Computers & Operations Research 56 (2015) 84–92

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Dynamic design of sales territories

Hongtao Lei [a,*], Gilbert Laporte [b], Yajie Liu [a], Tao Zhang [c]

[a] College of Information Systems and Management, National University of Defense Technology, Changsha, Hunan, 410073, PR China
[b] Canada Research Chair in Distribution Management and CIRRELT, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montreal, Canada H3T 2A7
[c] College of Information Systems and Management and State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha, Hunan, 410073, P.R. China

ARTICLE INFO

Available online 26 November 2014

Keywords:
Districting
Traveling salesman
Adaptive large neighbourhood search

ABSTRACT

We introduce the Multiple Traveling Salesmen and Districting Problem with Multi-periods and Multi-depots. In this problem, the compactness of the subdistricts, the dissimilarity measure of districts and an equity measure of salesmen profit are considered as part of the objective function, and the salesman travel cost on each subdistrict is approximated by the Beardwood–Halton–Hammersley formula. An adaptive large neighbourhood search metaheuristic is developed for the problem. It was tested on modified Solomon and Gehring & Homberger instances. Computational results confirm the effectiveness of the proposed metaheuristic.

1. Introduction

The problem considered in this paper is the Multiple Traveling Salesman and Districting Problem with Multi-Periods and Multi-Depots, where the customers of a sales territory dynamically evolve over the periods of a planning horizon. The problem consists of designing districts and subdistricts for a multiple traveling salesman problem with dynamic customers over several periods. Each salesman services all customers of his district over the planning horizon but performs a single route in a subdistrict in each period. The customers on the territory vary dynamically over the planning horizon. A proportion η of the customers of the previous period leave the territory, and a proportion ψ of new customers enter it. Typically the number of customers of a territory will tend to increase over time. However, all information on customers, which includes their number and locations, is available at the beginning of each period. There also exist several depots on the territory at reasonable locations. The number of depots and their locations are static over time. Fig. 1 depicts a territory partitioned into districts and subdistricts, with depots, periods and salesmen's routes.

The problem is defined on an undirected graph $G=(V,E,P)$, where $P=\{1,\dots,T\}$ is the set of periods, $V=D\cup V^1\cup\dots\cup V^T$ is the vertex set, D is the depot set at which the salesmen are located, V^t is the set of customers at period t, and $E=\{(v_i,v_j): v_i,v_j\in V, i<j\}$ is the edge set. A symmetric matrix of Euclidean travel times, equal to travel costs, is defined on E. The problem consists of designing several contiguous districts served in each period and subdistricts served in each working day such that (1) all customers within the same district are served by the same salesman, (2) each customer is visited once by one salesman, (3) a service time s is incurred when visiting a customer, (4) each salesman 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 salesman cost (number of districts), a subdistrict compactness measure, a district partition dissimilarity measure and a salesmen profit equity measure is minimised.

Several companies, such as Coca-Cola, DHL and FedEx, face this problem. They need to segment or partition their customers into clusters or territories in order to efficiently handle marketing and distribution decisions over different periods, and the customer base is not static. In such contexts, it is desirable to consistently assign almost the same customers to each salesman, to create relatively stable districts, and to design equitable subdistricts in terms of workload.

There exists a rich literature on districting. Most of it deals with deterministic problems. The relevant papers include the drawing of political districts [28,5,6,21] the design of school districts [14], the construction of police districts [10], districting for home-care services [4], the alignment of commercial territories [37,12,20,31,26], and the solution of location-districting problems [29,7]. Research on stochastic districting problems has mostly been conducted in the context of vehicle routing. Haugland et al. [18] 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 are designed, and these are revealed only after the districting decisions are determined. A tabu search heuristic was provided for the problem. Lei et al. [24] proposed a vehicle routing and districting problem with stochastic customers. The problem was modeled and solved as a two-stage stochastic program in which the districting

* Corresponding author.
E-mail addresses: hongtaolei@aliyun.com, leihongtao@nudt.edu.cn (H. Lei), gilbert.laporte@cirrelt.ca (G. Laporte), liuyajie@nudt.edu.cn (Y. Liu), zhangtao@nudt.edu.cn (T. Zhang).

http://dx.doi.org/10.1016/j.cor.2014.11.008
0305-0548/

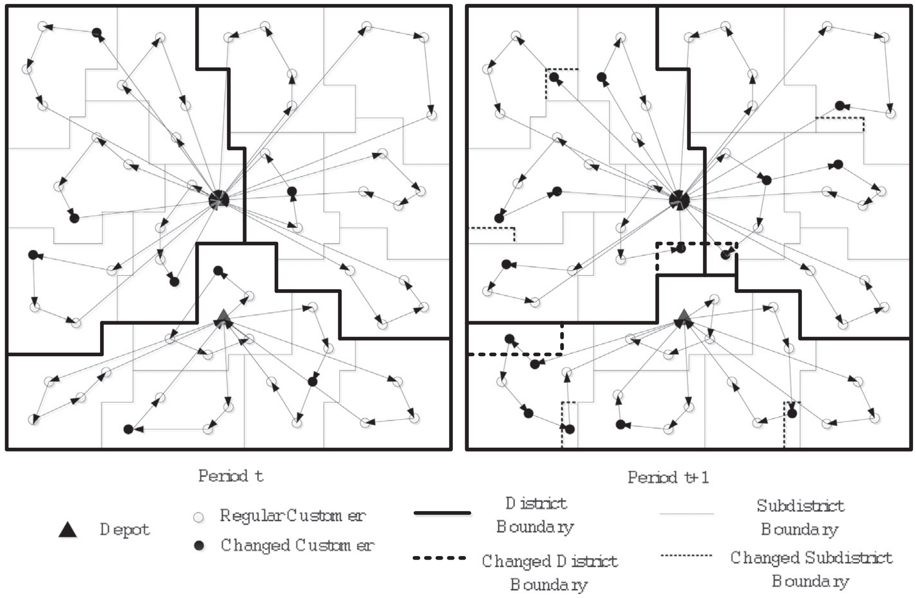

Fig. 1. An example of a territory partitioned into districts and subdistricts, with two depots and salesmen's routes over two periods.

decisions are made in the first stage and the Beardwood–Halton–Hammersley formula was used to approximate the expected routing cost of each district in the second stage. A large neighbourhood search metaheuristic was also developed for the problem. Carlsson and Delage [9] 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 [8] has studied an uncapacitated stochastic vehicle routing problem in which vehicle depot locations are fixed and customer locations in a service region are unknown, but are assumed to be independent and identically distributed from a given probability density function.

To the best of our knowledge, this paper is the first to consider dynamic customers in the context of a joint multiple traveling salesmen and districting problem with multi-periods and multi-depots. We have considered the salesman cost, the subdistrict compactness measure, a district partition dissimilarity measure and a salesman profit equity measure in the objective function. Instead of explicitly determining the salesman routes, we approximate their cost by means of the Beardwood–Halton–Hammersley theorem [3]. We integrate this approximation within a large neighbourhood search metaheuristic for the districting phase.

The remainder of the paper is organised as follows. The mathematical model is presented in Section 2. An adaptive large neighbourhood search metaheuristic for the problem is described in Section 3, followed by computational experiments in Section 4, and by conclusions in Section 5.

2. Mathematical model

We introduce the following additional notation: $V^t=\{1,\dots,n^t\}$ is the set of the customers on the territory at period t, where n^t is the associated number of the customers; $D=\{D_1,\dots,D_Z\}$ is the set of the depots, and D_k^t is the depot assigned to district k at period t; m^t is the number of districts at period t; V_k^t is the customer set which are located in the district k at period t, $V_{k,i}^t$ is the customer set which are located in the subdistrict i of district k at period t, and n_k^t is the number of the customers in district k at period t; a is the unit revenue generated by serving a customer; h is the duration limit of a route; $d_{k,i}^t$ is the distance between D_k^t and the customer of $V_{k,i}^t$ closest to D_k^t.

We assume that a period lasts several weeks with a maximum number of working days each week (e.g., from Monday to Friday). The problem is modeled as follows. For period t, the solution is a decomposition of V^t into m^t districts, and partition of district V_k^t into s_k^t subdistricts, each of which corresponding to a salesman tour on a working day. A feasible district and subdistrict plan $x=\left\{V_1^t\{V_{1,1}^t,\dots,V_{1,s_1^t}^t\},\dots,V_{m^t}^t\{V_{m^t,1}^t,\dots,V_{m^t,s_{m^t}^t}^t\}\right\}$ must satisfy following constraints: (1) $\forall D_k^t, D_k^t\in D$; (2) $\{V_1^t,\dots,V_{m^t}^t\}$ is a partition of V^t; (3) $\{V_{k,1}^t,\dots,V_{k,s_k^t}^t\}$ is a partition of district V_k^t.

After the design of districts and subdistricts, the closest depot $D_k^t(D_k^t\in D)$ to the district V_k^t is assigned to the district, and the cost of the salesman tour on $\{D_k^t\}\cup V_{k,i}^t$ is computed for each subdistrict $V_{k,i}^t$. The workload of a subdistrict $V_{k,i}^t$ is approximated as the length of an optimal traveling salesman problem tour over $V_{k,i}^t$, plus twice the distance $d_{k,i}^t$ between D_k^t and the customer of $V_{k,i}^t$ closest to D_k^t. The number m^t of designed districts at period t is a decision variable.

The objective of the model is

$$\min_x F(x)=\sum_{t=1}^{T}\left(\alpha_m m^t+\alpha_{comp}F_{comp}^t(x)+\alpha_{dissim}F_{dissim}^t(x)+\alpha_{equ}F_{equ}^t(x)\right),\tag{1}$$

where x denotes a feasible solution. The objective function minimises the sum over m^t of districts, of the compactness measure $F_{comp}^t(x)$ of the subdistricts, the dissimilarity measure $F_{dissim}^t(x)$ of the district partition, and of the equity measure $F_{equ}^t(x)$ of the salesmen over all periods, weighted by the positive user-defined parameter α_m, α_{comp}, α_{dissim} and α_{equ}. The computation of $F_{comp}^t(x)$, $F_{dissim}^t(x)$ and $F_{equ}^t(x)$ is detailed in Sections 2.1, 2.2 and 2.3 respectively.

2.1. Compactness measure of the subdistricts

As in Bozkaya et al. [6], we use the following formula to measure the compactness of a subdistrict:

$$F_{comp}^t(x)=\left(\sum_{k=1}^{m^t}\sum_{i=1}^{s_k^t}B_{k,i}^t(x)-B^t\right)\Bigg/\left(2B^t\sum_{k=1}^{m^t}s_k^t\right),\tag{2}$$

where $B_{k,i}^t(x)$ and B^t are respectively the perimeters of subdistrict $V_{k,i}^t$ and of the entire territory at period t in solution x, s_k^t

is the number of subdistricts of district V_k^t and m^t is the number of districts.

2.2. Dissimilarity measure of a district partition

As is sometimes the case in some vehicle routing settings (e.g., [17]), the salesmen may prefer to visit more or less the same set of customers over time. We have therefore developed a measure to compare the dissimilarity of the districts of one period with that of next period, based on Bozkaya et al. [5]:

$$F^t_{dissim}(x) = 1 - \sum_{k=1}^{m^t} O^t_k(x)/A^t, \tag{3}$$

where $O^t_k(x)$ is the largest overlap area of district V_k^t at period t in solution x compared to the districts at period $t-1$, m^t is the number of districts, and A^t is the area of the entire territory at period t. When $t=1$, $F^t_{dissim}(x)=0$.

2.3. Equity measure of the salesmen's profit

The equity $F^t_{equ}(x)$ of the salesmen's profit can be measured by the profit $P^t_k(x)$ generated by visiting the customers of their district. Its objective to minimise the variance on the average profit P^t of the salesmen. This measure can be computed as

$$F^t_{equ}(x) = \sum_{k=1}^{m^t} |P^t_k(x) - P^t|/(m^t P^t), \tag{4}$$

where $P^t = \sum_{k=1}^{m^t} P^t_k(x)/m^t$ and $P^t_k(x) = an^t_k - \sum_{i=1}^{s^t_k} F^t_{rout\ k,i}(x)$. In formula (4), the profit $P^t_k(x)$ of the salesman assigned to district k at period t is equal to the revenue derived from serving customers in the district, minus the sum of the travel costs in the subdistricts over s^t_k working days. $F^t_{rout\ k,i}(x)$ is the routing cost of a salesman in subdistrict i of district k at period t, and its detailed computation is described in Section 2.4.

2.4. Approximation of the routing cost in a subdistrict

Computing the salesman travel cost of a given subdistrict requires the solution of a Traveling Salesman Problem (TSP) over all customers of subdistrict. We use the Beardwood–Halton–Hammersley theorem [3] to approximate the tour cost of a given subdistrict.

Theorem 1. *Let $\{X_1,\dots,X_n\}, n\geq 1$, 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*

$$L^*/n^{(dim-1)/dim} \to \beta_{dim} \int_{\mathbb{R}^{dim}} f(x)^{(dim-1)/dim}\, dx, \text{ with probability 1, as } n\to\infty, \tag{5}$$

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 $L^{t*}_{k,i}$ for subdistrict $V^t_{k,i}$ reduces to

$$L^{t*}_{k,i} \approx \beta_2 \sqrt{n^t_{k,i} A^t_{k,i}}, \tag{6}$$

where $A^t_{k,i}$ is the area of subdistrict $V^t_{k,i}$, $n^t_{k,i}$ is its number of customers, and β_2 is a constant. The value of β_2 is truly asymptotic. Applegate et al. [1], who have conducted extensive experiments, conclude that β_2 is empirically related to $n^t_{k,i}$ for values varying between 100 and 2000, as shown in Table 1.

Considering that the salesman usually will not visit more than 100 customers per day in his subdistrict, we have also conducted extensive experiments to estimate the value of β_2 for values of $n^t_{k,i} (n^t_{k,i} = 20, \dots, 90)$. Since we are solving smaller instances than Applegate et al. [1] we have conducted our own experiments. We have randomly generated 100 instances on the 100×100 square for each size $(n^t_{k,i}=20,\dots,90)$ and each instance was optimally solved by the software Concorde Applegate et al. [1] which is publicly available. The average values of β_2 and standard deviations σ_2 on these instances are reported in Table 2.

Table 1
Empirical value of β_2 as a function of $n^t_{k,i}$ Applegate et al. [1].

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

Table 2
Experimental estimated value of β_2 and σ_2 as a function of $n^t_{k,i}$.

n_k	β_2	σ_2	σ_2/β_2 (%)
20	0.8584265	0.0688363	8.02
30	0.8269698	0.0508781	6.15
40	0.8129900	0.0431063	5.30
50	0.7994125	0.0363534	4.55
60	0.7908632	0.0276979	3.50
70	0.7817751	0.0303670	3.88
80	0.7775367	0.0252234	3.24
90	0.7773827	0.0254466	3.27

Table 3
Modified Solomon and Gehring & Homberger instances.

Instance	Type	$\lvert V\rvert$	$\lvert D\rvert$	$\lvert T\rvert$	τ
mS-C1-100	C1	100	2	2	10
mS-C2-100	C2	100	2	2	10
mS-R-100	R	100	2	3	10
mS-RC-100	RC	100	2	3	10
mGH-C1-150	C1	150	2	3	10
mGH-C2-150	C2	150	2	3	10
mGH-R1-150	R1	150	2	3	10
mGH-R2-150	R2	150	2	3	10
mGH-RC-150	RC	150	2	3	10
mGH-C1-200	C1	200	3	2	10
mGH-C2-200	C2	200	3	2	10
mGH-R1-200	R1	200	3	2	10
mGH-R2-200	R2	200	3	2	10
mGH-RC-200	RC	200	3	2	10
mGH-C1-300	C1	300	3	2	10
mGH-C2-300	C2	300	3	2	10
mGH-R1-300	R1	300	3	2	10
mGH-R2-300	R2	300	3	2	10
mGH-RC-300	RC	300	3	2	10
mGH-C1-400	C1	400	4	2	10
mGH-C2-400	C2	400	4	2	10
mGH-R1-400	R1	400	4	2	10
mGH-R2-400	R2	400	4	2	10
mGH-RC-400	RC	400	4	2	10

The workload of subdistrict $V_{k,i}^t$ can be calculated as

$$W_{k,i}^t = 2d_{k,i}^t + \beta_2 \sqrt{n_{k,i}^t A_{k,i}^t} + \tau n_{k,i}^t, \tag{7}$$

where $d_{k,i}^t$ is the driving time between the assigned depot of district V_k^t and the customer of subdistrict $V_{k,i}^t$ closest to the assigned depot, $n_{k,i}^t$ is the number of subdistrict $V_{k,i}^t$ and τ is the service time of a customer. With respect to the working time limit, the routing cost $F_{rout\ k,i}^t(x)$ in subdistrict $V_{k,i}^t$ can be calculated as

$$F_{rout\ k,i}^t(x) = \begin{cases} W_{k,i}^t & \text{if } W_{k,i}^t \le h, \\ h + \theta(W_{k,i}^t - h) & \text{otherwise,} \end{cases} \tag{8}$$

where θ is the overtime rate.

3. Adaptive large neighbourhood search metaheuristic

Since our problem includes the design of the district and subdistrict on the territory and should be solved for multi-periods and multi-depots, to efficiently solve the problem, we have devised an Adaptive Large Neighbourhood Search (ALNS) metaheuristic which dynamically modifies the solution. For simplicity, a district is assigned to the depot whose location is the closest to the location of its centre. Ties are broken arbitrarily.

The ALNS metaheuristic framework was proposed by Ropke and Pisinger [32] and is rooted in the work of Shaw [35,36]. It has already been successfully applied to several problems (e.g. Ropke and Pisinger [32], Pisinger and Ropke [30], Goel and Gruhn [16], Laporte et al. [22], Lei et al. [23], Hong [19], Ribeiro and Laporte [33], Lei et al. [25], Demir et al. [11], Masson et al. [27], Salazar-Aguilar et al. [34] and Azi et al. [2]). This metaheuristic must be fine tuned to each application. In this section we describe its application to the proposed problem.

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 using one of the three insertion operators, where q is randomly selected in the interval $[\lceil 0.1 n_{bou} \rceil, \lceil 0.2 n_{bou} \rceil]$ as in Laporte et al. [22], and n_{bou} is the total number of the boundary basic units of current solution at the first period. A probabilistic mechanism is applied to select the removal and insertion operators and these remove-insert operators are combined to efficiently explore the solution space. Contiguity is always maintained.

3.1. Definition of the basic units

As noted by Haugland et al. [18], the notion of contiguity is difficult to define in discontinuous settings. In order to operationalise this concept for the design of district and subdistrict, it is necessary to

Table 4
Computational results on the modified Solomon and Gehring & Homberger instances.

Instance	Construction heuristic					ALNS metaheuristic					Seconds	Imp (%)	Dev (%)
	$\overline{m}$	$\overline{F}_{comp}$	$\overline{F}_{dissim}$	$\overline{F}_{equ}$	F	$\overline{m}$	$\overline{F}_{comp}$	$\overline{F}_{dissim}$	$\overline{F}_{equ}$	F			
mS-C1-100	2	0.2159	0	0.2173	4.8663	2	0.2384	0	0.0074	4.4916	39.05	7.70	1.32
mS-C2-100	2	0.2381	0.0388	0.2612	5.0762	2	0.2546	0	0.0227	4.5547	41.81	10.27	13.17
mS-R-100	2	0.2918	0.0087	0.4344	8.2046	2	0.2812	0.0133	0.0129	6.9220	57.98	15.63	19.94
mS-RC-100	2	0.2561	0	0.3492	7.8161	2	0.2665	0.0017	0.0158	6.8519	55.08	12.34	1.36
100-average	2	0.2505	0.0119	0.3155	6.4908	2	0.2602	0.0038	0.0147	5.7050	48.48	11.49	8.95
mGH-C1-150	5	0.1563	0.0385	0.4455	16.9210	3	0.3445	0.0066	0.0525	10.2108	157.17	39.66	7.68
mGH-C2-150	5	0.1781	0.0210	0.4810	17.0404	3	0.2929	0.0144	0.0314	10.0160	150.52	41.22	13.12
mGH-R1-150	5	0.1897	0.0109	0.6753	17.6277	3	0.3007	0.0110	0.0275	10.0175	147.80	43.17	7.41
mGH-R2-150	4.67	0.1895	0.0153	0.9041	17.3267	3	0.2608	0.0013	0.0157	9.8337	146.31	43.25	20.95
mGH-RC-150	4	0.1668	0.0403	0.1124	12.9586	4	0.1811	0.0206	0.0738	12.8267	162.2969	1.01	19.97
150-average	4.73	0.1761	0.0252	0.5237	16.3749	3.20	0.2760	0.0108	0.0402	10.5809	152.82	33.66	13.83
mGH-C1-200	6	0.1441	0.0445	0.5125	13.4022	3	0.3890	0.0086	0.0184	6.8319	148.88	49.02	13.62
mGH-C2-200	4	0.1899	0.0265	0.1351	8.7028	4	0.2021	0.0193	0.0374	8.5176	99.25	2.13	0.36
mGH-R1-200	4	0.2001	0.0291	0.0940	8.6464	4	0.2304	0.0134	0.0344	8.5563	123.61	1.04	9.53
mGH-R2-200	5	0.1914	0.0330	0.4987	11.4462	4	0.2139	0.0087	0.0318	8.5088	117.84	25.66	14.60
mGH-RC-200	5	0.1516	0.0109	0.8417	12.0085	5	0.1603	0.0025	0.8000	11.9256	141.5313	00.69	2.19
200-average	4.80	0.1754	0.0288	0.4164	10.8412	4	0.2391	0.0105	0.1844	8.8680	126.22	15.71	8.06
mGH-C1-300	11	0.1102	0.0499	0.8170	23.9543	7	0.2599	0.0084	0.8571	16.2507	257.28	32.16	11.02
mGH-C2-300	11	0.1192	0.0148	0.6854	23.6388	6	0.4093	0.0101	0.6667	14.1722	348.66	40.05	10.69
mGH-R1-300	12.50	0.1072	0.0261	0.9145	27.0956	7	0.2980	0.0052	1.1429	16.8920	413.14	37.66	9.21
mGH-R2-300	10	0.1282	0.0311	0.5214	21.3613	6	0.4061	0.0202	0.0839	13.0204	472.27	39.05	9.78
mGH-RC-300	14	0.1010	0.0434	1.0599	30.4085	7.50	0.4515	0.0196	1.5686	19.0792	437.36	37.26	10.60
300-average	11.70	0.1132	0.0331	0.7996	25.2917	6.70	0.3649	0.0127	0.8638	15.8829	385.7406	37.23	10.26
mGH-C1-400	21.50	0.0890	0.0316	1.2768	45.7946	9	0.1739	0.0217	0.6667	19.7247	667.23	56.93	9.01
mGH-C2-400	12.50	0.1101	0.0579	0.9568	27.2497	5	0.4089	0.0215	0.4000	11.6607	590.86	57.21	17.70
mGH-R1-400	12.50	0.1202	0.0281	0.6301	26.5568	8	0.3242	0.0167	0.2500	17.1817	543.06	35.30	6.74
mGH-R2-400	12.50	0.1149	0.0224	0.6738	26.6220	7	0.2792	0.0128	0.0533	14.6907	554.98	44.82	13.24
mGH-RC-400	17.50	0.0975	0.0492	0.9949	37.2833	7	0.3229	0.0150	0.8571	16.3901	494.22	56.04	10.97
400-average	15.30	0.1063	0.0378	0.9065	32.7013	7.20	0.3018	0.0175	0.4454	15.9296	570.07	50.06	11.53
Average	**7.94**	**0.1607**	**0.0280**	**0.6039**	**18.8337**	**4.73**	**0.2896**	**0.0114**	**0.3220**	**11.6303**	**265.34**	**30.39**	**10.59**

embed the customers in basic units which partition the territory under study. Given a set of customer locations, we construct the basic units as follows. Assume that the location of customer i in the instance territory is described by the coordinates (x_i, y_i). The territory is defined by $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$, where $x_{min} = \min_{i \in V}\{x_i\}$, $x_{max} = \max_{i \in V}\{x_i\}$, $y_{min} = \min_{i \in V}\{y_i\}$ and $y_{max} = \max_{i \in V}\{y_i\}$. 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 $d = \min\left\{ \min_{i,j \in V}\{|x_i - x_j|\}, \min_{i,j \in V}\{|y_i - y_j|\} \right\}$. If $d = 0$, d is set to the smallest measurement unit of the coordinate (e.g., 1). Any unit with no customer is merged with the nearest unit having at least one customer.

3.2. Initial solution

To generate a good feasible initial solution, we have devised the following construction heuristic. The heuristic first randomly selects a basic unit as the seed unit to initialise the first subdistrict of the first district. The heuristic gradually extends this subdistrict by adjoining adjacent units to it. The workload of the extended subdistrict does not exceed the duration limit h. If adjoining an adjacent unit would cause the workload to exceed h, this unit is not included in the subdistrict but serves as a seed unit for a new subdistrict. If the number of subdistricts of the extended district exceeds the number of working days in a week, a new district is then generated. Any subdistrict with only one basic unit is eliminated and merged with the adjacent subdistrict so as to yield the lowest increased workload.

3.3. Removal and insertion operators

We have designed three removal operators and three insertion operators.

3.3.1. Removal operators

1. *Great remove (GR)*: This operator focuses on the districts with larger areas. The operator first sorts the districts in non-increasing order of the values of their area. If the number m^1 of districts of the current solution at the first period is larger than q, the operator randomly removes a boundary basic unit from each of the first q districts without disconnecting them, and stops. Otherwise, the operator randomly removes a boundary basic unit from the first district without disconnecting it and sorts the districts again after the removal, until q boundary basic units have been removed.

2. *Large remove (LR)*: This operator concentrates on those districts with larger numbers of customers. The operator first sorts the districts in non-increasing order of their number of customers. If

Table 5
Performance of the insertion and removal operators measured by the mean value of the final weight.

Removal			Insertion		
GR	LR	RR	LCI	FI	SI
6.5354	6.7798	6.3427	6.2715	7.2471	5.9819

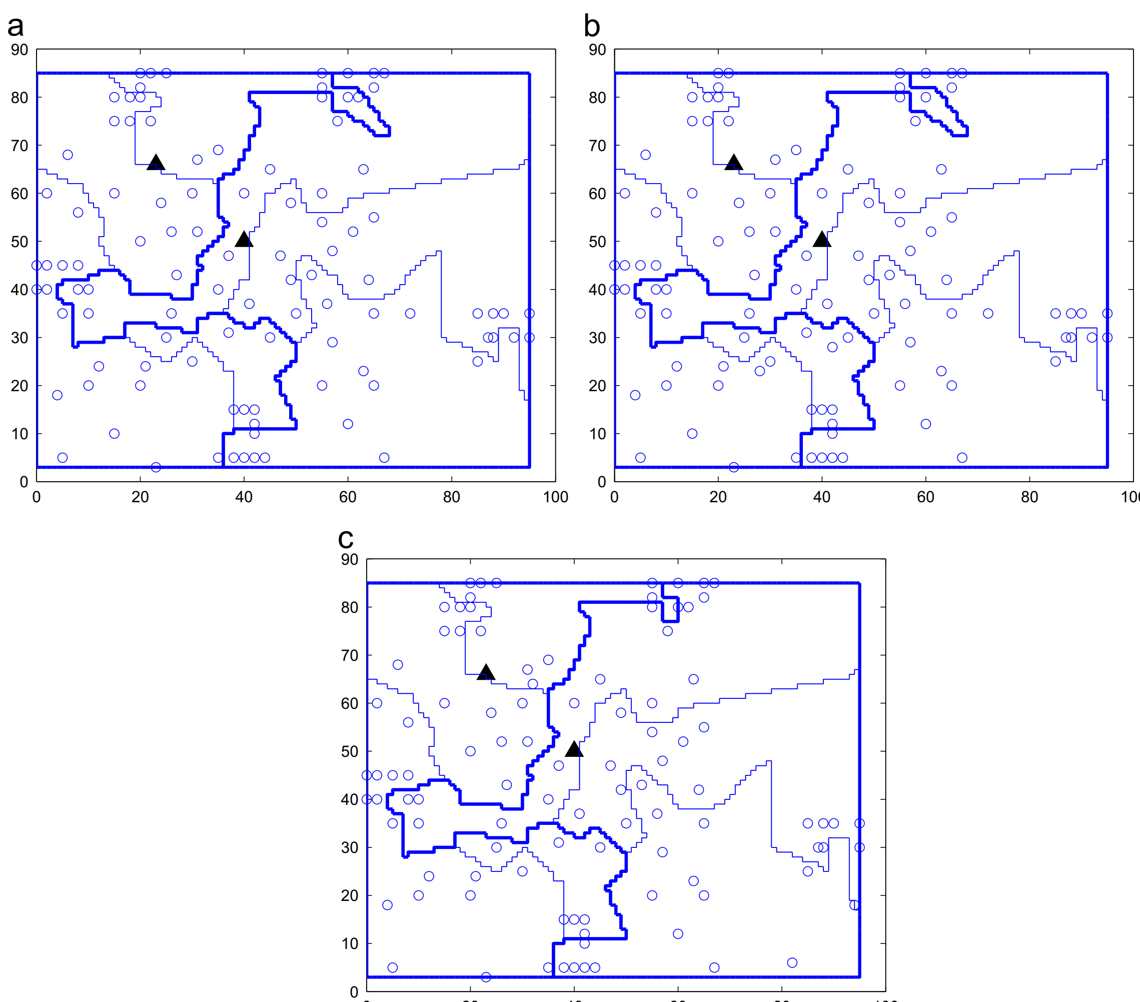

Fig. 2. Solution of instance mS-RC-100. (a) First period, (b) second period, and (c) third period.

$m^1 > q$, the operator randomly removes a boundary basic unit from each of the first q districts without disconnecting them and stops. Otherwise, a boundary basic unit is randomly removed from the first district without disconnecting it, then the districts are sorted again. If q boundary basic units have been removed, the operator stops.

3. *Random remove (RR)*: The operator randomly selects q boundary basic units and removes them from their subdistrict without disconnecting it.

3.3.2. Insertion operators

1. *Lowest cost insertion (LCI)*: This operator reinserts the removed units into the adjacent subdistricts with the lowest increase in the workload. The operator first selects one unit from the removed units, and chooses an adjacent subdistrict with the lowest increase in the workload value as the best subdistrict, respecting the duration limit h and the tabu tenure described in Section 3.5. Then, the operator reinserts the unit into the best subdistrict. The operator stops if q removed units have been reinserted.

2. *Smallest insertion (SI)*: This operator inserts the removed units into the adjacent subdistricts with the smallest area. The operator first sorts the adjacent subdistricts in increasing order of the values of their area, and selects the subdistrict with smallest area to reinsert the units. The operator reinserts the removed units until the q units have been reinserted. The reinsertion respects the duration limit and the tabu tenure.

3. *Fewest insertion (FI)*: This operator reinserts the q removed units into the adjacent subdistricts with the least customer number. The operator sorts the adjacent subdistricts in increasing order of the values of their customer number, and chooses the best subdistrict with the least customer number. The operator reinserts the removed units into the best subdistrict, respecting the duration limit and the tabu tenure.

3.4. Adaptive selection mechanism

The removal and insertion operators are each associated a weight which dynamically changes throughout the algorithm. At each iteration, the selection of a removal operator and of an insertion operator is based on a roulette-wheel selection principle. Given l operators with weights w_i, operator j is chosen with probability $w_j/\sum_{i=1}^{l} w_i$. The weight of each operator is updated every $\varrho(\varrho=50)$ iterations. Initially, all operators have a weight of 1. The weight w_{ij} of operator i at the j^{th} sequence of ϱ iterations is computed as

$$w_{i,j+1} = w_{ij}(1-r)+r\frac{\sigma_{ij}}{\varepsilon_{ij}}, \tag{9}$$

where ε_{ij} and σ_{ij} are the number of times operator i has been used in the jth sequence of ϱ iterations, and the score of operator i in the jth sequence of ϱ iterations, respectively, and $r \in [0,1]$ is an arbitrary parameter equal to 0.1 in our implementation.

The score σ_{ij} can be computed by the following rules: the score is set to 0 at the start of the jth sequence of ϱ iterations; it is increased by $\sigma_{ij}^1=30$ if the removal-insertion operation results in a new global best solution; it is increased by $\sigma_{ij}^2=10$ if the removal-insertion operation results in a new solution improved on the current solution; it is increased by $\sigma_{ij}^3=6$ if the removal-insertion operation results in a new solution which does not improve upon the current solution but is accepted.

3.5. Tabu tenure for avoiding remove-insert cycling

We use a tabu tenure to avoid cycling when reinserting the removed units, as in Bozkaya et al. [5]. The moves that reinsert the basic unit i back into its previous subdistrict is declared tabu for ϕ iterations. As recommended by Haugland et al. [18], ϕ is set equal to the number of districts at the first period in the initial solution. As is common in tabu search, any tabu move may be still implemented if it yields a new incumbent solution.

3.6. Territories design over different periods

As mentioned in Section 1, we are interested in designing districts and subdistricts over several periods. However, not all customers change over time. There is therefore no need to run the full ALNS metaheuristic at each period. Instead, we can only adjust the subdistricts with the changes of the customers comparing to the previous period.

The heuristic first selects the basic units for which the customers of the current period have changed compared to those of the previous period. These units are considered for the reinsertion into the adjacent subdistricts respecting the duration limit. For each reinserted unit, the heuristic chooses the best adjacent subdistrict with the largest drop in the workload. If there is no adjacent subdistrict yielding a smaller workload, the unit then remains in its subdistrict of the previous period.

Note that the number of districts is not fixed over different periods. First, the number of districts may be reduced. Though not all customers change over times, the customers of a district may change along time periods and the number of subdistrict of a district is not fixed (noted in Section 2). If the number of customers of a district reduces to 0, all the basic units of the subdistricts of the district will be merged by its adjacent district and the number of districts goes down. Second, the number of districts can also be increased. Since the changed basic units are considered for the reinsertion into the adjacent subdistricts respecting the duration limit, if all the adjacent subdistricts and the original subdistrict will violate the duration limit for the reinsertion of the changed basic units (caused by the new entering customers) and the number of subdistricts is the maximum value at the same time, a new district will be generated and the number of districts will go up.

3.7. Acceptance and stopping criteria

We have used the record-to-record travel (RRT) algorithm introduced by Dueck [13] to define the acceptance criterion for a new solution. Consider f^* is the objective value of the best known solution, called a record. 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 $f_{x'} < f^*$, where δ is a small number called deviation. In our implementation we set $\delta = 0.1 f^*$.

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 2500 and 5000 respectively in our implementation.

3.8. Summary of the ALNS metaheuristic

Our ALNS implementation can be summarised as follows.

Step 1: Initialise 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 corresponding *best cost*, compute the *deviation*. Set the initial solution as the *record* and define it as the *current solution*.

Step 2: Choose a removal operator and an insertion operator using roulette-wheel selection principle based on the weights of the current iteration. Generate the first period part of a new solution from the current solution using the chosen removal and insertion operators without disconnecting the districts and subdistricts, and while respecting the tabu tenure. Then, use the territories design method over different periods.

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 *record*. 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 *best cost*, update the *record*, the *best cost* and the *deviation*.

Step 4: Update the score and the number of times used of the chosen operators. Update the weights of all removal and insertion operators and reset their score and number of times used for the next iterations, if necessary (every 50 iterations).

Step 5: If the stopping criterion is met, go to Step 6; otherwise go to Step 2.

Step 6: Output the *record* and the *best cost*.

4. Computational experiments

The algorithm described in Section 3 was coded using Matlab 7.0.4 and run on a PC with 3.1 GHz four processors and 2 GB RAM. We now describe the results of our computational experiments. Since no previous computational results are available for our problem, we have conducted tests aimed at evaluating the efficiency of the ALNS improvement heuristic compared with a construction heuristic, and we have conducted several sensitivity analyses.

4.1. Experimental design

We have generated test instances derived from those of Solomon [38] and those of Gehring and Homberger [15]. The coordinates of the customers at the first period are the same as in the Solomon instances and the Gehring & Homberger instances, while the demands and time windows are not used. The coordinates of the customers at the other periods are generated by the rule described in Section 1, where a proportion η (taken randomly in the interval (0, 0.05)) of all customers of previous period exit from the territory, and a proportion ψ (taken randomly in the interval (0, 0.10)) of new customers add into the territory. We have also used the coordinate of the depot in the original instances as the coordinate of the first depot in the test instances, and the coordinates of the other depots are randomly generated among $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$.

In the Solomon instances, 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. In the Gehring and Homberger instances, the coordinates of RC1 and RC2 are the same, but those of C1 and C2 are not identical and neither are those of R1 and R2. We consider five types of instances: R1, R2, C1, C2, and RC.

We respectively consider the 100, 150, 200, 300 and 400 customers in the tests, where the customers of the instances with 150 customers are chosen from the first 150 customers of the instances with 200 customers. The service times of customers are equal to 10, and h is set to 250. The detailed information of the modified Solomon and Gehring & Homberger instances tested is shown in Table 3.

Table 6
Computational results on the modified Solomon and Gehring & Homberger instances with different value of α_{comp}.

Instance	$\alpha_{comp}=1$					$\alpha_{comp}=10$					$\alpha_{comp}=100$				
	$\overline{m}$	$\overline{F}_{comp}$	$\overline{F}_{dissim}$	$\overline{F}_{equ}$	F	$\overline{m}$	$\overline{F}_{comp}$	$\overline{F}_{dissim}$	$\overline{F}_{equ}$	F	$\overline{m}$	$\overline{F}_{comp}$	$\overline{F}_{dissim}$	$\overline{F}_{equ}$	F
mS-C1-100	2	0.2384	0	0.0074	4.4916	2	0.2017	0	0.0433	8.1209	2	0.1539	0	0.0991	34.9738
mS-C2-100	2	0.2546	0	0.0227	4.5547	2	0.1500	0.0095	0.0278	7.0747	2	0.1827	0.0095	0.2353	41.0203
mS-R-100	2	0.2812	0.0133	0.0129	6.9220	2	0.2254	0.0035	0.0112	12.8067	2	0.1893	0.0124	0.0393	62.9382
mS-RC-100	2	0.2665	0.0017	0.0158	6.8519	2	0.2027	0	0.0323	12.1773	2	0.1864	0.0143	0.0984	62.2703
100-average	2	0.2602	0.0038	0.0147	5.7050	2	0.1950	0.0033	0.02866	10.0449	2	0.1781	0.0091	0.1180	50.3007
mGH-C1-150	3	0.3445	0.0066	0.0525	10.2108	3	0.2379	0.0111	0.0368	16.2794	5	0.1618	0.0276	0.7216	65.7776
mGH-C2-150	3	0.2929	0.0144	0.0314	10.0160	4	0.1785	0.0566	0.2203	18.1846	4	0.1607	0.0171	0.1590	60.7238
mGH-R1-150	3	0.3007	0.0110	0.0275	10.0175	3	0.2129	0.0045	0.0735	15.6214	3	0.1916	0.0050	0.0287	66.5910
mGH-R2-150	3	0.2608	0.0013	0.0157	9.8337	3	0.2191	0.0085	0.0683	15.8046	3	0.1939	0.0049	0.2837	68.0313
mGH-RC-150	4	0.1811	0.0206	0.0738	12.8267	4	0.1923	0.0257	0.0739	18.0676	4	0.1796	0.0487	0.5544	67.6962
150-average	3.20	0.2760	0.0108	0.0402	10.5809	3.40	0.2081	0.0213	0.0946	16.7915	3.80	0.1775	0.0206	0.3495	65.7640
mGH-C1-200	3	0.3890	0.0086	0.0184	6.8319	4	0.2311	0.0295	0.0621	12.8046	6	0.1450	0.0496	0.5167	42.1289
mGH-C2-200	4	0.2021	0.0193	0.0374	8.5176	4	0.1970	0.0204	0.0749	12.1313	4	0.1679	0.0010	0.5059	42.5852
mGH-R1-200	4	0.2304	0.0134	0.0344	8.5563	4	0.2414	0.0162	0.1093	13.0793	5	0.1682	0.0109	0.2335	44.1381
mGH-R2-200	4	0.2139	0.0087	0.0318	8.5088	4	0.1981	0.0089	0.0543	12.0885	6	0.1484	0.0164	0.5212	42.7589
mGH-RC-200	5	0.1603	0.0025	0.8000	11.9256	4	0.2174	0.0141	0.5000	13.3759	5	0.1604	0.0192	0.3866	42.8982
200-average	4	0.2391	0.0105	0.1844	8.8680	4	0.2170	0.0178	0.1601	12.6959	5.20	0.1580	0.0194	0.4328	42.9019
mGH-C1-300	7	0.2597	0.0084	0.8571	16.2507	9	0.1409	0.0321	0.3912	21.6640	9	0.1339	0.0158	1.5234	47.8521
mGH-C2-300	6	0.4093	0.0101	0.6667	14.1722	7	0.1756	0.0365	1.1567	19.8976	9	0.1408	0.0402	0.9414	48.1217
mGH-R1-300	7	0.2980	0.0052	1.1429	16.8920	8	0.1689	0.0147	1.2500	21.9073	12	0.1171	0.0239	0.6090	48.6897
mGH-R2-300	6	0.4061	0.0202	0.0839	13.0204	6	0.2665	0.0215	0.0848	17.5436	10	0.1324	0.0350	0.6041	47.7526
mGH-RC-300	7.50	0.4515	0.0196	1.5686	19.0792	6	0.3796	0.0047	1.5692	22.7399	12	0.1042	0.0187	0.8467	46.5617
300-average	6.70	0.3649	0.0127	0.8638	15.8829	7.20	0.2263	0.0219	0.8904	20.7505	10.40	0.1257	0.0267	0.9049	47.7956
mGH-C1-400	9	0.1739	0.0217	0.6667	19.7247	10	0.2211	0.0263	2.1651	28.8042	14	0.1086	0.0157	1.7181	53.1814
mGH-C2-400	5	0.4089	0.0215	0.4000	11.6607	6	0.3674	0.0511	0.3685	20.1882	11	0.1082	0.0417	0.5431	44.8121
mGH-R1-400	8	0.3242	0.0167	0.2500	17.1817	8	0.2602	0.0133	0.5188	22.2674	12	0.1120	0.0200	0.6954	47.8323
mGH-R2-400	7	0.2792	0.0128	0.0533	14.6907	5	0.3336	0.0110	0.4456	17.5857	11	0.1144	0.0117	0.5407	45.9835
mGH-RC-400	7	0.3229	0.0150	0.8571	16.3901	10	0.1737	0.0219	0.8020	25.1217	13	0.0953	0.0379	0.9444	47.0262
400-average	7.20	0.3018	0.0175	0.4454	15.9296	7.80	0.2712	0.0247	0.8600	22.7934	12.20	0.1077	0.0254	0.8883	47.7671
Average	**4.73**	**0.2896**	**0.0114**	**0.3220**	**11.6303**	**5**	**0.2247**	**0.0184**	**0.4225**	**16.8890**	**6.92**	**0.1482**	**0.0207**	**0.5562**	**50.9310**

4.2. Computational results

Table 4 presents computational results for the modified Solomon and Gehring & Homberger instances with $\alpha_m = 1$, $\alpha_{comp} = 1$, $\alpha_{dissim} = 1$ and $\alpha_{equ} = 1$. Each instance was solved five times and the best results are presented in Table 4. The column "Construction heuristic" summarises the results obtained from the construction heuristic of Section 3.2. The column "ALNS metaheuristic" summarises the results obtained by applying the heuristic of Section 3. The column "$\overline{m}$" gives the average number of districts of the solutions over periods. The column "$\overline{F}_{comp}$" presents the average compactness measure cost of the subdistricts of the solutions over periods, computed by Eq. (2). The column "$\overline{F}_{dissim}$" shows the average dissimilarity measure cost of districts of the solutions over periods, computed by Eq. (3). The column "$\overline{F}_{equ}$" provides the average equity measure cost of salesmen profit of the solutions over periods, computed by Eq. (4). The column "F" is the total cost of the solutions, computed by Eq. (1). We also report the total CPU time in seconds in the "Seconds" column for "ALNS metaheuristic". The "Imp(%)"column shows the percentage improvement in "Total cost" obtained by "ALNS metaheuristic", compared with "Construction heuristic". The "Dev(%)"column describes the stability of "ALNS metaheuristic" (five runs), which is computed by the equations $\sum_{i=1}^{5}|F^i - \overline{F}|/5\overline{F}$ and $\overline{F} = \sum_{i=1}^{5} F^i/5$.

Table 4 clearly shows that the solutions of "ALNS metaheuristic" are much better than those of "Construction heuristic". The average improvement percentage is 30.39%. The average number of districts of the solutions of "ALNS metaheuristic" is less than that of "Construction heuristic", and so is the average compactness measure cost, the average dissimilarity measure cost, the average equity measure cost of salesmen profit and the total cost. The average CPU time for "ALNS metaheuristic" is 265.34 s. The average "Dev" value of "ALNS metaheuristic" is 10.59%. Fig. 2 presents the solution of the instance mS-RC-100. It indicates that the changes of customers do not cause the territory design to change at the second period, but yield a small change at the third period.

We have also evaluated the performance of each insertion and removal operators described in Section 3.3. Table 5 reports the weights associated with each operator. This weight is obtained as the mean value of final weights used to generate the results of Tables 4. The best removal operators are GR and LR, whereas the best insertion operators are LCI and FI.

4.3. Computational experiments with different parameters

We have run computational tests with different parameters by successively varying the multiplier α_{comp} of F^t_{comp} and the multiplier α_{equ} of F^t_{equ} in the objective function. Table 6 provides the solution values obtained with $\alpha_{comp} = 1$, $\alpha_{comp} = 10$ and $\alpha_{comp} = 100$, leaving the other multipliers unchanged. As expected, when the value of α_{comp} becomes larger, the average value of $\overline{F}_{comp}$ becomes smaller. In

Table 7
Computational results on the modified Solomon and Gehring & Homberger instances with different value of α_{equ}.

Instance	$\alpha_{equ} = 1$					$\alpha_{equ} = 10$					$\alpha_{equ} = 100$				
	$\overline{m}$	$\overline{F}_{comp}$	$\overline{F}_{dissim}$	$\overline{F}_{equ}$	F	$\overline{m}$	$\overline{F}_{comp}$	$\overline{F}_{dissim}$	$\overline{F}_{equ}$	F	$\overline{m}$	$\overline{F}_{comp}$	$\overline{F}_{dissim}$	$\overline{F}_{equ}$	F
mS-C1-100	2	0.2384	0	0.0074	4.4916	2	0.3039	0	0.0008	4.6246	2	0.2832	0	0.0001	4.5919
mS-C2-100	2	0.2546	0	0.0227	4.5547	2	0.2438	0	0.0073	4.6331	2	0.3504	0.0139	0.0000	4.7349
mS-R-100	2	0.2812	0.0133	0.0129	6.9220	2	0.2990	0	0.0025	6.9729	2	0.3099	0	0.0009	7.1987
mS-RC-100	2	0.2665	0.0017	0.0158	6.8519	2	0.2812	0.0074	0.0060	7.0455	2	0.2673	0.0181	0.0004	6.9802
100-average	2	0.2602	0.0038	0.0147	5.7050	2	0.2820	0.0018	0.0042	5.8190	2	0.3027	0.0080	0.0004	5.8764
mGH-C1-150	3	0.3445	0.0066	0.0525	10.2108	4	0.2453	0.0241	0.0180	13.3493	4	0.2500	0.0218	0.0062	14.6634
mGH-C2-150	3	0.2929	0.0144	0.0314	10.0160	4	0.3291	0.0133	0.0122	13.3942	4	0.2218	0.0339	0.0100	15.7586
mGH-R1-150	3	0.3007	0.0110	0.0275	10.0175	3	0.2488	0.0154	0.0136	10.2012	4	0.2641	0.0102	0.0103	15.8982
mGH-R2-150	3	0.2608	0.0013	0.0157	9.8337	4	0.2385	0.0108	0.0223	13.4160	3	0.2791	0.0057	0.0037	10.9621
mGH-RC-150	4	0.1811	0.0206	0.0738	12.8267	4	0.3019	0.0165	0.0200	13.5560	4	0.2825	0.0200	0.0118	16.4358
150-average	3.20	0.2760	0.0108	0.0402	10.5809	3.80	0.2727	0.0160	0.0172	12.7834	3.80	0.2595	0.0183	0.0084	14.7436
mGH-C1-200	3	0.3890	0.0086	0.0184	6.8319	7	0.2501	0.0175	0.2857	20.2494	6	0.2671	0.0313	0.0171	16.0166
mGH-C2-200	4	0.2021	0.0193	0.0374	8.5176	4	0.2841	0.0072	0.0100	8.7837	4	0.2199	0.0101	0.0084	10.1337
mGH-R1-200	4	0.2304	0.0134	0.0344	8.5563	4	0.2811	0.0043	0.0129	8.8282	4	0.2429	0.0115	0.0068	9.8764
mGH-R2-200	4	0.2139	0.0087	0.0318	8.5088	4	0.2475	0.0039	0.0171	8.8444	4	0.2604	0.0087	0.0044	9.4105
mGH-RC-200	5	0.1603	0.0025	0.8000	11.9256	5	0.2728	0.0028	0.0193	10.9371	5	0.2080	0.0086	0.0100	12.4347
200-average	4	0.2391	0.0105	0.1844	8.8680	4.80	0.2671	0.0071	0.0690	11.5285	4.60	0.2397	0.0140	0.0093	11.5744
mGH-C1-300	7	0.2597	0.0084	0.8571	16.2507	10	0.3314	0.0310	0.2000	24.7249	10	0.2195	0.0063	0.0300	26.4401
mGH-C2-300	6	0.4093	0.0101	0.6667	14.1722	8	0.1621	0.0191	0.5000	26.3624	10	0.2702	0.0355	0.0316	26.9226
mGH-R1-300	7	0.2980	0.0052	1.1429	16.8920	13	0.3793	0.0179	0.1568	29.9313	12	0.2347	0.0169	0.0432	33.1427
mGH-R2-300	6	0.4061	0.0202	0.0839	13.0204	9	0.2976	0.0129	0.0373	19.3670	9	0.2303	0.01225	0.0370	25.8789
mGH-RC-300	7.50	0.4515	0.0196	1.5686	19.0792	10	0.2443	0.0228	0.2057	24.6491	12	0.1808	0.0180	0.0393	32.2523
300-average	6.70	0.3649	0.0127	0.8638	15.8829	10	0.2830	0.02076	0.2200	25.0069	10.60	0.2271	0.0178	0.0362	28.9273
mGH-C1-400	9	0.1739	0.0217	0.6667	19.7247	12	0.3754	0.0141	0.1673	28.1242	14	0.2234	0.0127	0.5714	142.7579
mGH-C2-400	5	0.4089	0.0215	0.4000	11.6607	9	0.2519	0.0217	0.4444	27.4360	10	0.1711	0.0247	0.0268	25.7437
mGH-R1-400	8	0.3242	0.0167	0.2500	17.1817	10	0.2871	0.0196	0.0339	21.2919	11	0.1884	0.0120	0.0311	28.6195
mGH-R2-400	7	0.2792	0.0128	0.0533	14.6907	9	0.2384	0.0131	0.0316	19.1354	10	0.2128	0.0088	0.0282	26.0860
mGH-RC-400	7	0.3229	0.0150	0.8571	16.3901	11	0.1931	0.02791	0.3636	29.7148	11	0.1879	0.0127	0.0366	29.7175
400-average	7.20	0.3018	0.0175	0.4454	15.9296	10.20	0.2692	0.0193	0.2082	25.1405	11.20	0.1968	0.0142	0.1388	50.5849
Average	**4.73**	**0.2896**	**0.0114**	**0.3220**	**11.6303**	**6.33**	**0.2745**	**0.0135**	**0.1079**	**16.4822**	**6.63**	**0.2427**	**0.0147**	**0.0402**	**23.0274**

contrast, the average values of $\overline{m}$, $\overline{F}_{dissim}$, $\overline{F}_{equ}$ and F become larger. Table 7 provides the comparison of the computational solutions with $\alpha_{equ}=1$, $\alpha_{equ}=10$ and $\alpha_{equ}=100$, leaving the other multipliers unchanged. Like in Table 6, increasing α_{equ} means that the average value of $\overline{F}_{equ}$ becomes smaller, and the average values of $\overline{m}$, $\overline{F}_{dissim}$ and F become larger.

5. Conclusions

We have introduced, modeled and solved a combined multiple traveling salesman and districting problem, where the customers of territory are dynamically changed with the multi-periods and several possible depots. We have solved the problem with several criteria, including subdistrict compactness, a dissimilarity measure of the district partition and an equity measure of salesmen's profits. Travel costs were approximated by the Beardwood–Halton–Hammersley formula. We have developed an adaptive large neighbourhood search metaheuristic for the problem. Modified Solomon and Gehring & Homberger instances were used to assess the quality of the proposed algorithm. 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 No. 39682-10, by the Chinese National Natural Science Foundation under Grant Nos. 71201170 and 71371181, and by the Hunan Provincial Natural Science Foundation of China under Grant No. 13JJ4010, and by the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20124307120024. This support is gratefully acknowledged. Thanks are due to the referees for their valuable comments.

References

[1] Applegate DL, Bixby RE, Chvátal V, Cook WJ. The travelling salesman problem: a computational study. Princeton University Press; 2006.
[2] Azi N, Gendreau M, Potvin J-Y. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. Comput Oper Res 2014;41(1):167–73.
[3] Beardwood J, Halton JH, Hammersley JM. The shortest path through many points. In: Mathematical proceedings of the Cambridge Philosophical Society, vol. 55, no. 4. Cambridge University Press; 1959. p. 299–327.
[4] Blais M, Lapierre SD, Laporte G. Solving a home care districting problem in an urban setting. J Oper Res Soc 2003;54(11):1141–7.
[5] Bozkaya B, Erkut E, Laporte G. A tabu search heuristic and adaptive memory procedure for political districting. Eur J Oper Res 2003;144(1):12–26.
[6] Bozkaya B, Erkut E, Haight D, Laporte G. Designing new electoral districts for the city of Edmonton. Interfaces 2011;41(6):534–47.
[7] Carlsson JG, Devulapalli R. Dividing a territory among several facilities. INFORMS J Comput 2013;25(4):730–42.
[8] Carlsson JG. Dividing a territory among several vehicles. INFORMS J Comput 2012;24(4):565–77.
[9] Carlsson JG, Delage E. Robust partitioning for stochastic multivehicle routing. Oper Res 2013;61(3):727–44.
[10] D'Amico SJ, Wang SJ, Batta R, Rump CM. A simulated annealing approach to police district design. Comput Oper Res 2002;29(6):667–84.
[11] Demir E, Bektaş T, Laporte G. The bi-objective Pollution-Routing Problem. Eur J Oper Res 2014;232(3):464–76.
[12] Drexl A, Haase K. Fast approximation methods for sales force deployment. Manage Sci 1999;45(10):1307–23.
[13] Dueck G. New optimization heuristics: the great deluge algorithm and the record-to-record travel. J Comput Phys 1993;104(1):86–92.
[14] Ferland JA, Guénette G. Decision support system for the school districting problem. Oper Res 1990;38(1):15–21.
[15] Gehring H, Homberger J. 1999. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In: Miettinen K, Mäkelä M, Toivanen J, editors. 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; 1999. p. 57–64.
[16] Goel A, Gruhn V. A general vehicle routing problem. Eur J Oper Res 2008;191(3):650–60.
[17] Groër C, Golden BL, Wasil EA. The consistent vehicle routing problem. Manuf Serv Operations Manage 2009;11(4):630–43.
[18] Haugland D, Ho SC, Laporte G. Designing delivery districts for the vehicle routing problem with stochastic demands. Eur J Oper Res 2007;180(3):997–1010.
[19] Hong L. An improved LNS algorithm for real-time vehicle routing problem with time windows. Comput Oper Res 2012;39(2):151–63.
[20] Kalcsics J, Nickel S, Schröder M. Towards a unified territorial design approach - applications, algorithms and GIS integration. TOP 2005;13(1):1–56.
[21] King DM, Jacobson SH, Sewell EC, TamCho WK. Geo-Graphs: an efficient model for enforcing contiguity and hole constraints in planar graph partitioning. Oper Res 2012;60(5):1213–28.
[22] Laporte G, Musmanno R, Vocaturo F. An adaptive large neighbourhood search heuristic for the capacitated arc routing problem with stochastic demands. Transp Sci 2010;44(1):125–35.
[23] Lei H, Laporte G, Guo B. The capacitated vehicle routing problem with stochastic demands and time windows. Comput Oper Res 2011;38(12):1775–83.
[24] Lei H, Laporte G, Guo B. Districting for routing with stochastic customers. EURO J Transp Logist 2012;1(1):67–85.
[25] Lei H, Laporte G, Guo B. The vehicle routing problem with stochastic demands and split deliveries. INFOR 2012;50(2):59–71.
[26] López-Pérez JF, Ríos-Mercado RZ. Embotelladoras ARCA uses operations research to improve territory design plans. Interfaces 2013;43(3):209–20.
[27] Masson R, Lehuédé F, Péton O. The Dial-A-Ride problem with transfers. Comput Oper Res 2014;41(1):12–23.
[28] Mehrotra A, Johnson EL, Nemhauser GL. An optimization based heuristic for political districting. Manage Sci 1998;44(8):1100–14.
[29] Novaes AGN, Souza de Cursi JE, da Silva ACL, Souza JC. Solving continuous location-districting problems with Voronoi diagrams. Comput Oper Res 2009;36(1):40–59.
[30] Pisinger D, Ropke S. A general heuristic for vehicle routing problems. Comput Oper Res 2007;34(8):2403–35.
[31] Ríos-Mercado RZ, Fernández E. A reactive GRASP for a commercial territory design problem with multiple balancing requirements. Comput Oper Res 2009;36(3):755–76.
[32] Ropke S, Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp Sci 2006;40(4):455–72.
[33] Ribeiro GM, Laporte G. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Comput Oper Res 2012;39(3):728–35.
[34] Salazar-Aguilar MA, Langevin A, Laporte G. The multi-district team orienteering problem. Comput Oper Res 2014;41(1):76–82.
[35] Shaw P. A new local search algorithm providing high quality solutions to vehicle routing problems. Technical report, Department of Computer Science, University of Strathclyde, Glasgow; 1997.
[36] Shaw P. Using constraint programming and local search methods to solve vehicle routing problems. In: Principles and practice of constraint programming – CP98, lecture notes in computer science, vol. 1520. Berlin: Springer-Verlag; 1998. p. 417–31.
[37] Skiera B, Albers S. COSTA: contribution optimizing sales territory alignment. Mark Sci 1998;17(3):196–213.
[38] Solomon MM. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper Res 1987;35(2):254–65.