of the solution. The problem is defined on an undirected graph $G = (V, E)$, where $V = \{v_0, \overline{V}, \widetilde{V}\}$ is the vertex set and $E = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ is the edge set. Vertex v_0 is a depot at which are based several identical vehicles, $\overline{V}$ is the set of deterministic (regular) customers, and $\widetilde{V}$ is a set of the stochastic customers whose locations and presence in the solution are uncertain. Thus our setting differs from those of the Probabilistic Traveling Salesman Problem (Jaillet (1988), Laporte et al. (1994)) and of the Vehicle Routing Problem with Stochastic Customers (Gendreau et al., 1996) in which the potential locations of the stochastic customers are known a priori. A symmetric matrix of Euclidean travel times, equal to travel costs, is defined on E. The VRDPSC consists of designing several contiguous vehicle districts such that (1) all customers (including regular and stochastic customers) within the same district are serviced by the same vehicle, (2) each customer vertex is visited once by one vehicle, (3) a service time s is incurred when visiting a vertex, (4) each vehicle route has a normal duration limit h, but overtime is paid at rate θ if its duration exceeds h, and (5) an objective function combining vehicle cost, routing cost and a district compactness measure is minimized. Because of the presence of stochastic customers, the duration of a route in a district is a random variable.

The VRDPSC arises in the operations of courier companies such as DHL, FedEx, TNT Express or UPS. Each driver is assigned a district containing a set of regular customers, but other occasional customers also arise on a stochastic basis. In such contexts it is desirable to consistently assign the same sets of customers to drivers, and hence to create stable districts, in order to improve service (Groër et al., 2009).

The VRDPSC can be modeled and solved as a stochastic mathematical program. The most common solution methodology for this class of problems is called a *priori optimization*, a concept initially proposed by Bertsimas et al. (1990) and applied by several authors to the field of vehicle routing (e.g. Bertsimas (1992), Gendreau et al. (1996), Laporte et al. (2002), Tan et al. (2007), Mendoza et al. (2010), Laporte et al. (2010) and Lei et al. (2011)). In a priori optimization, a *first-stage solution* consisting of a set of districts is first constructed, and the realizations of the random variables (presence or absence of stochastic customers) are then revealed. In the *second-stage solution*, a vehicle route is constructed in each district to serve all its regular and stochastic customers. Whenever the maximal duration of a route is exceeded, an overtime cost is incurred. We therefore solve a stochastic problem with simple recourse.

There exists a rich literature on districting. Most of it deals with deterministic problems. These include the drawing of political districts (Mehrotra et al. (1992), Bozkaya et al. (2003)), the design of school districts (Ferland and Guénette (1990)), the construction of police districts (D'Amico et al. (2002)), districting for home-care services (Blais et al. (2003)), the alignment of commercial territories (Skiera and Albers (1998), Drexl (1999), Kalcsics et al. (2005), Ríos-Mercado and Fernández (2009)), and the solution of location-districting problems (Novaes et al. (2009), Carlsson (2011a)).

Research on stochastic districting problems has mostly been conducted in the context of vehicle routing. Haugland et al. (2007) have considered the problem of designing districts for vehicle routing problems with stochastic demands. The demands are assumed to be uncertain at the time when the districts were made, and these are revealed only after the districting decisions are determined.

A tabu search heuristic was provided for the problem. Carlsson and Delage (2011) introduced a robust framework for distributing the load of a vehicle routing problem over a fleet of vehicles when the location of demand points and their distribution are not known with certainty. Carlsson (2011b) has studied an uncapacitated stochastic vehicle routing problem in which vehicle depot locations are fixed and client locations in a service region are unknown, but are assumed to be independent and identically distributed samples from a given probability density function.

To our knowledge, this paper is the first to consider stochastic customers in the context of a joint vehicle routing and districting problem. Instead of explicitly determining the vehicle routes, we approximate their expected cost by means of the Beardwood-Halton-Hammersley theorem (Beardwood et al., 1959). We integrate this approximation within a large neighbourhood search heuristic for the districting phase.

The remainder of the paper is organized as follows. The mathematical formulation of the problem is described in Section 2. In Section 3, we provide the detailed description of the approximation of the expected cost of routing in a district. In Section 4, we define the compactness measure of a district. A large neighbourhood search metaheuristic for the problem is described in Section 5, followed by computational experiments in Section 6, and by conclusions in Section 7.

2 Mathematical modeling as a stochastic program

The VRDPSC is modeled as a two-stage stochastic program. The first-stage solution is a decomposition of $\overline{V}$ into m districts, $V_1, ..., V_m$. A feasible district plan $x = \{V_1, ..., V_m\}$ must satisfy three conditions: (1) $v_0 \in V_k (k = 1, ..., m)$; (2) $\{V_1 \setminus \{v_0\}, ..., V_m \setminus \{v_0\}\}$ is a partition of $\overline{V}$; (3) the district plan must induce a partition of the region into contiguous districts. After the first-stage solution has been computed, the sets $\widetilde{V}_k$ of stochastic customers are revealed and, in the second-stage solution, the cost of a vehicle route on $\{v_0\} \cup \overline{V}_k \cup \widetilde{V}_k$ is computed for each district k, where $\overline{V}_k = V_k \setminus \{v_0\}$. The workload W_k of district k is approximated as the expected length of an optimal Traveling Salesman Problem (TSP) tour over $\overline{V}_k \cup \widetilde{V}_k$, plus twice the travel time between v_0 and the vertex of $\overline{V}_k$ closest to v_0. The number m of districts is a decision variable.

The VRDPSC consists of computing

$$\min_x \; F(x) = \alpha_m m + \alpha_{erc} F_{erc} + \alpha_{comp} F_{comp} \tag{1}$$

such that $x = \{V_1, ..., V_m\}$ is a feasible districting plan. The objective function is a linear combination of three terms weighted by non-negative user-defined parameters α_m, α_{erc} and α_{comp}. The first term is the number of vehicles. The second term is the expected routing cost in district k. The third term is a compactness measure of the districts. The computations of F_{erc} and F_{comp} are detailed in Section 3 and 4, respectively. Contiguity is enforced through the construction and search mechanisms described in Section 5.

3 Approximation of the expected routing cost in a district

Computing the expected routing cost of a given district requires the solution of a Traveling Salesman Problem (TSP) over all deterministic and stochastic customers. We use the Beardwood-Halton-Hammersley theorem (Beardwood et al., 1959) to approximate this cost.

Theorem 1 *Let $\{X_1, ..., X_n\}, n \geq 1$, be a set of random variables in $\mathbb{R}^{dim}$, independently and identically distributed with compact support. Then the length L^* of a shortest traveling salesman tour through the points X_i satisfies*

$$L^*/n^{(dim-1)/dim} \rightarrow \beta_{dim} \int_{\mathbb{R}^{dim}} f(x)^{(dim-1)/dim} dx, \text{ with probability 1, as } n \rightarrow \infty, \tag{2}$$

where $f(x)$ is the absolutely continuous part of the distribution of the X_i and β_{dim} is a constant which depends on dim but not on the distribution.

Since our problem is defined in two dimensions, the optimal tour cost L_k^* for district k is

$$L_k^* \approx \beta_2 \sqrt{n_k A_k}, \tag{3}$$

where A_k is the area of district k, $n_k = \overline{n}_k + \widetilde{n}_k$ is its number of customers, which includes the $\overline{n}_k$ regular customers and the $\widetilde{n}_k$ stochastic customers, and β_2 is a constant. The value of β_2 is truly asymptotic. Applegate et al. (2006), who have conducted extensive experiments, conclude that β_2 is empirically related to n_k, as shown in Table 1.

Table 1: Empirical value of β_2 as a function of n_k (Applegate et al., 2006)

n_k	β_2
100	0.7764689
200	0.7563542
300	0.7477629
400	0.7428444
500	0.7394544
600	0.7369409
700	0.7349902
800	0.7335751
900	0.7321114
1000	0.7312235
2000	0.7256264

The workload of district k can be calculated as

$$W_k = 2d_k + \beta_2 \sqrt{n_k A_k} + s n_k, \tag{4}$$

where d_k is the shortest driving time between depot and the customer of district k closest to the depot. The computation of (4) is distribution-dependent. For example, assuming the number $\widetilde{n}_k$ of stochastic customers in district k follows a Poisson(λ_k) distribution, the expected cost of routing in district k can be calculated as

$$E(W_k^+) = \sum_{i=0}^{\infty} \frac{e^{-\lambda_k} \lambda_k^i}{i!} (2d_k + \beta_2 \sqrt{(\overline{n}_k + i) A_k} + s(\overline{n}_k + i))^+, \tag{5}$$

where $(\bullet)^+ = (\bullet)$ if $(\bullet) \leq h$, $(\bullet)^+ = h + \theta\big((\bullet) - h\big)$ otherwise, and θ is the overtime rate. Therefore, the expected routing cost of solution x is

$$F_{erc}(x) = \sum_{k=1}^{m} E(W_k^+). \tag{6}$$

Note that the application of the Beardwood-Halton-Hammersley formula is particularly well suited to our problem since it uses no information on the precise location of the stochastic customers. We only require the distribution of their number to compute the expected routing cost.

4 Compactness measure of a district

As in Bozkaya et al. (2003), we use the following formula to measure the compactness of district:

$$F_{comp}(x) = \frac{\sum_{k=1}^{m} B_k(x) - B}{2Bm}, \tag{7}$$

where $B_k(x)$ is the perimeter of district k in solution x, and B is the perimeter of the entire region. This formula computes the average normalized length of the inner boundaries of the districts. It is simple to implement and yields visually compact districts.

5 Large neighbourhood search heuristic

Since our problem embeds a stochastic TSP and should be solved for relatively large sizes, we have devised a large neighbourhood search heuristic for it. This type of heuristic was introduced by Shaw (1997) and has already been successfully applied to several routing problems (e.g. Shaw (1997), Shaw (1998), Ropke and Pisinger (2006), Pisinger and Ropke (2007), Goel and Gruhn. (2008), Laporte et al. (2010), Lei et al. (2011), Hong (2012) and Ribeiro and Laporte (2012)). This metaheuristic must be fine tuned to each application. In this section we describe its application to the VRDPSC.

We obtain a contiguous initial solution by means of a construction heuristic. At each iteration, q boundary units are removed from their districts by using one of the three removal operators, and are reinserted by means of an insertion operator, where q is randomly selected in the interval $[\lceil 0.1n_{bou}\rceil, \lceil 0.2n_{bou}\rceil]$ as in Laporte et al. (2010), and n_{bou} is the total number of the boundary basic units of current solution. The removal operators are randomly selected at each iteration and these remove-insert operators are combined to efficiently explore the solution space. Contiguity is always maintained.

5.1 Objective function

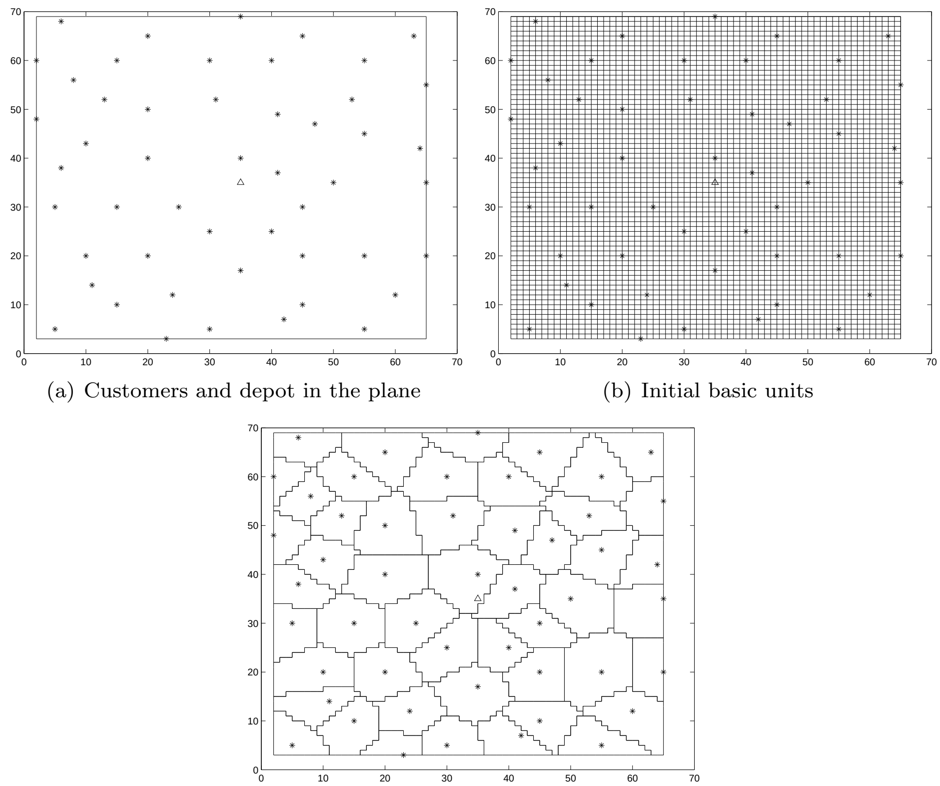
The objective function used in the heuristic search is the following:

$$\min_x F(x) = \alpha_m m + \alpha_{erc} F_{erc} + \alpha_{comp} F_{comp}. \tag{8}$$

In our implementation, α_m is set to 1 and α_{erc} and α_{comp} are tested with different values.

5.2 Definition of the basic units

In order to operationalize the concept of contiguity, it is necessary to embed the regular customers in basic units which partition the region under study. Given a set of regular customer locations, we construct the basic units as follows. Assume that the location of customer i in the instance region is described by the coordinates (x_i, y_i). The region is defined by $\big[x_{min} =$



(a) Customers and depot in the plane

(b) Initial basic units

(c) Final basic units

Figure 1: Generation of the basic units of instance mS-R-50

$\min_{i\in\overline{V}}\{x_i\}, x_{max} = \max_{i\in\overline{V}}\{x_i\}\big]$ and $\big[y_{min} = \min_{i\in\overline{V}}\{y_i\}, y_{max} = \max_{i\in\overline{V}}\{y_i\}\big]$. It is partitioned into n_u initial basic units, where $n_u = \lceil (x_{max} - x_{min})/d \rceil \times \lceil (y_{max} - y_{min})/d \rceil$ and $d = \min\big\{\min_{i,j\in\overline{V}}\{|x_i - x_j|\}, \min_{i,j\in\overline{V}}\{|y_i - y_j|\}\big\}$. Any unit with no regular customer is merged with the nearest unit having at least one regular customer. Figure 1(a) shows the customers and depot in the plane on the instance mS-R-50 described in Section 6.1.1. Figure 1(b) presents the generated initial basic units of instance mS-R-50. Figure 1(c) shows the final basic units of mS-R-50.

Given a set of basic units, which define a partition of the entire region, it is straightforward to use an adjacency list to indicate whether any two basic units are adjacent. Note that two neighbour units which only have a finite number of points in common, as opposed to an edge, are considered to be non-adjacent. To avoid creating the disconnected districts or enclaves, the construction heuristic as well as the removal and insertion operations comply with the following rules: (1) all exchange operations are performed on the boundary units of the districts; (2) operations that would disconnect some units from the remainder of the district are not performed.

5.3 Construction of an initial solution

We have devised the following construction heuristic to generate a good feasible initial solution. The heuristic first randomly selects a basic unit which includes at least one regular customer as the seed unit to initialize the first district. The heuristic gradually extends this district by adjoining to it the adjacent units yielding the least increase in the district workload. The expected workload of the extended district does not exceed the duration limit h. If adjoining an adjacent unit would cause the expected workload to exceed h, this unit is not included in the district but serves as a seed unit for a new district. Any district with only one basic unit is eliminated and merged with the adjacent district so as to yield the lowest increased workload.

5.4 Removal and insertion operators

We now describe three removal operators and one insertion operator.

5.4.1 Long removal operator

This operator concentrates on those districts with longer perimeters and are less likely to be compact.

Step 0. Set $\vartheta = 0$ and $k = 1$.

Step 1. Sort the districts in non-increasing order of the values of their perimeter.

Step 2. If the number $m(x)$ of districts of the current solution x is larger than q, randomly remove a boundary basic unit from each of the first q districts without disconnecting them, and stop.

Step 3. Randomly remove a boundary basic unit from district k without disconnecting it.

Step 4. Set $\vartheta = \vartheta + 1$ and $k = k + 1$. If $k > m(x)$, set $k = 1$. If $\vartheta = q$, stop; otherwise go to Step 3.

5.4.2 Large removal operator

This operator focuses on the districts with larger number of regular customers which are likely to have a larger workload.

Step 0. Set $\vartheta = 0$ and $k = 1$.

Step 1. Sort the districts in non-increasing order of their number of regular customers.

Step 2. If $m(x) > q$, remove the boundary basic unit with the largest number of regular customers from each of the first q districts without disconnecting them, and stop.

Step 3. Remove the boundary basic unit with the largest number of regular customers from district k without disconnecting it.

Step 4. Set $\vartheta = \vartheta + 1$ and $k = k + 1$. If $k > m(x)$, set $k = 1$. If $\vartheta = q$, stop; otherwise go to Step 3.

When each basic unit includes the same number of regular customers, the removed basic unit is randomly chosen from the boundary units with regular customers, without disconnecting the district.

5.4.3 Random removal operator

The operator randomly selects q boundary basic units and removes them from their district without disconnecting it.

5.4.4 Insertion operator

This operator reinserts the removed units into the adjacent districts with the lowest increase in the objective function value.

Step 0. Set $l = 1$.

Step 1. Select a unit u_l from the removed units, and compute the increase in the objective function values when the selected unit is reinserted in the adjacent districts.

Step 2. Choose a adjacent district with the lowest increase in the objective function value as the best district, respecting the tabu tenure described in Section 5.5. If no feasible insertion district is found, go to Step 4.

Step 3. Insert the unit u_l into the best district. If $l = q$, stop; otherwise set $l = l + 1$ and go to Step 1.

Step 4. Use u_l to initialize a new district. Set $m(x) = m(x) + 1$. If $l = q$, stop; otherwise set $l = l + 1$ and go to Step 1.

5.5 Recency-based memory and tabu tenure

As in Bozkaya et al. (2003), we use recency-based memory and tabu tenure to avoid cycling. Any move that reinserts the basic unit i back into its previous district is declared tabu for ϕ iterations. As recommended by Haugland et al. (2007), ϕ is set equal to the number of districts in the initial solution. As usual, tabu move may be still performed if it yields a new incumbent solution.

5.6 Acceptance and stopping criteria

We use the record-to-record travel (RRT) algorithm introduced by Dueck (1993) to define the acceptance criterion for a new solution. Assume f^* is the value of the best current solution, called a record. The unique and positive parameter δ of the RRT algorithm is called a deviation. Let x be a solution, x' a neighbour of x, and $f_{x'}$ the objective value of solution x'. Solution x' is accepted if $f_{x'} < f^* + \delta$, and f^* is updated if $f_{x'} < f^*$. We set $\delta = 0.1f^*$. The search stops if solution quality has not improved for a given number of iterations or if a preset number of iterations have been executed. We set these values as 300 and 1000 respectively in our implementation.

5.7 Summary of the large neighbourhood search heuristic

Our implementation of the large neighbourhood search heuristic can be summarized as follows.

Step 1. Initialize the parameters, and use the construction heuristic to generate an initial solution. Set the objective value of the initial solution as the *record* and the *best cost*, compute the *deviation*. Set the initial solution as the *best solution* and define it as the *current solution*.

Step 2. Randomly select a removal operator from the three removal operators to remove q boundary units from the current solution without disconnecting the districts. Then apply the insertion operator to repair the solution and to generate a new solution, respecting the recency-based memory and tabu tenure.

Step 3. If the objective value of the new solution is smaller than the *best cost*, set the new solution as the *best solution* and set its objective value as the *best cost*. If the new solution is accepted using the RRT criterion, set it as the current solution. If the objective value of the new solution is smaller than the *record*, update the *record* and the *deviation*.

Step 4. Update the total number of the boundary units of current solution, and update the value of q.

Step 5. If the stopping criterion is met, output the *best solution* and the *best cost*. Otherwise go to Step 2.

6 Computational experiments

The algorithm described in Section 5 was coded using Matlab 7.0.4 and run on a laptop with 2 GHz dual processor and 2 GB RAM. We now describe the results of extensive computational experiments.

We assume that all stochastic customers are Poisson distributed and the mean number λ_k of stochastic customers of each district is equal to the number $\overline{n}_k$ of regular customers of the same district.

6.1 Experiments on the modified Solomon instances

We now describe a first set of experiments performed on the modified Solomon instances.

6.1.1 Experimental design

We have generated test instances derived from those of Solomon (1987). The coordinates of the regular customers are the same as in the Solomon instances, while the demands and time windows are not used. We consider six classes of instances: R1, R2, C1, C2, RC1 and RC2. The coordinates of R1 and R2 are the same, and so are those of RC1 and RC2. However, the coordinates of C1 and C2 are not identical. Hence we consider four types of instances: R, C1, C2, and RC. We respectively choose the first 50, 75 and 100 customer vertices as the regular customers in the tests. The service times of customers are equal to 10, and h is set to 480.

Because there are no comparative data and no competing heuristic exist for our problem, comparisons with best known solutions are not possible. However, we can compare the initial solutions generated by our construction heuristic of Section 5.3 with the solutions obtained by our heuristic of Section 5. The detailed information of the modified Solomon instances tested is shown in Table 2.

6.1.2 Computational results

Table 3 presents computational results for the modified Solomon instances with $\alpha_m = 1$, $\alpha_{ecr} = 1$ and $\alpha_{comp} = 1$. The column "Construction heuristic" summarizes the results obtained from the construction heuristic of Section 5.3. The column "LNS heuristic" summarizes the results obtained by applying the heuristic of Section 5. The column "m" gives the number of districts of the solutions. The column "F_{ecr}" presents the expected routing cost of the solutions, computed by Formula (6). The column "F_{comp}" shows the compactness measure cost of the solutions, computed by Formula (7). The column "F" is the total expected cost of the solutions, computed by Formula (8). We also report the total CPU time in seconds in the "Seconds" column for "LNS heuristic". The "Imp(%)"column shows the percentage improvement in "Total cost" obtained by "LNS heuristic", compared with "Construction heuristic".

Table 2: Modified Solomon instances

Instance	Type	$\lvert\overline{V}\rvert$	$E[\lvert\widetilde{V}\rvert]$	s
mS-C1-50	C1	50	50	10
mS-C2-50	C2	50	50	10
mS-R-50	R	50	50	10
mS-RC-50	RC	50	50	10
mS-C1-75	C1	75	75	10
mS-C2-75	C2	75	75	10
mS-R-75	R	75	75	10
mS-RC-75	RC	75	75	10
mS-C1-100	C1	100	100	10
mS-C2-100	C2	100	100	10
mS-R-100	R	100	100	10
mS-RC-100	RC	100	100	10

Table 3: Computational results on the modified Solomon instances

	Construction heuristic				LNS heuristic					
Instance	m	F_{ecr}	F_{comp}	F	m	F_{ecr}	F_{comp}	F	Seconds	Imp(%)
mS-C1-50	4	1536.43	0.25	1540.68	4	1480.61	0.34	1484.95	87.08	3.62
mS-C2-50	4	1685.23	0.25	1689.48	4	1579.45	0.35	1583.80	102.80	6.26
mS-R-50	6	1782.45	0.21	1788.66	4	1567.53	0.34	1571.87	136.86	12.12
mS-RC-50	6	2139.12	0.18	2145.30	4	1961.18	0.30	1965.48	161.33	8.38
50-average	5	1785.80	0.22	1791.02	4	1647.19	0.33	1651.52	122.02	7.59
mS-C1-75	6	2674.04	0.21	2680.25	6	2437.72	0.26	2443.98	297.00	8.82
mS-C2-75	6	2523.72	0.21	2529.93	6	2453.91	0.25	2460.16	169.31	2.76
mS-R-75	10	2537.26	0.17	2547.43	6	2273.95	0.27	2280.22	229.22	10.49
mS-RC-75	7	2745.37	0.18	2752.55	6	2560.07	0.30	2566.37	186.81	6.76
75-average	7.25	2620.20	0.19	2627.54	6	2431.42	0.27	2437.69	220.59	7.21
mS-C1-100	8	3366.82	0.19	3375.01	7	3216.03	0.23	3223.26	346.58	4.50
mS-C2-100	10	3559.60	0.18	3569.78	7	3240.70	0.30	3248.00	347.95	9.01
mS-R-100	10	3217.65	0.18	3227.83	7	2895.51	0.26	2902.77	546.67	10.07
mS-RC-100	12	3624.82	0.15	3636.97	7	3228.23	0.26	3235.49	330.38	11.04
100-average	10	3442.22	0.17	3452.39	7	3155.69	0.26	3162.95	398.97	8.36
Average	**7.42**	**2616.04**	**0.19**	**2623.65**	**5.67**	**2411.43**	**0.29**	**2417.39**	**247.19**	**7.72**

Table 3 clearly shows that the solutions of "LNS heuristic" are better than those of "Construction heuristic". The average improvement percentage in the total cost F is 7.72%. The average number of districts of the solutions of "LNS heuristic" is less than that of "Construction heuristic", and so is the average expected routing cost. However, the average compactness cost of the solutions of "LNS heuristic" is more than that of "Construction heuristic". The average CPU time for "LNS heuristic" is 247.19s.

6.2 Experiments on the modified Gehring and Homberger instances

We next present a second set of experiments performed on the modified Gehring and Homberger instances.

6.2.1 Experimental design

We have also generated test instances derived from those of Gehring and Homberger (1999). Like the modified Solomon instances, the coordinates of the Gehring and Homberger instances are used as the coordinates of the regular customers of the tested instances, and the demands, time windows and service times are not used. We consider six classes of instances: R1, R2, C1, C2, RC1 and RC2. And in the Gehring and Homberger instances, the coordinates of RC1 and RC2 are the same, but the coordinates of C1 and C2 are not identical and neither are those of R1 and R2. Hence we consider five types of instances: R1, R2, C1, C2, and RC. We respectively choose the 150, 200, 300 and 400 customer vertices as the regular customers in the tests, and the regular customers of the instances with 150 regular customers are chosen from the first 150 customer vertices of the instances with 200 customers. The values of the service times of customers and the duration h of each district are the same as those of the modified Solomon instances. The detailed information of the modified Gehring and Homberger instances tested is shown in Table 4.

6.2.2 Computational results

Table 5 shows the computational results for the modified Gehring and Homberger instances with $\alpha_m = 1$, $\alpha_{ecr} = 1$ and $\alpha_{comp} = 1$. It indicates that the solutions of "LNS heuristic" are better than those of "Construction heuristic". The average improvement percentage is 9.46%. Similar to the results on the modified Solomon instances, the average number of districts of the solutions of "LNS heuristic" is less than that of "Construction heuristic", and so is the average expected routing cost, but the average compactness cost of the solutions of "LNS heuristic" is more than that of "Construction heuristic". The average CPU time for "LNS heuristic" is 3023.61s.

6.3 Experiments with different parameters

We have performed tests with different parameters by successively varying the multiplier α_{comp} of F_{comp} and the multiplier α_{erc} of F_{erc} in the objective function. Table 6 provides the solution values obtained with $\alpha_{comp} = 1$, $\alpha_{comp} = 50$ and $\alpha_{comp} = 100$, leaving the other multipliers unchanged. As expected, when the value of α_{comp} becomes larger, the average value of F_{comp} becomes smaller. In contrast, the average values of m and F_{erc} become larger. Figures 2, 3 and 4 show the districts of the final best solution of the instance mS-R-50 when $\alpha_{comp} = 1$, $\alpha_{comp} = 50$ and $\alpha_{comp} = 100$, respectively. We can see that when the value of α_{comp} increases, the districts of the final best solution become more compact. Table 7 provides the comparison of the computational

Table 4: Modified Gehring and Homberger instances

Instance	Type	$\lvert\overline{V}\rvert$	$E[\lvert\widetilde{V}\rvert]$	s
mGH-C1-150	C1	150	150	10
mGH-C2-150	C2	150	150	10
mGH-R1-150	R1	150	150	10
mGH-R2-150	R2	150	150	10
mGH-RC-150	RC	150	150	10
mGH-C1-200	C1	200	200	10
mGH-C2-200	C2	200	200	10
mGH-R1-200	R1	200	200	10
mGH-R2-200	R2	200	200	10
mGH-RC-200	RC	200	200	10
mGH-C1-300	C1	300	300	10
mGH-C2-300	C2	300	300	10
mGH-R1-300	R1	300	300	10
mGH-R2-300	R2	300	300	10
mGH-RC-300	RC	300	300	10
mGH-C1-400	C1	400	400	10
mGH-C2-400	C2	400	400	10
mGH-R1-400	R1	400	400	10
mGH-R2-400	R2	400	400	10
mGH-RC-400	RC	400	400	10

solutions with $\alpha_{ecr} = 0.01$, $\alpha_{ecr} = 0.1$ and $\alpha_{ecr} = 1$, leaving the other multipliers unchanged. Like in Table 6, increasing α_{ecr} means that the average value of F_{erc} becomes smaller, and the average values of m and F_{comp} become larger.

7 Conclusions

We have introduced, modeled and solved a combined vehicle routing and districting problem with stochastic customers. The problem was solved by means of a two-stage program. In the first stage, the districting decision is made. The second stage expected routing cost of each district is approximated by the Beardwood-Halton-Hammersley formula. We have developed a large neighbourhood search heuristic for the problem. Modified Solomon instances and modified Gehring and Homberger instances were used to assess the quality of the proposed heuristic. The computational results confirm the effectiveness of our approach.

Acknowledgements

This work was partly supported by the Canadian Natural Sciences and Engineering Research Council under grant 39682-10, by the Elite Plan Program of the Chinese National University of Defense Technology and by the Chinese National Natural Science Foundation under grant 70971132. This support is gratefully acknowledged.

Table 5: Computational results on the modified Gehring and Homberger instances

	Construction heuristic				LNS heuristic					
Instance	m	F_{ecr}	F_{comp}	F	m	F_{ecr}	F_{comp}	F	Seconds	Imp(%)
mGH-C1-150	17	6322.66	0.15	6339.81	14	5816.51	0.20	5830.71	819.81	8.03
mGH-C2-150	16	6148.09	0.15	6164.24	12	5617.07	0.21	5629.28	790.77	8.68
mGH-R1-150	17	6465.75	0.15	6482.90	13	5908.93	0.19	5922.12	838.50	8.65
mGH-R2-150	16	6302.94	0.17	6319.11	13	5807.17	0.20	5820.37	795.20	7.89
mGH-RC-150	15	6248.89	0.15	6264.04	13	5815.67	0.20	5828.87	798.55	6.95
150-average	16.20	6297.66	0.15	6314.01	13	5793.07	0.20	5806.27	808.57	8.04
mGH-C1-200	20	8138.64	0.15	8158.79	17	7582.57	0.18	7599.75	1338.98	6.85
mGH-C2-200	21	7961.85	0.13	7982.98	16	7178.10	0.18	7194.28	1389.89	9.88
mGH-R1-200	23	8446.55	0.14	8469.69	17	7371.73	0.17	7388.90	1550.95	12.76
mGH-R2-200	23	8300.51	0.13	8323.64	16	7329.83	0.20	7346.03	1743.02	11.74
mGH-RC-200	26	8659.73	0.11	8685.84	16	7547.65	0.18	7563.83	1409.05	12.92
200-average	22.60	8301.46	0.13	8324.19	16.40	7401.98	0.18	7418.56	1486.38	10.83
mGH-C1-300	40	15612.43	0.10	15652.53	28	13550.48	0.14	13578.62	3805.09	13.25
mGH-C2-300	33	13564.15	0.12	13597.27	27	12406.86	0.15	12434.01	3462.64	8.56
mGH-R1-300	37	14900.03	0.11	14937.14	28	13120.54	0.15	13148.69	3685.58	11.97
mGH-R2-300	37	14623.46	0.11	14660.57	29	13028.82	0.15	13057.97	3507.06	10.93
mGH-RC-300	33	14502.79	0.11	14535.90	30	13494.62	0.14	13524.76	3420.94	6.96
300-average	36	14640.57	0.11	14676.68	28.40	13120.26	0.15	13148.81	3576.26	10.33
mGH-C1-400	50	19740.44	0.09	19790.53	37	17994.84	0.12	18030.96	6507.61	8.89
mGH-C2-400	45	17506.35	0.09	17551.44	32	15791.24	0.14	15823.38	6638.89	9.85
mGH-R1-400	49	18824.63	0.10	18873.73	36	16886.36	0.14	16922.50	6363.39	10.34
mGH-R2-400	45	18151.06	0.11	18196.17	39	16805.59	0.14	16844.73	5969.23	7.43
mGH-RC-400	46	18602.25	0.10	18648.35	40	17358.87	0.12	17397.99	5637.13	6.70
400-average	47	18564.95	0.10	18612.04	36.80	16967.38	0.13	17003.91	6223.25	8.64
Average	**30.45**	**11951.16**	**0.12**	**11981.73**	**23.65**	**10820.67**	**0.17**	**10844.39**	**3023.61**	**9.46**

Table 6: Computational results on the modified Solomon instances with different value of α_{comp}

	$\alpha_{comp} = 1$				$\alpha_{comp} = 50$				$\alpha_{comp} = 200$			
Instance	m	F_{ecr}	F_{comp}	F	m	F_{ecr}	F_{comp}	F	m	F_{ecr}	F_{comp}	F
mS-C1-50	4	1480.61	0.34	1484.95	4	1485.20	0.33	1505.70	4	1491.60	0.23	1541.10
mS-C2-50	4	1579.45	0.35	1583.80	4	1571.71	0.30	1590.78	4	1606.85	0.24	1659.49
mS-R-50	4	1567.53	0.34	1571.87	4	1571.02	0.27	1588.30	4	1573.18	0.22	1620.20
mS-RC-50	4	1961.18	0.30	1965.48	4	1959.47	0.27	1977.15	4	1954.15	0.27	2011.43
mS-C1-75	6	2437.72	0.26	2443.98	5	2503.03	0.26	2520.81	5	2479.31	0.26	2537.00
mS-C2-75	6	2453.91	0.25	2460.16	6	2491.32	0.24	2509.23	6	2472.39	0.28	2534.29
mS-R-75	6	2273.95	0.27	2280.22	6	2253.35	0.26	2272.30	6	2256.69	0.30	2321.92
mS-RC-75	6	2560.07	0.30	2566.37	6	2567.33	0.26	2586.45	6	2562.24	0.22	2611.48
mS-C1-100	7	3216.03	0.23	3223.26	8	3203.15	0.25	3223.75	8	3172.95	0.21	3222.74
mS-C2-100	7	3240.70	0.30	3248.00	7	3184.94	0.27	3205.45	9	3259.26	0.20	3308.83
mS-R-100	7	2895.51	0.26	2902.77	7	2910.06	0.31	2932.37	7	2910.46	0.25	2968.33
mS-RC-100	7	3228.23	0.26	3235.49	8	3228.71	0.24	3248.88	7	3232.38	0.31	3301.21
Average	**5.67**	**2407.91**	**0.29**	**2413.87**	**5.75**	**2410.77**	**0.27**	**2430.10**	**5.83**	**2414.29**	**0.25**	**2469.84**

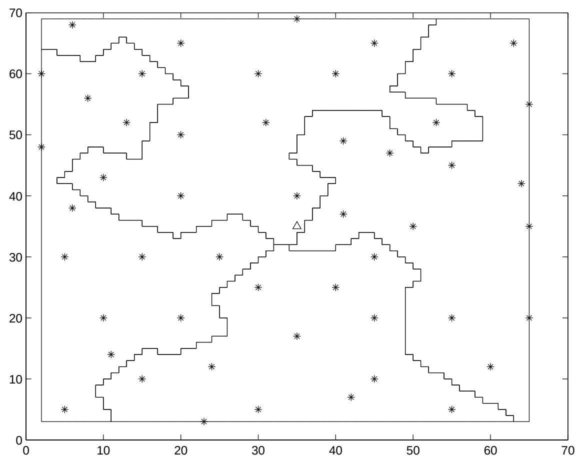


Figure 2: Solution for instance mS-R-50 with $\alpha_{comp} = 1$

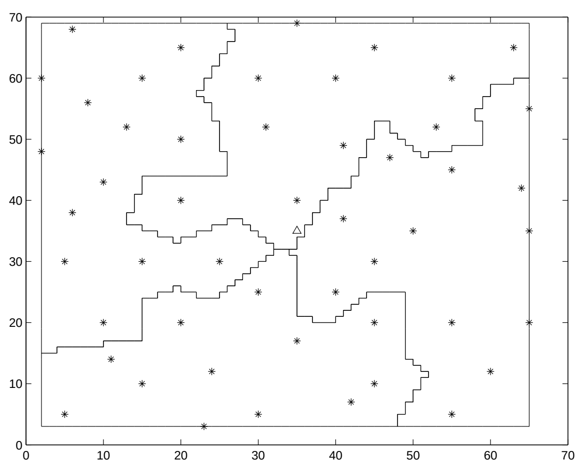


Figure 3: Solution for instance mS-R-50 with $\alpha_{comp} = 50$

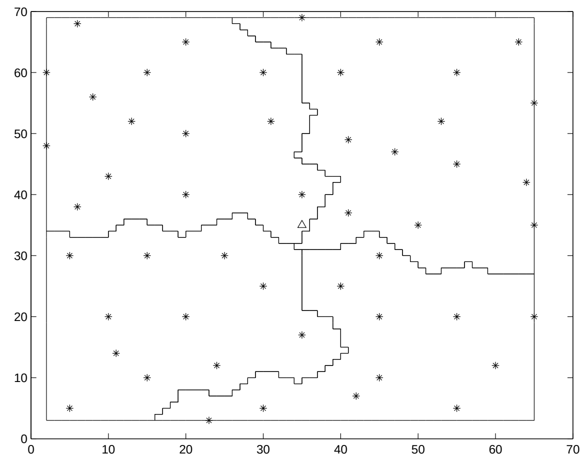


Figure 4: Solution for instance mS-R-50 with $\alpha_{comp} = 200$

Table 7: Computational results on the modified Solomon instances with different value of α_{erc}

	$\alpha_{erc} = 0.01$				$\alpha_{erc} = 0.1$				$\alpha_{erc} = 1$			
Instance	m	F_{ecr}	F_{comp}	F	m	F_{ecr}	F_{comp}	F	m	F_{ecr}	F_{comp}	F
mS-C1-50	3	1511.60	0.25	18.36	4	1486.01	0.33	152.93	4	1480.61	0.34	1484.95
mS-C2-50	4	1586.70	0.28	20.15	4	1582.41	0.33	162.57	4	1579.45	0.35	1583.80
mS-R-50	4	1569.68	0.25	19.95	4	1636.59	0.25	167.91	4	1567.53	0.34	1571.87
mS-RC-50	4	1975.59	0.28	24.04	4	1953.89	0.27	199.66	4	1961.18	0.30	1965.48
mS-C1-75	5	2540.38	0.24	30.64	6	2465.22	0.23	252.75	6	2437.72	0.26	2443.98
mS-C2-75	5	2485.32	0.31	30.16	6	2449.61	0.28	251.24	6	2453.91	0.25	2460.16
mS-R-75	5	2263.35	0.25	27.89	5	2285.50	0.33	233.88	6	2273.95	0.27	2280.22
mS-RC-75	5	2596.50	0.28	31.25	5	2574.01	0.25	262.66	6	2560.07	0.30	2566.37
mS-C1-100	7	3200.60	0.25	39.26	7	3196.75	0.24	326.91	7	3216.03	0.23	3223.26
mS-C2-100	7	3261.55	0.26	39.88	7	3206.39	0.25	327.89	7	3240.70	0.30	3248.00
mS-R-100	6	2992.81	0.26	36.19	7	2899.58	0.27	297.23	7	2895.51	0.26	2902.77
mS-RC-100	7	3231.04	0.28	39.59	7	3251.33	0.28	332.41	7	3228.23	0.26	3235.49
Average	**5.17**	**2434.59**	**0.27**	**29.78**	**5.50**	**2415.61**	**0.28**	**247.34**	**5.67**	**2407.91**	**0.29**	**2413.87**

References

Applegate, D. L., R. E. Bixby, V. Chvátal, W. J. Cook. 2006. The Travelling Salesman Problem: A Computational Study. Princeton University Press.

Beardwood, J., J.H. Halton, J.M. Hammersley. 1959. The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society* **55**(4). Cambridge University Press, 299–327.

Bertsimas, D.J. 1992. A vehicle routing problem with stochastic demand. *Operations Research* **40**(3) 574–585.

Bertsimas, D.J., P. Jaillet, A.R. Odoni. 1990. A priori optimization. *Operations Research* **38**(6) 1019–1033.

Blais, M., S.D. Lapierre, G. Laporte. 2003. Solving a home care districting problem in an urban setting. *Journal of the Operational Research Society* **54**(11) 1141–1147.

Bozkaya, B., E. Erkut, G. Laporte. 2003. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research* **144**(1) 12–26.

Carlsson, J.G. 2011a. Dividing a territory among several facilities. *INFORMS Journal of Computing*, to appear.

Carlsson, J.G. 2011b. Equitable partitioning for multi-depot vehicle routing. *INFORMS Journal of Computing*, to appear.

Carlsson, J.G., É. Delage. 2011. Robust Partitioning for Stochastic Multi-Vehicle Routing. Technical report of the Boeing Company.

D'Amico, S.J., S.J. Wang, R. Batta, C.M. Rump. 2002. A simulated annealing approach to police district design. *Computers & Operations Research* **29**(6) 667–684.

Drexl, A., K. Haase. 1999. Fast approximation methods for sales force deployment. *Management Science* **45**(10) 1307–1323.

Dueck, G. 1993. New Optimization Heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics* **104**(1) 86–92.

Ferland, J.A., G. Guénette. 1990. Decision support system for the school districting problem. *Operations Research* **38**(1) 15–21.

Fleischmann, B., J. Paraschis. 1988. Solving a large scale districting problem: A case report. *Computers & Operations Research* **15**(6) 521–533.

Gehring, H., J. Homberger. 1999. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In: Miettinen, K., M. Mäkelä, J. Toivanen (eds.), *Proceedings of EUROGEN99 - Short Course on Evolutionary Algorithms in Engineering and Computer Science, Reports of the Department of Mathematical Information Technology*, Series A. Collections, No. A 2/1999. University of Jyväskylä 57–64.

Gendreau, M., G. Laporte, R. Séguin. 1996. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research* **44**(3) 469–477.

Goel, A., V. Gruhn. 2008. A general vehicle routing problem. *European Journal of Operational Research* **191**(3) 650–660.

Groër, C., B. Golden, E. Wasil. 2009. The consistent vehicle routing problem. *Manufacturing & Service Operations Management* **11**(4) 630–643.

Haugland, D., S. C. Ho, G. Laporte. 2007. Designing delivery districts for the vehicle routing problem with stochastic demands. *European Journal of Operational Research* **180**(3) 997–1010.

Hong, L. 2011. An improved LNS algorithm for real-time vehicle routing problem with time windows. *Computers & Operations Research* **39**(2) 151–163.

Jaillet, P. 1988. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research* **36**(6) 929–936.

Kalcsics, J., S. Nickel, M. Schröder. 2005. Towards a unified territorial design approach - Applications, algorithms and GIS integration. *Top* **13**(1) 1–56.

Laporte, G., F.V. Louveaux, H. Mercure. 1994. A priori optimization of the probabilistic traveling salesman problem. *Operations Research* **42**(3) 543–549.

Laporte, G., F.V. Louveaux, L. Van hamme. 2002. An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research* **50**(3) 415–423.

Laporte, G., R. Musmanno, F. Vocaturo. 2010. An adaptive large neighbourhood search heuristic for the capacitated arc routing problem with stochastic demands. *Transportation Science* **44**(1) 125–135.

Lei, H., G. Laporte, B. Guo. 2011. The capacitated vehicle routing problem with stochastic demands and time windows. *Computers & Operations Research* **38**(12) 1775–1783.

Mehrotra, A., E.L. Johnson, G.L. Nemhauser. 1992. An optimization based heuristic for political districting. *Management Science* **44**(8) 1100–1114.

Mendoza, J.E., B. Castanier, C. Guéret, A.L. Medaglia, N. Velasco. 2010. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research* **37**(11) 1886–1898.

Novaes, A.G.N., J.E. Souza de Cursi, A.C.L. da Silva, J.C. Souza. 2009. Solving continuous location-districting problems with Voronoi diagrams. *Computers & Operations Research* **36**(1) 40–59.

Pisinger, D., S. Ropke. 2007. A general heuristic for vehicle routing problems. *Computers & Operations Research* **34**(8) 2403–2435.

Ribeiro, G.M., G. Laporte. 2012. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research* **39**(3) 728–735.

Ríos-Mercado, R.Z., E. Fernández. 2009. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. *Computers & Operations Research* **36**(3) 755–776.

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

Shaw, P. 1997. A new local search algorithm providing high quality solutions to vehicle routing problems. Technical report, Department of Computer Science, University of Strathclyde, Glasgow.

Shaw, P. 1998. Using constraint programming and local search methods to solve vehicle routing problems. *Principles and Practice of Constraint Programming - CP98*, *Lecture Notes in Computer Science*, vol. 1520. Springer-Verlag, Berlin, 417–431.

Skiera, B., S. Albers. 1998. COSTA: Contribution optimizing sales territory alignment. *Marketing Science* **17** (3) 196–213.

Solomon, M.M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* **35**(2) 254–265.

Tan, K.C., C.Y. Cheong, C.K. Goh. 2007. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research* **177**(2) 813–839.