

求解大规模 CVRP 问题的快速贪婪算法

饶卫振, 金 淳

(大连理工大学 系统工程研究所, 辽宁 大连 116024)

摘要: 为求解大规模具有能力约束的车辆路径问题 (Capacitated Vehicle Routing Problem, CVRP), 提出了一种快速改进贪婪算法 CVRP-IMGR。基于贪婪算法思想设计了求解 CVRP 问题的贪婪算法 CVRP-GR, 在此基础上进一步采用 $K-d$ tree 法和 Held Karp 模型改进了 CVRP-GR 的求解速度和求解质量, 从而得到 CVRP-IMGR。CVRP-IMGR 的复杂度可以达到 $O(n \log n)$, 能够快速求解大规模 (顾客数量大于 500) CVRP 问题。为验证 CVRP-IMGR 的有效性, 分别采用 CVRP-GR、CVRP-IMGR 和经典构建型算法 Savings 求解了当前 24 个最大规模的 CVRP 算例, 结果表明: CVRP-IMGR 的求解速度远快于复杂度为 $O(n^2 \log n)$ 的 CVRP-GR 和 Savings; CVRP-IMGR 对所有算例的求解质量优于 CVRP-GR, 并且对 18 个算例的求解质量优于 Savings。

关键词: 能力约束车辆路径问题; 贪婪算法; $K-d$ 树; Held Karp 模型

中图分类号: F502 **文献标识码:** A **文章编号:** 1004-6062(2014)02-0045-10

0 引言

车辆路径问题 (Vehicle Routing Problem, VRP) 是典型的组合优化难题^[1], 该问题根据所考虑实际因素的不同可分为不同类型, 其中考虑车辆能力约束的车辆路径问题 (Capacitated Vehicle Routing Problem, CVRP) 是最为常见的形式之一^[2,3]。由于 CVRP 属于 NP-Hard 问题, 所以目前启发式算法是求解 CVRP 问题的主要方法^[2]。启发式方法又可分为构建型启发式方法和改进型启发式方法^[1]。构建型算法是从最初的空解开始, 迭代的添加解成分到解中, 直到完全构建解。典型的构建型算法有 Savings 算法^[4]、插入算法^[5]、生成树法^[6]等。改进型算法是以构建型算法的解为起点, 耗费更多的时间反复迭代逐步改进解的质量。常见的改进型算法有 $k-opt$ 算法^[7]、禁忌搜索法^[8]、遗传算法^[9]、分散搜索算法^[10]、标号算法^[11]、蚁群算法^[12]、量子算法^[13,14]等。

然而到目前为止, 非常鲜见复杂度低于 $O(n^2)$ 的有效算法^[1,2]。因此随着 CVRP 问题规模的扩大, 很多启发式算法也难以满足实际应用的需求, 目前文献中常用的 CVRP 标准算例的节点数在 1200 以下^[15], 但实际应用中的规模可以达到上万甚至几万个节点^[16], 因此有学者提出了节点数达 20000 的标准算例^[16]。另外, 在实际物流配送过程中经常有顾客需求变动、交通堵塞等信息导致当前方案不可行或者求解质量下降^[1], 因此要求算法能够快速的处理最新的变动信息。然而很多启发式算法中经常涉及到不同参数的调节, 有些算法的参数数量甚至可达数十个^[17~20], 这无疑增加了算法的复杂性和使用的困难性, 导致这类算法在实际应用中存在较大限制。换言之, 在实践中算法的快速性、简易性、灵活

性和求解质量有着同等重要的意义^[2], 比如求解 CVRP 问题的经典 Savings 算法, 由于其具有简明的规则和满意的求解质量等特点, 自 1964 年提出以来在实践中一直被广泛的应用^[1]。因此, 开发面向大规模 CVRP 问题的快速、简易、质量满意的算法十分必要。

贪婪算法 (Greedy Heuristic, GR) 由于其简洁有效的特点, 已经被应用到众多的组合优化问题中, 如旅行商问题^[21]、匹配问题^[22]、调度问题^[23]等, 但目前求解 CVRP 问题方面还缺乏成熟的应用。对此, 本文提出一种能够快速求解大规模 CVRP 问题的 CVRP-IMGR 算法, 该算法框架包括求解 CVRP 的标准贪婪算法 CVRP-GR、基于 $K-d$ tree 的加速策略、基于 Held Karp 模型的改进质量策略三部分。CVRP-IMGR 的复杂度仅为 $O(n \log n)$, 并且求解质量比当前求解 CVRP 的经典构建型算法 Savings 更优。

1 CVRP 问题的描述及模型

CVRP 问题: 存在图 $G = (V, E)$, 其中节点集合 $V = \{0, 1, 2, \dots, n\}$ 代表 1 个配送中心和 n 个顾客, 边集合 $E = \{(i, j) \mid 0 \leq i \neq j \leq n\}$ 代表任意两节点形成的边, 边长用 d_{ij} 表示。设在配送中心 0 处有足够多的载量同为 Q 的车辆, n 个的需求量分别为 q_1, q_2, \dots, q_n , 并假设: (1) 所有 d_{ij} 已知; (2) $\max(q_i) \leq Q$ ($1 \leq i \leq n$)。请问: 如何安排 m (m 可以是变量或指定数量, $1 \leq m \leq n$) 辆车满足这 n 个顾客的需求, 且使所有车辆行走的总路径长度最短? 并且安排车辆路径时需满足以下条件: (1) 每辆车均以配送中心 0 作为起点和终点; (2) 每辆车均不允许超过最大载量 Q ; (3) 每个顾客只能由 1 辆车

收稿日期: 2011-11-28 修回日期: 2012-08-08

基金项目: 国家自然科学基金重大课题资助项目 (70890080, 70890083); 教育部博士点基金资助项目 (20100041110024)

作者简介: 饶卫振 (1981—), 男, 江西省丰城市人, 博士研究生, 主要从事物流系统优化、算法分析研究。

服务 1 次; (4) 每辆车行驶的距离不超过 L (有些 CVRP 问题不考虑该约束)。该问题的数学规划模型如下所示^[1]:

目标函数:

$$\min \sum_{k=1}^m \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ijk} \quad (1)$$

约束条件:

$$\sum_{i=1}^n \sum_{j=1}^n q_i x_{ijk} \leq Q (1 \leq k \leq m) \quad (2)$$

$$\sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ijk} \leq L (1 \leq k \leq m) \quad (3)$$

$$\sum_{j=1}^n \sum_{k=1}^m x_{ijk} = 1 (1 \leq i \leq n) \quad (4)$$

$$\sum_{i=1}^n \sum_{k=1}^m x_{ijk} = 1 (1 \leq j \leq n) \quad (5)$$

$$\sum_{j=1}^n \sum_{k=1}^m x_{0jk} = \sum_{i=1}^n \sum_{k=1}^m x_{i0k} = m (1 \leq i, j \leq n) \quad (6)$$

$\sum_{i,j \in S \times S} x_{ijk} \leq |S| - 1, S \subset V_k$ 且 $\neq \varphi, V_k$ 表示第 k 辆车服务的顾客集合 ($1 \leq k \leq m$) (7)

$$x_{ijk} = \begin{cases} 1 & \text{车辆 } k \text{ 经过弧 } (i, j) \\ 0 & \text{其他} \end{cases} (0 \leq i, j \leq n, 1 \leq k \leq m) \quad (8)$$

式 (1) 为目标函数, 表示所有 m 辆车的行车总路径长度最小; 式 (2) 为每辆车的载量约束; 式 (3) 为每辆车行驶距离约束; 式 (4)、(5) 表示每个客户恰好仅被 1 辆车服务; 式 (6) 约束了所有车辆的起终点都在配送中心; 式 (7) 表示每辆车行驶的路径轨迹恰好为一个简单圈。通过该整数模型式 (8) 可知, 如果顾客数量为 $n=99$, 使用的车辆数量为 $m=10$ 时, 该模型的变量个数就达到 $(n+1) \times (n+1) \times m = 100000$, 这无疑是一个非常难求解的整数规划问题。

以上模型一般是精确算法 (如分枝定界法) 使用的模型, 虽然在很多研究求解 CVRP 启发式算法文献中也使用该模型, 但该模型其实不利于对启发式算法执行细节的理解, 也掩盖了 CVRP 的属于组合优化问题的本质特征。为便于阐述本文提出的启发式算法, 我们对以上 CVRP 模型重新表述如下。

目标函数:

$$\min L_{total} = \sum_{k=1}^m L_k \quad (9)$$

约束条件:

$$\sum_{i \in V_k} q_i \leq Q (1 \leq k \leq m) \quad (10)$$

$$L_k \leq L (1 \leq k \leq m) \quad (11)$$

$$V_1 \cup V_2 \cdots \cup V_m = V \setminus \{0\} \quad (12)$$

$$n_1 + n_2 + \cdots + n_m = n \quad (13)$$

其中,

$$\begin{cases} L_k = d_{0v_1^k} + \sum_{i=1}^{n_k-1} d_{v_i^k v_{i+1}^k} + d_{v_{n_k}^k 0} \\ V_k = \{v_1^k, v_2^k, \dots, v_{n_k}^k\} \end{cases} (1 \leq k \leq m)$$

V_k 代表第 k 辆车服务的顾客集合, 且 $|V_k| = n_k$ 表示第 k 辆车服务的顾客数量。

约束条件式 (10) 和 (11) 表示每辆车的车辆的载量约束和行驶距离约束, 约束条件式 (12) 和 (13) 表示所有的顾客均被服务, 且每个顾客只被其中 1 辆车服务 1 次。

由上述模型可知, CVRP 模型的求解关键问题是在既有约束条件下, 如何将 n 个顾客划分为 m 个子集合然后分配给对应的 m 辆车, 并确定每辆车服务顾客的先后次序。应该注意的是, 当没有载量和行驶距离约束且仅有 1 辆配送车辆时 (即 $Q = +\infty, L = +\infty, m = 1$), CVRP 等价于规模为 $n+1$ 的 TSP 问题。

2 算法设计的基础及分析

2.1 贪婪算法思想及复杂度分析

贪婪算法 (Greedy, GR) 的思想是以“贪婪”的方式构建组合优化问题的可行解, 并且在构建的过程中不违反可行解的特点^[24]。现在以求解 TSP 问题为例, 阐述 GR 的求解思想并分析 GR 的复杂度。TSP 问题可以简单描述为: 假设 n 个点中任意 2 点间的距离已知, 确定一个通过 n 个点的最小长度的简单圈 (Hamilton 回路)。GR 求解 TSP 的方式是将任意两节点构成的边按边长 d_{ij} 升序排列 (共有 $C_n^2 = n(n-1)/2$ 条边), 从最短边开始依次添加合法边至当前路径中, 直到添加完 n 条边形成一个 Hamilton 回路为止^[21]。显然, 合法边不违反 TSP 可行解的特点。TSP 可行解的特点是每个点均和 2 条边连接, 且仅存在 1 个通过所有 n 个点的简单圈。因此 GR 求解 TSP 时判断一条边是否为合法边的准则是: (1) 添加该边后不会使任何点连接边的数量大于 2; (2) 该边也不会使图中产生边数小于 n 的圈。

通过上述过程可知: GR 要对所有的 $n(n-1)/2$ 条边进行排序, 然后逐条判断出 n 条符合添加条件的边, 所以 GR 的复杂度可由两部分组成, 即排序复杂度 $Sort()$ 和判断复杂度 $Judge()$ 。因复杂度由最坏情况下的计算量 (将一次判断、赋值、加、减、乘、除运算视为一次计算量) 确定, 所以要判断 GR 的复杂度就要分别计算排序和判断计算量。由于当前最快的排序算法复杂度为 $O(n \log n)$, 那么排序计算量 $Sort(x)$ 为:

$$\begin{aligned} Sort(x) &= x \log x \Rightarrow Sort(n(n-1)/2) \\ &= \frac{n(n-1)}{2} \log \frac{n(n-1)}{2} \\ &= \frac{n^2 - n}{2} [\log(n^2 - n) - 1] \end{aligned} \quad (14)$$

因为计算复杂度要在最坏情况下分析, 也就是说, 要考虑 $Judge()$ 的最坏情况, 即判断所有的 $n(n-1)/2$ 条边是否符合 2 条判断准则后, 才能得到 n 条合格的添加边, 那么 $Judge(x)$ 为:

$$\begin{aligned} Judge(x) &= 2x \Rightarrow Judge(n(n-1)/2) \\ &= 2 \times \frac{n(n-1)}{2} = n^2 - n \end{aligned} \quad (15)$$

所以 GR 的总计算量可由式 (14) 和 (15) 的加和计算

得到:

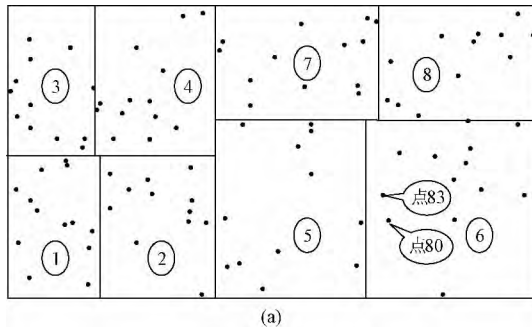
$$\begin{aligned} f(n) &= \frac{n^2 - n}{2} [\log(n^2 - n) - 1] + n^2 - n \\ &= \frac{n^2 - n}{2} [\log(n^2 - n) + 1] \leq \frac{n^2}{2} \log n^2 = n^2 \log n \end{aligned} \quad (16)$$

根据复杂度为计算量多项式的最高次方确定的理论,由式(16)可知 GR 的复杂度为 $O(n^2 \log n)$ 。

2.2 K-d tree 方法

K-d tree 方法(K-dimensional binary search tree ($K \geq 1$))即 K 维空间二叉搜寻树,其主要原理是将包含 n 个点所在 K 维空间的最小区域进行分割,然后将分割后的区域视为二叉树的节点,从而将整个区域用二叉树表示^[25]。这种表示方法能够快速的确定某个点所在区域,提高搜寻效率。

如图 1 所示,将在二维空间(平面 $K=2$)中的 100 个点所在区域进行分割,并采用 2-d tree (此时 K 等于 2)表示。图 1(a) 表示 100 个点所在的平面区域被分割为 8 部分,每



部分包含 12 或 13 个点;图 1(b) 表示该分割结构的 2-d tree。该 2-d tree 包含两种节点,一种表示可继续分割的区域节点(用方框表示),另一种是不可继续分割区域节点(用椭圆表示)。该树可以用于快速近似搜索节点的最邻近点,如在图 1(a) 中要寻找节点 80 的最邻近节点 83,如果没有建立该二叉搜寻树,则必须比较节点 80 与其他所有 99 个节点的距离后才能确定其最邻近节点。借助该二叉搜寻树,可通过两阶段寻找。阶段一:通过点 80 的横坐标值,首先可以确定其属于“方框 3”区域(直接排除“方框 2”包含的所有点),然后根据点 80 的纵坐标值可以确定点 80 更确切的位置属于“方框 6”,最后再根据点 80 的横坐标值,可以定位点 80 的位置为“椭圆 6”(定位结束);阶段二:找到在“椭圆 6”中其他 11 个点与点 80 最邻近的点 83,以此点近似为点 80 的全局最邻近点。显然,借助该二叉搜寻树,可使搜寻次数由 99 次降为 14 次。当 n 越大时,节省的搜寻次数会越多。

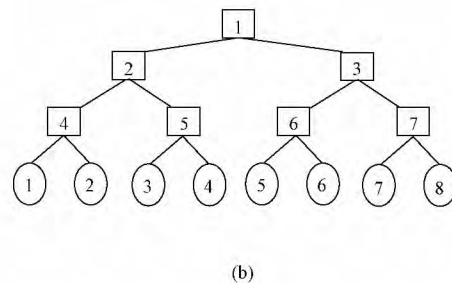


图 1 2-d tree 示例

需要注意的是, K-d tree 分割 n 个点的区域时, K-d tree 的层数 t 为参数,可以确定不可继续分割区域数量为 2^t 个,如本例取 $t=3$,所以最后的不可继续分割区域数量为 8。由于 K-d tree 用于求节点的最邻近节点,所以每个不可继续分割区域中的点数量大于等于 2,所以 t 为小于等于 $\log n - 1$ 大于等于 0 的整数($t=0$ 表示点所在区域未分割)。研究表明建立 K-d tree 的复杂度为 $O(Kn \log n)$,在 K-d tree 中搜寻某点的

最邻近点的复杂度为 $O(t + n/2^t)$ ^[21-25]。

2.3 Held Karp 模型

Held 和 Karp 提出了经典的 Held Karp 模型,用于求解 TSP 问题的最优解下界。该模型的原理是使用一个实数向量 $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ 改造 TSP 问题距离矩阵的模型,如式(17)所示^[26]。

$$\text{Held Karp model} \begin{cases} D' = (d'_{ij})_{n \times n} \begin{cases} d'_{ij} = d_{ij} + \pi_i + \pi_j (i \neq j) \\ d'_{ij} = MM \text{ 为足够大的数} (i = j) \end{cases} \\ \pi_i \in R (1 \leq i \leq n) \end{cases} \quad (17)$$

其中 D' 代表改变后的距离矩阵, R 代表任意实数集合。

Held 和 Karp 已经证明该模型不会改变 TSP 问题的最优解,其原理是 TSP 的可行解是 Hamilton 回路,在该回路中每个节点恰好连接两条边。因此该模型相当于将解空间中所有的可行解的路径长度都加了 $\Delta = 2 \sum_{i=1}^n \pi_i$,然而,当确定一个实数向量 π 以后, Δ 为常数。所以如果一个解根据原始距离矩阵 $D = (d_{ij})_{n \times n}$ 计算路径长度是解空间中的最优解,那么根据 Held Karp 模型改变后的距离矩阵 D' 计算仍然是最优解^[26]。

3 CVRP-IMGR

3.1 CVRP-IMGR 原理

GR 在求解旅行商^[21]、匹配^[22]、调度^[23]等组合优化问题中得到了有效的应用,但目前尚未应用于求解 CVRP 问题。本文提出的求解大规模 CVRP 的快速改进贪婪算法(Improved Greedy heuristic for solving CVRP, 简称 CVRP-IMGR),包括三个组成部分:求解 CVRP 的标准 GR 算法(Greedy heuristic for solving CVRP, 简称 CVRP-GR)、基于 K-d tree 的提高速度策略、基于 Held Karp 模型的改进质量策略,详细描述如下。

3.1.1 CVRP-GR

由于 CVRP 问题的可行解要求形成 m 个简单圈, 且每个圈经过的节点对应需求总量不超过 Q , 每个圈的长度不超过 L 。因此 CVRP-GR 的求解过程是:

- (1) 将所有任意两个顾客对应节点形成的 $C_n^2 = n(n-1)/2$ 条边, 按边长从小到大排序;
- (2) 并依次判断是否为合法边, 是合法边添加至当前路径, 否则不予考虑继续判断下一条边;
合法边应该满足以下 4 条规则:
规则 1: 添加该边后不会使任何点的连接边数量大于 2;
规则 2: 添加该边也不会使图中产生圈;
规则 3: 添加该边生成的单条路径包含的所有顾客需求总量, 不会超过最大载量 Q ;
规则 4: 添加该边所在的路径长度与该路径两端点至配送中心的长度之和, 不会超过最大路径长度限制 L ;
- (3) 直到没有任何合法边存在;
- (4) 得到 m 条路径, 将每条路径的两端点分别和配送中心节点 0 连接形成 m 个 Hamilton 回路。

按以上法则, CVRP-GR 与求解 TSP 问题的 GR 相同也要对所有的 $n(n-1)/2$ 条边排序, 但判断规则为 4 条多于 GR 的 2 条, 所以 CVRP-GR 的计算量比 GR 求解 TSP 的计算量式 (16) 所示更大:

$$\begin{aligned} f(n) &= \frac{n^2-n}{2} [\log(n^2-n) - 1] + 2(n^2-n) \\ &= \frac{n^2-n}{2} [\log(n^2-n) + 3] \leq \frac{n^2}{2} \log n^2 = n^2 \log n \end{aligned} \quad (18)$$

如式 (16) 和 (18) 所示, 虽然 CVRP-GR 和 GR 的计算量略有不同, 但它们的计算量的上界一致, 所以它们具有相同的复杂度 $O(n^2 \log n)$ 。该复杂度随着算例规模 n 的增加, 其耗费的时间会迅速增加, 不适合求解大规模的算例。另外, CVRP-GR 存在一个很大的求解质量弊端: 其形成的 m 条路径的端点可能离配送中心节点 0 的距离非常远, 因此最后形成的 m 个回路的总长度会比较长, 也就是说 CVRP-GR 的求解质量不高。

为了能够使 CVRP-GR 求解大规模的 CVRP 问题, 并得到较为满意的解, 我们分别借助 K -d tree 方法和 Held Karp 模型用于降低 CVRP-GR 的复杂度和提高求解质量。

3.1.2 基于 K -d tree 提高速度策略

CVRP-GR 非常耗时的主要原因是计算了所有 $n(n-1)/2$ 条边的边长, 且将其按升序排列。我们结合 K -d tree 方法提出了一种新的执行方式, 步骤如下:

- (1) 基于 CVRP 的 n 个顾客的坐标建立 K -d tree;
- (2) 建立长度为 n 的结构数组 $Heap$ 用于存放潜在的合法边信息, 确切的说, $Heap$ 中存放可连接节点 (已连接的边数小于 2) 与对应的最邻近点 (基于 K -d tree 搜寻) 形成的边 (不违反 4 条准则) 和对应边长, 并以升序的方式排列;
- (3) 提取 $Heap$ 中的最短边, 依照 3.1.1 中的 4 条规则判

断是否为合法边, 如果是执行 (4), 否则执行 (5);

- (4) 将合法边添加至当前路径;
- (5) 更新 $Heap$, 判断 $Heap$ 是否为空, 如果否转到步骤 (3), 否则执行终止。

该方法的合理性: 因为 $Heap$ 结构数组中存放了每个节点与最邻近节点对应的边, 显然 $Heap$ 中最短的边必定为当前的最短边, 换言之, 该执行方式完全遵循了 CVRP-GR 的每次添加当前最短合法边的原理。另外, 构造和更新 $Heap$ 涉及的搜寻最邻近点计算均基于 K -d tree 执行, 从而可以有效提高该执行方式的效率 (详见 3.3)。

3.1.3 基于 Held Karp 模型改进质量策略

Held Karp 模型用于改造 TSP 的距离矩阵, 且不改变 TSP 问题的最优解。其实该模型同样可以用于 CVRP 问题, 即有定理 1 成立。

定理 1: 根据 Held Karp 模型使用向量 $\pi = (0, \pi_1, \pi_2, \dots, \pi_n)$ 改造 CVRP 问题中 1 个配送中心和 n 个顾客位置之间形成的 $(n+1)$ 阶距离矩阵, 不会改变 CVRP 问题的最优解。

证明: 由本文 1 中对 CVRP 的定义可知, 一个派送 k 辆车的 CVRP 解, 相当于在图 G 中过覆盖所有节点的 k 个圈, 且每个圈均经过配送中心节点 0, 且每个顾客节点 $i \in V \setminus \{0\}$ 的度数为 $du_i = 2$, 即每个顾客节点仅有 2 条边相连, 那么配送中心节点 0 的度数为 $du_0 = 2k$ 。由图论知识每条边均有 2 个端点可知, 该 k 个圈共包括 $(2k+2n)/2 = n+k$ 条边, 设该 $n+k$ 条边的集合用 SE 表示, 则 $(i, j) \in SE$ 表示边 (i, j) 是 $n+k$ 条边之一。如果使用向量 $\pi = (0, \pi_1, \pi_2, \dots, \pi_n)$ 按 Held Karp 模型 (见公式 (17)) 改变由 1 个配送中心 n 个顾客形成的 $n+1$ 阶距离矩阵, 那么 L_{total} 和 L'_{total} (L_{total} 为距离矩阵改变前 $n+k$ 条边长度之和; L'_{total} 为距离矩阵改变后 $n+k$ 条边长度之和) 的关系如下所示:

$$\begin{aligned} L'_{total} &= \sum_{(i,j) \in SE} d'_{ij} = \sum_{(i,j) \in SE} (d_{ij} + \pi_i + \pi_j) \\ &= \sum_{(i,j) \in SE} d_{ij} + \sum_{(i,j) \in SE} (\pi_i + \pi_j) \\ &= L_{total} + du_0 \times 0 + \sum_{i=1}^n (du_i \times \pi_i) \\ &= L_{total} + 2 \sum_{i=1}^n \pi_i \end{aligned}$$

因此, 在 CVRP 问题解空间中的每个可行解, 根据改造后距离矩阵计算总长度相当于增加了常数量 $2 \sum_{i=1}^n \pi_i$, 故原来在解空间中的最短 (最优) 路径仍然最短, 所以定理 1 成立。证毕!

由于 CVRP-GR 存在的求解质量缺陷是, 最后形成的 m 条路径的 $2m$ 个端点可能离配送中心的距离很远, 所以如果能够使 $2m$ 个端点均为离配送中心较近的点就能提高 CVRP-GR 的求解质量提高。那么如果能够优先连接离配送中心较远的点, 自然最后形成的路径端点会离配送中心较近。通过式 (17) 不难发现, Held Karp 模型相当于对每条边 d_{ij} 的边长增加了 $\pi_i + \pi_j$, 所以如果能够让离配送中心较远的边更短, 而离配送中心较近的边更长, CVRP-GR 将会优先连接离配

送中心较远的点。所以通过巧妙的设计向量 $\pi = (0, \pi_1, \pi_2, \dots, \pi_n)$ 的取值, 可以实现该思想。本文使用如式 (19) 的方式计算向量 π ,

$$\pi = (0, -\alpha d_{01}, -\alpha d_{02}, \dots, -\alpha d_{0i}, \dots, -\alpha d_{0n}) \quad (19)$$

其中 $\alpha \geq 0$ 为参数, 其绝对值 α 表示优先添加离配送中心较远边的优先程度; d_{0i} 表示顾客 i 到配送中心的距离。

3.2 CVRP-IMGR 步骤

CVRP-IMGR 的步骤流程图如图 2 所示, 每个步骤的说明如下:

Step1. [读取 CVRP] 读取 CVRP 问题信息, 包括: 配送中心和顾客位置的坐标值、顾客需求量 q_i 、 Q 和 L 信息;

Step2. [设参数 t 和 α] CVRP-IMGR 中包括两个参数, 即 K -d tree 的层数 t 和公式 (19) 中的 α ;

Step3. [建立 K -d tree] 根据算例顾客和配送中心的坐标信息、已设定参数 t 建立 K -d tree;

Step4. [计算向量 π] 根据坐标信息和已设定参数 α 按照公式 (19) 计算向量 π ;

Step5. [建立 Heap] 基于 K -d tree 和用向量 π 依据 Held Karp 模型改造后距离矩阵, 计算每个节点 i 对应的最邻近节点 j , 并将边 (i, j) 和对应长度 $d_{ij} + \pi_i + \pi_j$ 存至结构数组 Heap, 且 Heap 中的元素依照边长升序排列;

Step6. [从 Heap 提取最短边] 提取 Heap 中的第一条边, 假设为 (x, y) ;

Step7. [是合法边?] 如果添加边 (x, y) 至当前路径是否满足 3.1.1 中的所有 4 条规则, 满足返回“是”执行 Step9, 否则返回“否”执行 Step8;

Step8. [更新 Heap] 更新当前 Heap 数组, 即保证每个可连接节点与最邻近节点形成的边均在 Heap 中, 然后执行 Step6;

Step9. [添加边至路径] 添加边 (x, y) 到当前路径中;

Step10. [得到最终解?] 确定是否还有合法边存在, 如果有返回“否”执行 Step8; 否则返回“是”, 将形成的 m 条路径的端点分别与节点 0 连接得到最终解, 结束程序。

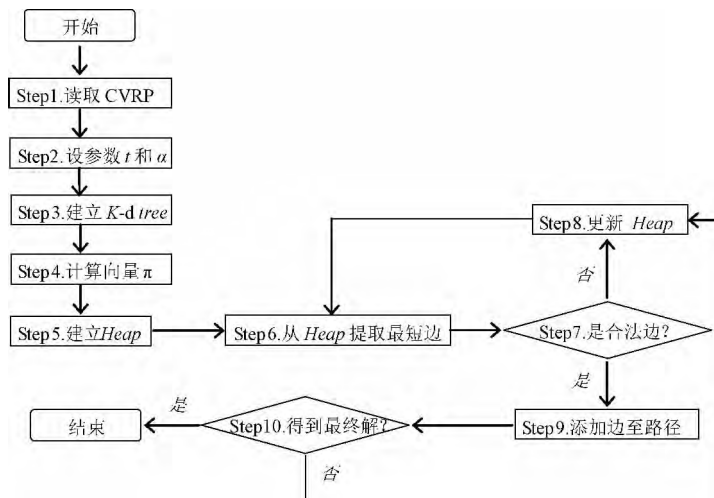


图 2 CVRP-IMGR 的执行步骤流程图

3.3 CVRP-IMGR 复杂度分析

根据 3.2 中 CVRP-IMGR 的步骤可知, CVRP-IMGR 的计算量主要包括建立 K -d tree、建立和不断更新结构数组 Heap。其中建立 K -d tree 的复杂度为 $O(Kn \log n)$ [24], 在我们的程序中通过调用函数 $Insert(i, NN(i), l)$ 实现建立和更新结构数组 Heap, 其中 $NN(i)$ 函数返回节点 i 的最邻近节点; $Insert(i, NN(i), l)$ 将节点 i 和其最邻近节点 $NN(i)$ 及对应的边长存入 Heap。另外, 函数 $NN()$ 基于 K -d tree 执行, 其复杂度为 $O(t + n/2^t)$, $Insert(i, NN(i), l)$ 的复杂度为 $O(\log s)$ (s 为 Heap 中当前存在的边数量, 且 $1 \leq s \leq n$)。因此调用函数 $Insert(i, NN(i), l)$ 的次数是决定 CVRP-IMGR 计算量的一个关键因素, Bentley 通过数据分析证明调用 $Insert(i, NN(i), l)$ 为 un 次, 且 u 为与 n 无关的常数量 [21, 25], 那么 CVRP-IMGR 的总计算量如下,

$$f(n) = Kn \log n + un \left(t + \frac{n}{2^t} \right) + un \log s \quad (20)$$

又因 t 为 $[0, \log n - 1]$ 范围内的整数 (见 2.2 节), s 为 $[1, n]$ 的整数。因此, 基于式 (20) 有如下不等式成立,

$$Kn \log n + un(\log n + 1) \leq f(n) = Kn \log n + un \left(t + \frac{n}{2^t} \right) + un \log s \leq Kn \log n + un^2 + un \log n \quad (21)$$

根据 t 取值的不同 CVRP-IMGR 的复杂度介于 $O(n \log n)$ 和 $O(n^2)$ 之间, t 取值越大 CVRP-IMGR 复杂度越小, 如果 $t = \lceil \log n - h \rceil$ ($h \geq 1$ 的整数), CVRP-IMGR 的复杂度可以达到 $O(n \log n)$ 。

4 算例求解与分析

4.1 求解说明

算例选择: 本文提出的 CVRP-IMGR 目标是能够快速求解大规模的 CVRP 问题, 因此选择算例的第一原则是规模足够大。目前, 有很多作者提出了标准的 CVRP 算例, 其中 Li 等在 2005 年提出了 12 个 n 为 560 ~ 1200 的算例 [15], 2007 年

Kytojoki 等在 Li 的基础上提出了 12 个 n 为 2400 ~ 20000 的算例^[16]。这 24 个算例是目前最大规模的 CVRP 算例。除此之外, 选择该 24 个算例的另一个原因是: 这些算例的顾客位置分布具有一定的规律性, 基于该规律性可以近似计算出该 24 个算例的最优解, 据此可以方便的评估算法的求解质量。

算法选择: 我们选择了三种算法进行比较, CVRP-IMGR、CVRP-GR、Savings 算法。选择 CVRP-GR 算法是为了考察本文提出的 CVRP-IMGR 算法的两个改进策略的实际效果。而选择 Savings 算法, 是因为 Savings 算法是非常经典有效的 CVRP 算法^[1], 被研究者们广泛认可, Savings 的复杂度与 CVRP-GR 相同为 $O(n^2 \log n)$, 而且和 CVRP-GR、CVRP-IMGR 同属于构建型算法, 所以选择 Savings 作为本文提出的 CVRP-IMGR 的比较基准可以更清晰的反映 CVRP-IMGR 的求解效率和求解质量。为了呈现 CVRP-IMGR 的鲁棒性, 我们在求解所有算例时 CVRP-IMGR 中的参数 α 均取值为 0.70, 使用的 $K-d$ tree 的层数 $t = \lceil \log n - 2 \rceil$ ($\lceil \cdot \rceil$ 表示取 “ ” 的整数部分, 注意此时 CVRP-IMGR 复杂度为 $O(n \log n)$), 参

数取值详见论文 4.3 部分。

求解环境: Studio Visual C++ 6.0; CPU 为 Inter(R) core (TM) Q9400, 内存为 4.0G, 主频为 2.66GHz。

4.2 求解结果

三种算法求解 24 个算例的结果如表 1 所示。表 1 中字段说明: 算例名称: 该 24 个算例的生成原理均由 Li 等提出, 所以我们命名算例的方式为 CVRP-Li + n ; 最优解长度: 由于该 24 个算例的顾客位置具有特殊规律性, Li 等和 Kytojoki 等根据此特点推测出了他们的近似最优解长度; 求解质量%: 该指标用三种算法的解超过最优解长度的百分比度量, 计算方法为 $100 \times (\text{算法求得解长度} / \text{最优解长度} - 1)$, 三种算法中的最优求解质量加粗表示; 求解耗时(s): 为准确计算求解每个算例的耗时, CVRP-IMGR 重复运行 100 遍, 另外两种算法运算 20 遍, 然后再计算平均耗时。另外, 由于算法 CVRP-GR 和 Savings 需要存储所有的 $n(n-1)/2$ 条边占用内存很大, 所以在本计算环境中无法求解最大算例 CVRP-Li20000, 对应的求解质量和耗时在表 1 中用 “-” 表示。

表 1 三种算法求解算例的结果

序号	算例名称	最大行驶距离限制 L	最优解长度	求解质量%			求解耗时(s)		
				CVRP-GR	CVRP-IMGR	Savings	CVRP-GR	CVRP-IMGR	Savings
1	CVRP-Li560	1800	16212.83	85.61	10.20	14.83	0.078	0.010	0.109
2	CVRP-Li600	1000	14652.28	35.76	8.03	5.27	0.093	0.012	0.110
3	CVRP-Li640	2200	18801.13	43.15	8.90	13.90	0.125	0.012	0.140
4	CVRP-Li720	2400	21389.43	85.51	7.03	15.99	0.141	0.014	0.172
5	CVRP-Li760	900	17053.26	24.98	8.56	4.37	0.171	0.016	0.188
6	CVRP-Li800	2500	23977.74	90.40	10.11	10.16	0.188	0.016	0.218
7	CVRP-Li840	900	18253.55	19.45	4.25	1.54	0.203	0.018	0.234
8	CVRP-Li880	2800	26566.04	89.92	10.73	13.90	0.235	0.017	0.266
9	CVRP-Li960	3000	29154.34	84.95	8.21	9.27	0.281	0.019	0.328
10	CVRP-Li1040	3200	31742.64	84.64	14.43	13.37	0.359	0.016	0.391
11	CVRP-Li1120	3500	34330.94	84.38	8.11	10.43	0.421	0.018	0.469
12	CVRP-Li1200	3600	36919.24	83.81	10.83	14.04	0.484	0.018	0.562
13	CVRP-Li2400	7800	75743.77	85.53	7.89	15.62	2.781	0.032	3.062
14	CVRP-Li3600	11700	114568.30	85.68	6.07	19.56	7.500	0.052	7.797
15	CVRP-Li6000	19500	192217.35	85.58	5.60	6.03	24.515	0.075	24.234
16	CVRP-Li7200	23400	231041.88	85.56	6.19	19.43	36.453	0.096	36.125
17	CVRP-Li8400	27300	269866.41	85.55	7.60	4.05	51.360	0.092	50.532
18	CVRP-Li9600	31100	308690.94	85.53	4.02	14.35	68.938	0.107	67.859
19	CVRP-Li10800	35100	347515.46	85.44	6.72	22.30	89.750	0.124	87.453
20	CVRP-Li12000	39000	386339.99	85.44	3.16	6.04	112.562	0.141	109.828
21	CVRP-Li13200	42900	425164.52	85.44	8.77	15.25	138.547	0.160	135.281
22	CVRP-Li14400	46800	463989.05	85.44	8.76	21.12	167.672	0.179	163.453
23	CVRP-Li16800	54500	541638.10	85.44	3.73	1.18	234.797	0.173	228.032
24	CVRP-Li20000	64800	645170.17	—	5.18	—	—	0.217	—
Average		76.22		7.63	11.83	40.768	0.068	39.863	
Standard deviation		21.70		2.64	6.20	—	—	—	

求解质量: 如表1所示, 对所有的24个算例 CVRP-IMGR 和 Savings 的求解质量均明显优于 CVRP-GR, 另外, CVRP-IMGR 有18个算例(占有算例的75%)的求解质量比 Savings 更好。CVRP-IMGR 求解19个算例的质量在10%以下, 只有5个算例在10%以上, 其平均求解质量为7.63%, 明显优于 CVRP-GR 的76.22% 和 Savings 的11.83%。另外

CVRP-IMGR 的求解质量分布在3.73%~14.43%之间, 解质量的标准差为2.64%; 而 CVRP-GR 和 Savings 的求解质量分布范围分别为19.45%~89.92%和1.18%~22.30%, 标准差分别为21.7%和6.20%。也就是说, CVRP-IMGR 的求解质量受算例规模差异性的影响最小, 更具鲁棒性。这三种算法的求解质量折线图如图3所示。

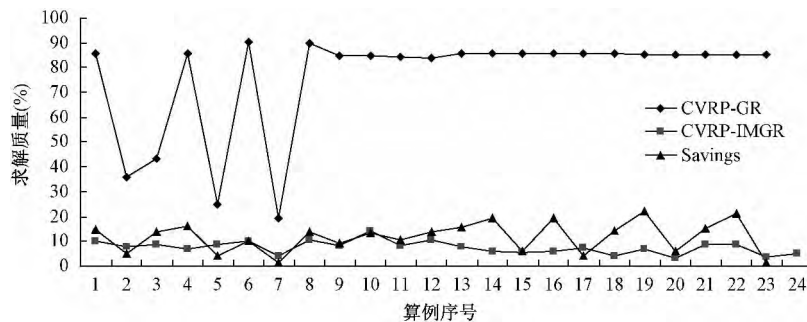


图3 算法 CVRP-GR、CVRP-IMGR 和 Savings 求解质量折线图

由上述图表数据可知, CVRP-IMGR 的平均求解质量为7.63%, 为可视化 CVRP-IMGR 求解质量的效果, 我们画出了 CVRP-IMGR 求解 CVRP-Li720 的解(求解质量为7.03%, 接近 CVRP-IMGR 的平均水平), 如图4(a)所示。图4(b)是文献[15]中设计该算例的作者根据点分布的几何属性推断出

来的理论最优解。通过对比两图可知, CVRP-IMGR 的解与最优解具有高度相似性(具有较多相同边), 也就是说, 改进型算法(如 k -opt 算法)在此基础上可以通过替换少量的边就能接近(得到)最优解。

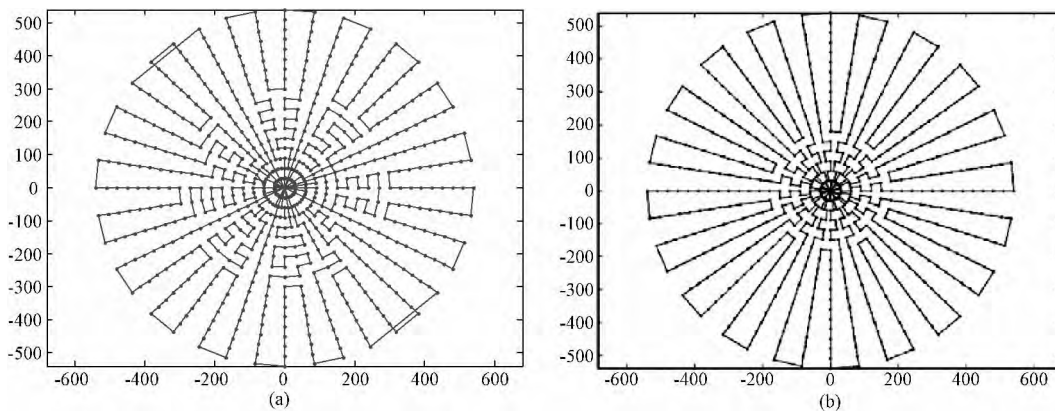


图4 CVRP-IMGR 求解 CVRP-Li720 的解路径图

求解耗时: 由于 CVRP-IMGR 的复杂度为 $O(n \log n)$, CVRP-GR 和 Savings 的复杂度均为 $O(n^2 \log n)$, 所以 CVRP-IMGR 的求解速度远远快于后两者。由表1可知, CVRP-IMGR 的求解速度大体比 Savings 快10~1300倍左右, 且 n 越大优势越明显。另外, CVRP-IMGR 求解当前最大规模的算例 CVRP-Li20000, CVRP-IMGR 耗时仅为0.217s, 所以 CVRP-IMGR 即使求解大规模 CVRP 问题, 其求解速度也非常快。需注意的是, 算法的耗时不由 n 唯一确定, 算例本身的差异性对算法的耗时也略有影响, 如 CVRP-IMGR 求解 CVRP-Li14400 的耗时为0.179s 反而高于更大规模算例 CVRP-Li16800 的0.173s。

通过比较 CVRP-IMGR 和 CVRP-GR、Savings 的结果可以发现, CVRP-IMGR 结合的 K -d tree 和 Held-Karp 模型策略能够非常有效的提高其求解速度和求解质量。CVRP-IMGR 的

求解速度和平均求解质量均优于经典的 CVRP 构建型算法 Savings。

4.3 CVRP-IMGR 的参数设定

CVRP-IMGR 中包含两个参数, 即 K -d tree 中的层数 t 和优先添加离配送中心节点0较远边的优先程度 α 。参数 t 的取值范围是大于等于1小于等于 $\log n - 1$ 的整数, 我们分别选择 t 等于 $\lceil \log n - 1 \rceil$ 、 $\lceil \log n - 2 \rceil$ 、 \dots 、 $\lceil \log n - 9 \rceil$, 共9个值(此时设参数 α 等于0.70)并采用对应 CVRP-IMGR 求解了以上24个算例, 每个算例运算了100遍, 平均求解质量和平均运算时间结果如图5(a)所示。由计算结果可知: 参数 t 的取值对求解质量影响不大(在7%~8%间波动); 而正如式(21)的分析结果, t 值的变动会明显影响 CVRP-IMGR 的复杂度即运算时间, 随着 h 的增加(即随 t 的减少), 平均耗时由0.057s 迅速增加至2.916s。因此我们建议参数 t 值取上

限 $[\log n - 1]$ 附近的值。

由式 (19) 可知, 参数 α 的取值范围为大于等于零的数, 当 $\alpha = 0$ 时表示不优先添加远边即 CVRP-IMGR 等同于基于 K - d tree 的 CVRP-GR, 当 $\alpha > 0$ 时表示优先添加远边, 且 α 值越大表示优先程度越高, 那么 α 值为多少时 CVRP-IMGR 的求解质量最好? 求解经验表明, 参数 α 值与 CVRP-IMGR 求

解质量存在一定的趋势关系, 我们分别取 α 等于 0.10、0.11、0.12、 \dots 、0.99, 共 90 个值 (此时设参数 t 等于 $[\log n - 2]$), 然后采用对应 CVRP-IMGR 求解了 5.1 中提到的 24 个算例, 并计算了不同参数值求解 24 个算例的平均求解质量, 其结果如图 5(b) 所示。

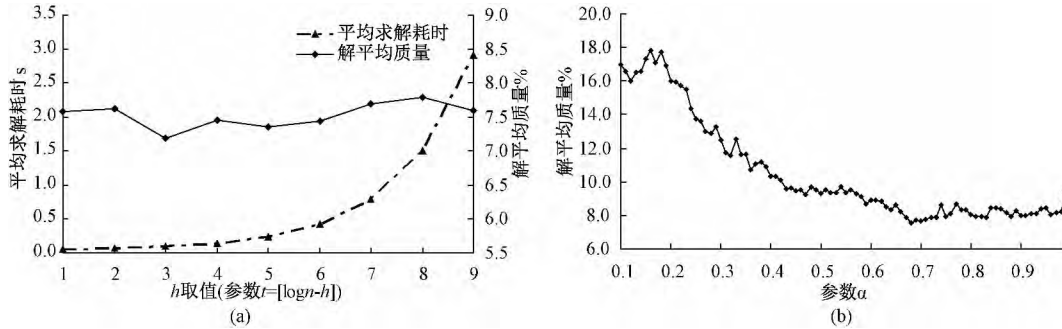


图 5 参数 t 和 α 取不同值 CVRP-IMGR 的性能

由图 5(b) 可知, 当参数 α 取值在 $[0.10, 0.99]$ 范围内 CVRP-IMGR 的平均求解质量具有一定的规律性, α 取值在 $[0.10, 0.45]$ 时, CVRP-IMGR 的平均求解质量由 17% 左右较快的提高至 10% 左右, 而取值在 $[0.46, 0.95]$ 范围内时 CVRP-IMGR 的平均求解质量在 7.5 ~ 9.5% 内波动, 当 α 大于 0.96 时 CVRP-IMGR 的平均求解质量有降低的趋势。通过该实验表明 α 的良好取值范围为 $[0.46, 0.95]$, 但为保证 CVRP-IMGR 的鲁棒性, 我们建议 α 的范围在 $[0.60, 0.80]$ 。

4.4 CVRP-IMGR 的性能评估

CVRP-IMGR 属于构建型算法, 即根据规则得到解后不再进行改进。在目前文献中有很多的改进型算法, 即在构建型算法求解的结果基础上, 进一步通过尝试删除一部分边, 然后再添加新的边等方式改进当前解, 并且不断循环迭代, 直到满足预先设定的终止规则为止。因此, 一般来说构建型算法的求解质量不如改进型算法, 但改进型算法的耗时会远远大于

构建型算法。为了评估 CVRP-IMGR 的综合性能, 除表 1 中与经典的构建型算法 Savings 比较之外, 我们在此将 CVRP-IMGR 的求解质量和求解耗时进一步和当前文献中的几种改进型算法进行了比较, 如表 2 所示。表 2 说明: (1) CVRP-IMGR 的参数 $t = [\log n - 2]$, $\alpha = 0.70$; (2) RTR, 为算法 Record-to-Record 算法的缩写^[27]; (3) NFTS, 为 Network Flow-based Tabu Search 算法缩写^[27]; (4) GTS, 为 Granular Tabu Search 算法的缩写^[28]; (5) 由于 CVRP-IMGR 速度很快, 为精确记录耗时, 每个算例重复计算 3000 遍, 记录总时间再求平均值; (6) 文献中的 3 种算法 RTR、NFTS、GTS 没有记录每个算例的耗时, 仅有大致的平均耗时^[15]; (7) CVRP-IMGR 的求解机器配备见本文 4.1, RTR、NFTS 的机器为 Pentium 100 MHz CPU, GTS 为 Pentium 200 MHz CPU; (8) 表 2 中的 8 个算例来自文献 [27], 算例名称最后的数据为顾客数量, 如 1-Golden 240 表示有 240 个顾客。

表 2 DCVRP-IMGR 与几种改进型算法的性能比较

序号	算例名称	最大行驶距离限制 L	最优解长度	求解质量%				求解耗时			
				CVRP-IMGR	RTR	NFTS	GTS	CVRP-IMGR(s)	RTR(m)	NFTS(m)	GTS(m)
1	1-Golden240	650	5639.36	10.74	3.46	—	1.72	0.0057	—	—	—
2	2-Golden320	900	8447.92	7.80	6.56	1.45	1.24	0.0066	—	—	—
3	3-Golden400	1200	11036.23	14.34	7.65	7.65	3.32	0.0089	—	—	—
4	4-Golden480	1600	13624.53	9.38	7.45	11.94	9.44	0.0114	—	—	—
5	5-Golden200	1800	6460.98	4.67	3.74	13.93	3.66	0.0044	—	—	—
6	6-Golden280	1500	8412.88	8.52	7.18	8.03	6.54	0.0057	—	—	—
7	7-Golden360	1300	10195.56	10.53	9.98	11.93	3.45	0.0081	—	—	—
8	8-Golden440	1200	11696.55	12.94	6.99	9.65	2.90	0.0102	—	—	—
Average				9.87	6.63	9.23	4.03	0.0076	37.15	1825.59	17.55

通过比较求解质量发现, CVRP-IMGR 的平均求解质量低于 3 种改进型算法 RTR、NFTS 和 GTS, 但对于某些算例如

4-Golden480、5-Golden200、7-Golden360 的求解质量甚至略优于 NFTS 和 (或) GTS。然后, 在求解速度方面 CVRP-IMGR 远

远快于3种改进型算法。需要注意的是 CVRP-IMGR 的平均速度为 0.0076 秒,3种改进算法 RTR、NFTS 和 GTS 的平均求解速度分别为 37.15、1825.59、17.55 分钟。为做保守比较,假设我们实验平台的速度是 Pentium 100 MHz CPU 和 Pentium 200 MHz CPU 的 100 倍,那么通过折算可比速度, CVRP-IMGR 的求解速度分别是 RTR、NFTS 和 GTS 的 2923.61 (计算方法 $37.15 \times 60 / (0.0076 \times 100) = 2923.61$)、143669.34、1381.14 倍。

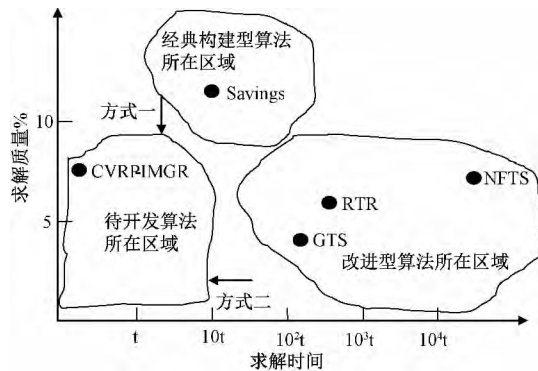


图6 不同算法的性能示意图

Toth 和 Vigo 采用求解质量和求解时间两个指标,评价了当前求解 CVRP 算法的性能以及算法研究的两种不同方式^[29]。总体来说,由于改进型算法是在构建型算法得出的解基础上进一步改进,所以其求解质量优于构建型算法,但求解时间高于构建型算法。然而改进型算法的求解时间可能是构建型算法的几百倍甚至成千上万倍。因此,Toth 和 Vigo 归纳当前改进 CVRP 算法的研究方式有两种:一是提高构建型算法的求解质量;二是提高改进型算法的求解速度,其中方式一的可能性要小于方式二^[29]。为便于比较当参数 $t = [\log n - 2]$ 和 $\alpha = 0.70$ 时的 CVRP-IMGR 与 Savings、RTR、NFTS 和 GTS 的相对性能,我们用图 6 描述了他们的求解耗时和求解质量性能关系。

总体来说, CVRP-IMGR 不仅求解质量优于经典构建型算法 Savings,甚至接近某些改进型算法,而且复杂度 $O(n \log n)$ 是当前 CVRP 算法中最低的。

5 结束语

本文基于经典贪婪算法简洁的思想,提出了求解 CVRP 问题的贪婪算法 CVRP-GR,然后根据 CVRP-GR 的复杂度高和求解质量的明显缺点,分别使用 $K-d$ tree 法和 Held Karp 模型提高了 CVRP-GR 的性能,由此得到有效的 CVRP-IMGR 算法。通过理论分析和求解算例得出如下主要结论:

(1) 提出的基于 $K-d$ tree 和 Held Karp 模型的两个策略,能够有效的提高 CVRP-GR 的求解速度和求解质量;

(2) CVRP-IMGR 的复杂度仅为 $O(n \log n)$,是当前求解 CVRP 复杂度最低的算法;

(3) CVRP-IMGR 的平均求解质量优于经典构建型算法 Savings,甚至接近某些改进型算法。

基于 CVRP-IMGR 快速和质量满意的特点可知, CVRP-

IMGR 的适用范围是求解规模非常大且(或)时效性要求很高的 CVRP 问题;另外, CVRP-IMGR 可以为改进型算法快速的提供满意的初始解群。我们在今后的研究中将采用 CVRP-IMGR 求解动态的车辆路径问题。

参考文献

- [1] Laporte G. Fifty years of vehicle routing[J]. Transport Science, 2009, 43(4): 408~416.
- [2] Potvin JY. State-of-the art review evolutionary algorithms for vehicle routing[J]. INFORMS Journal on computing, 2009, 21(4): 518~548.
- [3] Dantzig GB, Ramser JM. The truck dispatching problem[J]. Management Science, 1959, 6(1): 81~91.
- [4] Clarke G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points[J]. Operations Research, 1964, 12(4): 568~581.
- [5] Solomon MM. Algorithms for the vehicle routing and scheduling problem with time windows constraints[J]. Operations Research, 1987, 35(2): 254~265.
- [6] Labbe M, Laporte G, Mercure H. Capacitated Vehicle Routing Problems on Trees[J]. Operations Research, 1991, 39(4): 616~622.
- [7] Lin S. Computer solutions of the traveling salesman problem[J]. Bell System Technology Journal, 1965, 44: 2245~2269.
- [8] Gendreau M, Hert ZA, Laporte G. A tabu search heuristic for the vehicle routing problem[J]. Management Science, 1994, 40(10): 1276~1289.
- [9] 张建勇, 张军. 具有模糊旅行时间的 VRP 的一种混合遗传算法[J]. 管理工程学报, 2006, 20(4): 13~17.
- [10] 张军, 唐加福, 潘震东. 分散搜索算法求解带货物权重的车辆路径问题[J]. 系统工程学报, 2010, 25(1): 91~97.
- [11] 张燕, 周支立, 翟斌. 集货送货一体化的物流配送车辆路径问题的标号算法[J]. 运筹与管理, 2007, 16(3): 12~19.
- [12] Bullnheimer B, Hartl RF, Strauss C. An improved ant system algorithm for the vehicle routing problem[J]. Annals of Operations Research, 1999, 89(0): 319~328.
- [13] 赵燕伟, 彭典军, 张景玲, 等. 有能力约束车辆路径问题的量子进化算法[J]. 系统工程理论与实践, 2011, 29(2): 159~166.
- [14] 吴斌, 钱存华, 董敏, 等. 具有同时集送货需求车辆路径问题的混沌量子进化算法研究[J]. 控制与决策, 2010, 25(3): 383~388.
- [15] Li FY, Golden B, Wail Edward. Very large-scale vehicle routing: new test problems, algorithms, and results[J]. Computers & Operations Research, 2005, 32(5): 1165~1179.
- [16] Kytojokio J, Nuortio T, Braysy 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.
- [17] Figliozzi MA. An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows[J]. Transportation Research Part C, 2010, 18(5): 668~679.

- [18] Mestera D , Braysy O. Active-guided evolution strategies for large-scale capacitated vehicle routing problems [J]. Computers & Operations Research , 2007 , 34(10) : 2964 ~ 2975.
- [19] Alba E , Dorronsoro Bernabé. Computing nine new best-so-far solutions for capacitated VRP with a cellular genetic algorithm [J]. Information Processing Letters , 2006 , 98(6) : 225 ~ 230.
- [20] Lin SW , Lee ZJ , Ying KC , et al. Applying hybrid meta-heuristics for capacitated vehicle routing problem [J]. Expert Systems with Applications , 2009 , 36(2) : 1505 ~ 1512.
- [21] Bentley JL. Fast algorithm for geometric traveling salesman problem[J]. ORSA Journal on Computing , 1992 , 4(4) : 387 ~ 411.
- [22] Adamczyk M. Improved analysis of the greedy algorithm for stochastic matching [J]. Information Processing Letters , 2011 , 111(15) : 731 ~ 737.
- [23] Ribas I , Companys R , Tort-Martorell X. An iterated greedy algorithm for the flowshop scheduling problem with blocking [J]. Omega , 2011 , 39(3) : 293 ~ 301.
- [24] Curtis SA. The classification of greedy algorithms [J]. Science of Computer Programming , 2003 , 49(1 - 3) : 125 ~ 157.
- [25] Bentley JL. K -d trees for semidynamic point sets [C]. In Proc. 6th Ann. ACM Symp on Computational Geometry , 1990 , 187 ~ 197.
- [26] Held M , Karp RM. The traveling salesman problem and minimum spanning trees [J]. Operations Research , 1970 , 18(6) : 1138 ~ 1162.
- [27] Golden BL , Wasil EA , Kelly JP , et al. The impact of metaheuristics on solving the vehicle routing problem: algorithms , problem sets , and computational results [M]. In: Crainic T , Laporte G , editors. Fleet management and logistics. Boston , MA: Kluwer , 1998. pp. 33 ~ 56.
- [28] Toth P , Vigo D. The granular tabu search and its application to the vehicle-routing [J]. INFORMS Journal on Computing , 2003 , 15(4) : 333 ~ 346.
- [29] Toth P , Vigo D. The vehicle routing problem [M]. Philadelphia: SIAM , 2002. 130 ~ 149.

An Efficient Greedy Heuristic for Solving Large-scale Capacitated Vehicle Routing Problem

RAO Wei-zhen , JIN Chun

(Institute of Systems Engineering , Dalian University of Technology , Dalian 116024 , China)

Abstract: The *Vehicle Routing Problem* (VRP) was introduced by Dantzig and Ramser in 1959. VRP calls for the determination of the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers. VRP is one of the most studied combinatorial optimization problems. Hundreds of heuristics have been developed. However , all existing heuristics have complexity larger than $O (n^2)$, which can lead to the incapability of solving very large-scale VRP in short time. Therefore , we propose an improved greedy heuristic with complexity $O (n \log n)$ for solving large-scale *Capacitated Vehicle Routing Problem* (CVRP) , which is a classic version of VRP. This paper can be divided into five sections.

In Sections 1 , we define CVRP and present a new model for CVRP. Section 2 introduces Greedy heuristic (GR) , K -d tree and Held Karp model. GR is a classic construction heuristic for solving combinatorial optimization problems , but it hasn't been applied to solving CVRP. A greedy heuristic for solving CVRP (CVRP-GR) based on the traditional GR is proposed in Section 3. However , CVRP-GR has complexity $O (n^2 \log n)$ and is not efficient enough. In addition , CVRP-GR cannot generate solutions with good quality because it adds very long edges into solution. Two enhancements for improving the speed and solution quality of CVRP-GR are presented by using K -d tree and Held Karp model , respectively. Consequently , CVRP-GR is combined with two enhancements results in an improved CVRP-GR (CVRP-IMGR) . We have proved that CVRP-IMGR has only complexity $O (n \log n)$, which is very efficient , and can solve large-scale CVRP instances in short time. Furthermore , the detail steps of CVRP-IMGR are presented in the last part of Section 3.

In order to evaluate the performance of CVRP-IMGR , we solve 24 largest CVRP instances by using CVRP-GR , CVRP-IMGR and Savings that is the best-known heuristic for solving CVRP , in Section 4 of this paper. The computational results show that CVRP-IMGR is much more efficient than CVRP-GR and Savings , because the later two heuristics have complexity $O (n^2 \log n)$. For all 24 instances , CVRP-IMGR and Savings have better solutions than CVRP-GR. Moreover , CVRP-IMGR generates better solutions than Savings in 18 out 24 instances. The relation between parameters in CVRP-IMGR and performance of CVRP-IMGR is also discussed in Section 4.

In the last section , we draw the conclusion that (1) two enhancements based on K -d tree and Held Karp model are very efficient and effective; (2) CVRP-GR involving two enhancements can result in CVRP-IMGR that has only complexity $O (n \log n)$; and (3) CVRP-IMGR is able to generate better solutions than classic construction heuristic savings. Future researches may want to apply CVRP-IMGR to solving VRP that requires short response time , e. g. dynamical VRP (DVRP) .

Key words: Capacitated vehicle routing problem; Greedy heuristic; K -d tree; Held Karp model

中文编辑: 杜 健; 英文编辑: Charlie C. Chen