

基于 Voronoi 邻近的物流车辆路径快速优化算法

涂伟¹, 方志祥^{1,2*}, 李清泉^{1,2,3}, 鲁仕维¹

- (1. 武汉大学 测绘遥感信息工程国家重点实验室, 武汉 430079;
2. 武汉大学 时空数据智能获取技术与应用教育部工程研究中心, 武汉 430079;
3. 深圳大学 时空信息智能感知与服务深圳市重点实验室, 深圳 518061)

摘要: 现代物流业需要快速高效并智能化制定物流运输方案。传统路径优化方法适合处理中小规模的车辆路径问题, 计算时间较长, 方案质量较低, 故需发展短时间内能提供高质量路径方案的启发式算法。针对大规模物流车辆路径优化, 本文提出了一种 Voronoi 邻近的快速优化方法。该方法先创建初始解, 而后进行迭代优化。初始解创建利用 Voronoi 邻近关系, 顾及车辆容量约束, 自底向上进行客户点空间聚类, 将问题降维; 采用最廉价插入算法安排聚类内部路径, 生成性质良好的初始解。迭代优化在客户点 Voronoi 邻近内进行有效的局部搜索, 利用模拟退火机制接受较差解, 从而跳出局部最优, 不断提高解的质量。本文利用模拟生成的北京市大规模车辆路径问题进行实验, 结果表明: 本文算法能够在 4500s 内优化客户点高达 12 000 个物流车辆路径问题, 计算时间较短, 解的质量优良, 算法性能稳定。本文与其他算法比较, 能在较短时间内提供高质量车辆路径方案, 适用于大规模物流车辆路径的优化。

关键词: 物流; 车辆路径问题; 空间聚类; Voronoi; 模拟退火

DOI: 10.3724/SP.J.1047.2012.00781

1 引言

随着交通基础设施的完善, 物流行业得到了极大发展, 并逐渐向信息化、智能化方向发展^[1]。智能物流一方面集成 RFID 技术、空间信息技术和云计算, 管理物流信息, 提升业务效率^[2-4]; 另一方面, 智能化地进行物流设施选址、库存控制, 车辆路径设计和物流监控等决策, 促进物流系统的协调运转^[1-2]。车辆路径优化是物流智能化重点, 其目标是设计最佳的车辆路线进行货物配送, 满足客户需求, 从而减少物流费用, 降低物流成本, 提高客户满意度。城市中的快递配送、邮政投递、垃圾收集等物流活动每天服务于成千上万个客户点, 配送任务重, 时间紧, 需要发展高效的启发式算法在短时间内提供高质量的车辆路径方案^[5-7]。

物流车辆路径优化通常被建模为车辆路径问题 (Vehicle Routing Problem, VRP), 是运筹学、物流科学和交通运输工程等领域的研究热点。当

前有多种算法进行车辆路径优化, 如精确算法、经典启发式算法和现代启发式算法^[5]。精确算法如整数规划、行变换能够获得准确解, 但求解速度较慢, 只能处理少量的客户点。启发式算法获得近似最优解, 能够处理较大规模的车辆路径问题, 是进行大规模车辆路径优化的唯一途径。经典启发式算法如节约启发式、扫描启发式、先聚类后创建路径和先创建路径后聚类等, 利用简单规则快速生成解决方案, 但方案质量通常较差。现代启发式算法分为局部搜索的算法 (如模拟退火、禁忌搜索、变邻域搜索) 和群体搜索的算法 (如遗传算法和蚁群算法), 通常能够提供较高质量的车辆路径方案^[5]。

车辆路径问题求解难度随问题规模急剧增长^[5]。关键在于生成高质量的初始解和有效的搜索策略。前者创建性质良好的初始方案, 奠定后续优化基础; 后者引导局部搜索方向, 搜索更准确的邻域, 提高搜索效率。针对大规模车辆路径问题, 本文提出了基于 Voronoi 邻近的快速优化方法。

收稿日期: 2012-11-02; **修回日期:** 2012-12-01.

基金项目: 国家“863”计划项目 (2012AA12A403-4); 国家自然科学基金重点项目 (40830530; 41231171); 国家自然科学基金项目 (40971233)。

作者简介: 涂伟 (1984-), 男, 博士研究生, 研究方向为时空建模、分析和优化。E-mail: tuweiwhu@gmail.com

通讯作者: 方志祥 (1977-), 博士, 副教授, 主要从事时空分析研究。E-mail: zxfang@whu.edu.cn

该方法以“先聚类后安排路径”的思路,生成性质良好的初始解决方案;然后在 k 阶 Voronoi 邻域内进行路径优化,搜索有效的局部邻域,不断改善高路径方案的质量,从而在短时间内提供高质量的车辆路径方案,减少物流运输费用,提高运输效率。

2 车辆路径快速优化算法原理

2.1 车辆路径问题

从图论的角度,车辆路径问题定义如下: $G=(V, E)$, $V=\{0, 1, 2, \dots, n\}$ 为节点集, 0 表示仓库, $C=\{1, 2, \dots, n\}$ 表示客户点, n 为客户点的数量, 每个客户有一个非负需求 q_i ; E 为边集, $E=\{(i, j) \mid i, j \in V, i \neq j\}$, 每条边 e_{ij} 有一个非负费用 c_{ij} , 表示从客户点 i 到 j 的旅行费用, 通常采用欧式距离 d_{ij} 或旅行时间 t_{ij} 度量。 M 辆车从仓库出发, 配送货物, 满足客户的需求, 最终回到仓库。 每辆车最大容量为 Q , 最长行驶距离为 L 。 路径优化设计总长度或总服务时间最小的车辆路径方案, 遍历客户点并满足用户需求。 这些车辆路径须满足 4 个条件:

- (1) 车辆路径始于仓库, 并最终返回仓库。
- (2) 客户点必须且只能有由一辆车来服务。
- (3) 每条路径上客户点需求之和不超过 Q 。
- (4) 每条路径的长度不超过 L 。

2.2 算法基本原理

Voronoi 图根据生长点将空间划分为多个 Voronoi 多边形^[8]。 每个 Voronoi 多边形的边界由多条 Voronoi 边组成。 共享 Voronoi 边的生长点具有空间邻近关系, 互为空间近邻^[9]。 两个生长点之间路径跨过 Voronoi 边的最少数量为 Voronoi 距离^[10], 如图 1 所示, 点 v_i 和 v_j 之间的 Voronoi 距离为 3。 基于 Voronoi 距离, 可以将其拓展至 K 环 Voronoi 邻近^[10]。 图 1 中点状背景 Voronoi 多边形所在生长点是 v_i 的 1 阶 Voronoi 邻近。

Voronoi 空间邻近关系不仅表达了几何邻近关系, 还蕴含了拓扑邻近关系^[9]。 车辆路径方案中的顺次访问客户点通常也具有空间邻近关系。 因此, 本文以客户点的 Voronoi 邻近关系, 提出了一种局部搜索快速优化算法 (Voronoi Proximity-based Local Search algorithm, VPLS)。 该方法先创建初始路径方案, 然后迭代改善方案质量。 鉴于聚类方

法能够提供高质量的路径方案^[11-12], 本文利用一阶 Voronoi 邻近关系, 自下向上将客户点聚类, 进行聚类内部的车辆路径生成, 创建空间结构紧凑的初始路径方案; 在 k 阶 Voronoi 邻近内进行局部搜索, 利用模拟退火机制控制搜索走向, 迭代改进方案质量, 最终提供高质量的路径方案。

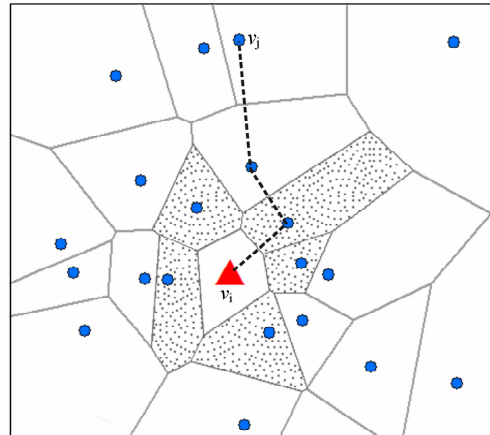


图 1 k 环 Voronoi 邻近

Fig. 1 The k -ring Voronoi neighbors

3 基于 Voronoi 邻近的局部搜索算法

3.1 基于 Voronoi 邻近的客户聚类

空间聚类方法通常可分为划分法、层次法、密度法和网格方法^[13], 一般并不考虑聚类对象的属性, 对聚类大小也不作限制。 文献[10]中聚类范围较大, 生成初始路径较为分散。 本文方法在 1 环 Voronoi 邻近内聚类, 路径结构更加紧密。 聚类过程自底向上进行, 并考虑车辆容量的约束, 要求聚类的客户需求之和不超过 Q , 从而每个聚类内部只有一条车辆路径, 具体如下:

(1) 创建客户点集 C 的 Voronoi 图, 填充每个客户点的 1 阶 Voronoi 邻近列表。

(2) 清空队列 R 。 遍历所有的 1 阶 Voronoi 邻近客户点对, 将其压入 R 中。

(3) 将列表 R 按客户点对欧式距离排序。

(4) 取出 R 最前端的客户点对 $i-j$, 进行聚类合并。 如果 i, j 在同一聚类中, 不进行合并操作。 否则, 判断能否合并聚类 C_i, C_j 。 如果 $Q(C_i) + Q(C_j) < Q$, 则合并聚类 C_i, C_j , 并更新相关客户点所在聚类, 其中, C_i, C_j 为分别为客户点 i, j 所在聚

类标识。

(5) 反复执行步骤(4),直到列表 R 为空。

3.2 初始方案生成

经过 Voronoi 空间聚类后,所有客户点依照邻近关系分割为多个聚类,每个聚类的客户需求和不超 Q 。针对每个聚类,初始方案生成采用最廉价插入算法^[14]生成聚类内部的路径。该方法规则简单,生成路径速度快。该方法先随机选择聚类内部的一个节点作为种子,生成初始路径。然后随机选择同一聚类内部未插入的节点 v 作为候选节点,按照式(1)计算插入 v 到路段 $i-j$ 中的费用 ΔS 。选择 ΔS 最少的位置执行插入操作,直到聚类内部所有节点均插入完毕。

$$\Delta S = d_{iv} + d_{vj} - \lambda d_{ij} \quad (1)$$

其中, λ 是预设随机参数, $\lambda \in [0.5, 1.5]$ 。

遍历所有聚类,生成车辆路径,从而创建初始路径方案。

3.3 路径方案的改善

路径改善阶段采用局部搜索方法不断改进方案质量。局部搜索过程在邻域内寻找更好的解。传统方法遍历当前解的所有邻域,效率较低^[14]。本文算法将局部搜索限定在客户点的 k 阶 Voronoi 邻近内,只搜索在最优解中最可能出现的边,从而有效减少搜索范围,提高效率。

本文使用 4 个常见局部搜索算子:重定位算子,交换算子,尾部交换算子,3-Opt 算子^[7]。重定位算子将一个节点插入到新的位置。交换算子互换两个节点的位置。尾部交换算子互换两条路径的尾部片段。3-Opt 算子删除一条路径中的两条边,添加两条新边,实现路径片段的反转。其中,重定位,交换操作,尾部交换在路径内或路径间均可操作,3-Opt 算子只在路径内操作。局部搜索过程随机产生一个算子和一个客户点,在客户点的 k 阶 Voronoi 邻近内搜索候选操作节点,并判断是否接受邻域解。

是否执行局部搜索操作标准如下:如果邻域解优于当前解,直接执行局部搜索;否则,依照模拟退火机制接受局部搜索操作。模拟退火机制模仿固体冷却过程,概率性接受局部搜索中的较差邻域解,从而跳出局部最优的限制^[15-16]。概率接受性准则首先产生 $(0, 1)$ 内的随机数,然后判断是否接受

较差解^[10]。利用不断下降的温度控制较差解的接受度,最终温度趋近于 0,从而收敛于高质量的解。本文采用线性梯度下降方法逐步降低温度。和文献^[10]不同,本文采用最低温度判断迭代是否终止,从而保证收敛于相同质量水平的解。

4 算法模拟实验与对比分析

本文采用模拟生成的大规模物流配送数据集进行实验,验证算法结果,测试有效性与稳定性;与其他算法进行比较,评价解的质量和算法效率。

4.1 实验数据

本文以北京市为例,利用导航数据中的兴趣点(POI)(共 74 041 个),按照文献^[17]中的方法,生成模拟的大规模物流车辆路径问题。每个实例从 POI 中随机选择一定数量的点,生成空间位置真实的客户点,每个客户需求 q_i 为 $[1, 100]$ 间的随机整数。仓库设定于北京市东部的通州物流园区内,拥有满足需求的运输车辆。每辆车最大容量为 $Q=2000$,行驶最大长度 $L=200\text{km}$ 。一生成 6 个大规模的配送实例进行实验,各个实例的具体参数如表 1 所示。

表 1 大规模物流车辆路径问题实例参数

Tab. 1 The parameters of large scale logistic vehicle routing problem instances

名称	N	TD	Q	$L(\text{km})$
BJ1	2000	102 702	2000	200
BJ2	4000	205 099	2000	200
BJ3	6000	307 781	2000	200
BJ4	8000	404 210	2000	200
BJ5	10 000	505 831	2000	200
BJ6	12 000	611 574	2000	200

注: N : 客户点数量; TD : 总需求量; Q : 车辆容量; L : 车辆最大行驶距离

4.2 VPLS 算法结果

利用 C++ 语言实现本文算法 VPLS。实验时运行于 Window 7 系统上的个人 PC 上, CPU 为 Intel(R) Core(TM) I3-2100 CPU @ 3.10GHz 3.10GHz, 内存为 8GB, 程序运行时只使用单个 CPU。经多次试验, 设定 VPLS 参数如下: 模拟退火策略起始温度 $T_0 = 0.5 \times$ 平均 Voronoi 邻近长度, 温度

下降速度 $c=0.992$, 最小温度 $T_{\min}=0.5$, 局部搜索中的 Voronoi 邻近阶数 $k=2$ 。算法运行 10 次, 记录计算时间、路径长度和路径数量。

以实例 BJ3 为例, 图 2 记录了路径长度和的迭代过程。随着局部搜索的进行, 路径长度逐步改善, 并在 1300s 左右趋于稳定。图 3 为 BJ3 的车辆路径安排方案(图中为了绘图清晰, 未绘制路径开始和结束时的路径段), 共 155 条路径, 绝大部分的路径段均存在空间邻近关系, 部分较长路径段在返回仓库的途中顺便访问个别客户点。

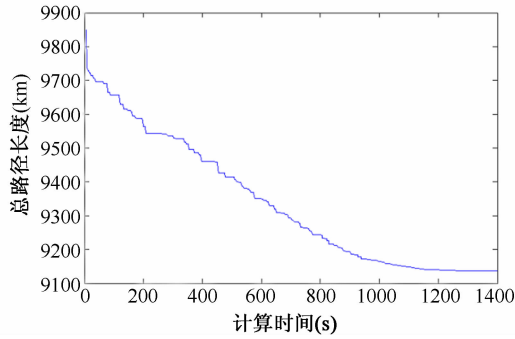


图 2 本文算法运行迭代曲线

Fig. 2 The iteration curve of the algorithm adopted in this study

表 2 报告了本文算法 VPLS 的计算结果, 包括初始方案和最终优化方案的相关明细。如表 2 所示, 基于 Voronoi 聚类的创建算法能在 8.96s 给出

12 000 个客户点的初始方案, 在 4453.23s 内优化客户点高达 12 000 个的车辆路径问题。相对于初始解, 优化过程能够减少路径数量, 6 个问题实例的路径数量和从 1127.8 下降到 1076.5, 使用的车辆更少; 路径长度和从 67 296.588km 下降到 62 306.641km, 降低约 7.41%, 路径方案质量改善明显。因此, 本文算法具有较高的效率, 能够快速提供车辆路径方案。相对于平均解, 最好解的路径数量更少, 为 1074 条; 路径长度更短, 相对于平均值的波动幅度在 $-0.38\% \sim -0.69\%$ 之间, 幅度比较小, 表明算法 VPLS 性能稳定。

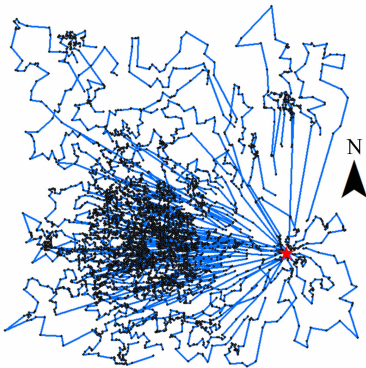


图 3 BJ3 的车辆路径方案

Fig. 3 The solution to the large scale logistic vehicle routing problem (BJ3)

表 2 本文算法 VPLS 的计算结果

Tab. 2 The result of VPLS algorithm for the large scale vehicle routing problems in Beijing

名称	客户点数量	初始方案			最终方案						
		RN_{avg}	S_{avg}	T_{avg}	RN_{avg}	S_{avg}	T_{avg}	RN_{best}	S_{best}	T_{best}	$gap(S_{best})$
BJ1	2000	55.2	3887.243	1.34	52.2	3461.545	111.63	52	3448.560	105.39	-0.38
BJ2	4000	110.7	6815.730	2.76	103.4	6306.724	351.21	103	6284.239	361.33	-0.36
BJ3	6000	162.0	9870.962	4.26	155.3	9 089.466	729.98	155	9044.635	726.34	-0.49
BJ4	8000	214.3	12 595.652	5.71	203.5	11 740.125	1508.90	203	11 659.627	1439.57	-0.69
BJ5	10000	266.7	15 775.322	7.33	254.6	14 446.197	2 646.60	254	14 358.172	2683.28	-0.61
BJ6	12000	319.0	18 351.679	8.96	307.5	17 262.584	4 453.23	307	17 178.470	4275.86	-0.49
路径数量和		1127.8			1076.5			1074			
路径长度和			67 296.588			62 306.641			61 973.703		
计算时间和				30.36			9821.56			9591.77	

注: RN: 路径数量; S: 路径长度和(km); T: 计算时间(s); $gap(S_{best}) = (S_{best} - S_{avg}) * 100 / S_{avg}$ 。

4.3 算法比较

利用 TransCAD 中的 Vehicle Routing 模块^[18], 文献[19]中的 VRPEJ 算法, 计算北京市大规模物流车辆路径问题, 并与 VPLS 结果进行比较。TransCAD 是一种专业的交通信息系统软件, 实验中采用推荐参数求解北京市大规模车辆路径

问题算例。VRPEJ 算法是第一个大规模车辆路径问题启发式算法^[20]的改进型, 利用局部搜索改善解的质量, 采用 k 近邻减少局部搜索范围, 并附加抽取-插入操作, 利用 C++语言实现。计算前述问题实例时采用其推荐参数: $k=30$, $max_tries=5$, $intensity=30$, $I=50\ 000$ 。TransCAD, VRPEJ 均

运行于和 VPLS 相同的 PC,均运行 10 次,记录最好解,并进行比较。

算法比较有 2 个方面:解的质量和计算效率。解的质量通常以相对最好解的百分比 gap 表示^[10],按照式(2)计算:

$$gap = (S - BEST) \times 100 / BEST \quad (2)$$

其中, S 为 TransCAD, VRPEJ, VPLS 各自的最好解, $BEST$ 为 TransCAD, VRPEJ, VPLS 中的最小值。 gap 为不小于 0 的值, gap 值越小,算法结果越接近最好解,解的质量越高。计算效率以计算时间衡量,时间越短,效率越高。

表 3 为 3 种算法的运行结果。从解的质量上看,TransCAD 解的质量最差,和最好解平均相差

12.94%,总共使用了 1077 车次。VRPEJ 解的质量次之,和最好解平均相差 0.64%,路径数量和为 1074。VPLS 解的质量最好,6 个大规模车辆路径问题实例均获得最好解,路径数量和 1074,使用车辆数量较少。因此,本文算法 VPLS 能够提供高质量的路径方案。

从计算时间上看,TransCAD 计算时间最长,6 个算例的总计算时间为 305865.76s。VRPEJ 次之,总共需要 182106.5s。VPLS 最短,总共需要 9591.77s,约为 VRPEJ 的 5.2%,TransCAD 的 3.1%。因此,本文算法能够在较短时间内提供高质量的车辆路径方案。

表 3 三种算法计算结果比较
Tab.3 Comparison of results of three heuristic algorithms

名称	TransCAD				VRPEJ				VPLS				BEST
	RN	Sbest	gap	T	RN	Sbest	gap	T	RN	Sbest	gap	T	
BJ1	52	3969.247	15.10	5950.20	52	3475.129	0.77	5415.93	52	3448.560	0.00	105.39	3448.560
BJ2	103	6984.889	11.15	24 112.04	103	6291.960	0.12	12 362.87	103	6284.239	0.00	361.33	6284.239
BJ3	155	10 282.808	13.69	36 697.06	155	9070.485	0.29	19 971.19	155	9044.635	0.00	726.34	9044.635
BJ4	204	13 044.297	11.88	57 547.15	203	11 757.008	0.84	33 963.23	203	11 659.627	0.00	1439.57	11 659.627
BJ5	255	16 220.932	12.97	73 278.59	254	14 514.475	1.09	52 628.33	254	14 358.172	0.00	2683.28	14 358.172
BJ6	308	19 372.656	12.77	108 280.72	307	17 304.653	0.73	57 764.94	307	17 178.470	0.00	4275.86	17 178.470
路径数量和	1077				1074				1074				
gap 平均值			12.93				0.64				0.00		
计算时间和				305 865.76				182 106.5				9591.77	

注:RN: 路径数量; S: 路径长度和(km); T: 计算时间(s); gap: 相对最好解的偏差百分比

5 结论

传统的车辆路径优化方法无法在较短时间内提供高质量的车辆路径安排方案。利用客户点间的空间邻近关系能够有效减少车辆路径优化的复杂度。鉴于 Voronoi 邻近特性,本文提出了一种适合大规模物流车辆路径优化的启发式算法。该算法首先利用 Voronoi 邻近进行空间聚类,将问题进行降维,然后利用最廉价插入算法生成路径,创建空间结构良好的初始解。在客户点的 k 阶 Voronoi 邻域内进行有效的局部搜索,利用模拟退火机制接受搜索过程中的较差解,迭代改善解的质量。北京市的大规模车辆路径问题实验表明:本文算法能在较短时间内提供高质量的车辆路径方案,算法性能稳定,适合于大规模物流车辆路径优化。本文算法

可集成在物流车辆监控与优化系统中,提供车辆路径设计服务,从而提高物流效率,降低物流费用,实现物流智能化。

参考文献:

[1] 高自友,孙会君. 现代物流与交通运输系统[M]. 北京: 人民交通出版社, 2003.

[2] Santos L, Coutinho-Rodrigues J, Antunes C H. A web spatial decision support system for vehicle routing using Google Maps[J]. Decision Support Systems, 2011. 51 (1): 1-9.

[3] 肖桂荣,聂乔,吴升. 面向物流的空间信息 Web 服务集成研究[J]. 地球信息科学学报, 2011, 13(2): 630-636.

[4] 戚铭尧. 面向物流的空间信息服务及其关键技术研究[D]. 北京:中国科学院遥感应用研究所,2006.

[5] Laporte G. Fifty years of vehicle routing[J]. Transpor-

- tation Science, 2009, 43(4): 408–416.
- [6] 李清泉, 张金亭, 黄经南. 一个物流配送优化算法[J]. 武汉大学学报·信息科学版, 2003, 28(1): 9–13.
- [7] Kytöjoki J, Nuortio T, Bräysy O, *et al.* An efficient variable neighborhood search heuristic for very large scale vehicle routing problems [J]. Computers & Operations Research, 2007, 34(9): 2743–2757.
- [8] Okabe A, Boots B, Sugihara K, *et al.* Spatial tessellations: Concepts and applications of Voronoi Diagrams [M]. Wiley, 2000.
- [9] Chen J, Li C, Li Z, *et al.* A Voronoi-based 9-intersection model for spatial relations [J]. International Journal of Geographical Information Science, 2001, 15(3): 201–220.
- [10] Fang Z X, Tu W, Li Q Q, *et al.* A Voronoi neighborhood-based search heuristic for distance/capacity constrained very large vehicle routing problems [J]. International Journal of Geographical Information Science, 2012, DOI: 10.1080/13658816.2012.707319.
- [11] 梅新, 崔伟宏, 高飞, 等. 基于空间聚类的物流配送决策研究[J]. 武汉大学学报·信息科学版, 2008, 33(4): 371–375.
- [12] Barreto S, Ferreira C, Paixão J, Santos B S. Using clustering analysis in a capacitated location-routing problem [J]. European Journal of Operational Research, 2007, 179(3): 968–977.
- [13] Han J W, Kamber M and Pei J. Data mining: Concepts and techniques [M]. The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers, 2011(3rd edition).
- [14] Toth P and Daniele V. The vehicle routing problem [M]. Society for Industrial and Applied Mathematics, 2002.
- [15] Kirkpatrick S, Gelatt Jr, C D, Vecchi M P. Optimization by simulated annealing [J]. Science, 1983, 220(4598): 671–680.
- [16] 张波, 叶家玮, 胡郁葱. 模拟退火算法在路径优化问题中的应用[J]. 中国公路学报, 2004, 17(1): 83–85.
- [17] Zachariadis E E and Kiranoudis C T. A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem [J]. Computers & Operations Research, 2010, 37(12): 2089–2105.
- [18] TransCAD. <http://www.caliper.com/tcovu.htm>
- [19] Groër C, Golden B, Wasil E. A library of local search heuristics for the vehicle routing problem [J]. Mathematical Programming Computation, 2010, 2(2): 79–101.
- [20] Li F, Golden B, Wasil E. Very large-scale vehicle routing: New test problems, algorithms, and results [J]. Computers & Operations Research, 2005, 32(5): 1165–1179.

A Fast Algorithm for Large Scale Vehicle Routing Optimization Based on Voronoi Neighbors

TU Wei¹, FANG Zhixiang^{1,2}, LI Qingquan^{1,2,3} and LU Shiwei¹

(1. State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; 2. Engineering Research Center for Spatio-Temporal Data Smart Acquisition and Application, Ministry of Education, Wuhan 430079, China; 3. Shenzhen Key Laboratory of Spatial-temporal Smart Sensing and Service, Shenzhen University, Shenzhen 518061, China)

Abstract: Logistic requires designing delivery plan intelligently and quickly. Traditional vehicle routing optimal algorithms can only solve vehicle routing problem no more than 2000 customers, and cost much computing efforts. This paper proposes a fast heuristic algorithm for large scale logistic vehicle routing optimization based on Voronoi neighbors. This algorithm creates an initial solution and improves it iteratively. The creation makes use of the Voronoi neighbors to cluster customers into groups from bottom to up, considering the vehicle capacity constraint. The route in each group is generated by the cheapest insertion algorithm. The improvement employs local search in k-order Voronoi neighbors to search the

promising neighbor solution. The simulated annealing criterion is adopted to accept some bad solutions, escaping from local minima. The computational test has been done with a synthetic large scale vehicle routing problem dataset in Beijing. The result indicates the proposed algorithm could solve vehicle routing problem up to 12 000 customers in 4500 seconds. It costs few computational efforts, and has a quite stable performance. Comparison to other heuristics shows the proposed algorithm could provide high quality solution in a short time. The conclusion indicates the proposed algorithm is suitable for large scale logistic vehicle routing optimization.

Key words: logistic; vehicle routing problem; spatial cluster; Voronoi; simulated annealing