

多车次同时送取货物车辆路径问题的量子蚁群算法

张惠珍¹, 赵玉苹²

(1. 上海理工大学 管理学院, 上海 200093; 2. 国网上海市电力公司物资公司, 上海 200002)

摘要: 研究了配送车辆载重量和工作时间有限, 考虑货物装卸时间的多车次同时送货和取货的车辆路径问题 (multi-trip vehicle routing problem with simultaneous deliveries and pickups, MTVRPSDP), 建立了以配送车辆启动成本和车辆行驶成本之和最小为目标的线性整数规划模型. 将量子计算和基本蚁群算法相结合提出了求解 MTVRPSDP 的量子蚁群算法, 该算法应用量子比特启发式因子改进了人工蚂蚁的转移概率, 从而提高了算法的全局搜索能力和稳定性, 有效改进了算法陷入局部最优的缺陷. 算例分析表明: MTVRPSDP 的线性整数规划模型在实际应用中是可行和有效的, 而且相比于基本蚁群算法和文献中所给其他算法的计算结果, 利用量子蚁群算法和 MTVRPSDP 的线性整数规划模型能够得到较好的满意解, 安排的车辆配送路线更加经济合理.

关键词: 车辆路径问题; 同时送取; 多车次; 蚁群算法; 量子计算

中图分类号: TP 18 **文献标志码:** A

Quantum-Inspired Ant Colony Algorithm for the Multi-Trip Vehicle Routing Problem with Simultaneous Deliveries and Pickups

ZHANG Huizhen¹, ZHAO Yuping²

(1. Business School, University of Shanghai for Science and Technology, Shanghai 200093, China;

2. State Grid Shanghai Procurement Company, Shanghai 200002, China)

Abstract: The multi-trip vehicle routing problem with simultaneous deliveries and pickups (MTVRPSDP) was studied in consideration of the loading and discharging time, maximum vehicle transport time and load capacity. A linear integer programming model for the MTVRPSDP was formulated, in which the objective function was to minimize the total distribution costs including vehicle costs and transportation costs. A quantum-inspired ant colony optimization (QACO) algorithm for solving the MTVRPSDP was proposed by combining the quantum computing and basic ant colony optimization (ACO). Owing that the transition probability of artificial ants was improved by using the heuristic factor of quantum bits in the QACO, the global search ability and stability of the algorithm have a better improvement, and its disadvantage of getting into the local optimum is also effectively weakened. The numerical results show that the linear integer programming model

收稿日期: 2017-04-06

基金项目: 国家自然科学基金资助项目(71401106); 上海市教育委员会科研创新项目(14YZ090); 教育部人文社科规划基金资助项目(16YJA630037)

第一作者: 张惠珍(1979-), 女, 副教授. 研究方向: 运筹学、智能优化. E-mail: zhzywz@163.com

for the MTVRPSPD is feasible and effective in real application. The solutions obtained by using the basic ACO, QACO and other state-of-art algorithms presented in literatures were compared, and the conclusion shows that the distribution routes, obtained by using the QACO to solve the linear integer programming model for MTVRPSPD, are better in terms of economic efficiency and reasonability.

Keywords: vehicle routing problem; simultaneous deliveries and pickups; multi-trip; ant colony algorithm; quantum computing

1 研究背景

同时送取货物的车辆路径问题(vehicle routing problem with simultaneous deliveries and pickups, VRPSDP)最早由 Min^[1]于 1989 年提出, Dethloff^[2]于 2001 年首次从逆向物流的角度研究了此问题. VRPSDP 是逆向物流车辆路径问题(vehicle routing problem, VRP)的一个重要组成部分, 与其并列的还有先送货后取货的车辆路径问题(vehicle routing problem with backhauls, VRPB), 以及混合送货和取货的车辆路径问题 VRPBM (vehicle routing problem with backhauls of mixed loads, VRPBM)^[3].

由于 VRPSDP 将正向物流和逆向物流同时加以考虑, 更接近于现实生活中的物流配送服务, 对实际问题更有指导意义, 近年来越来越多的学者对 VRPSDP 展开了研究^[3-9], 其研究成果可被分为两大部分: a. 对基本 VRPSDP 进行改进, 提出了多种 VRPSDP 的变种问题, 如带时间窗的 VRPSDP^[4]、具有车辆最大行程约束的 VRPSDP^[5]、随机旅行时间的 VRPSDP^[3]、车辆容量可调节的 VRPSDP^[6]、异车型 VRPSDP 等^[7]. b. 设计并提出了多种求解 VRPSDP 的方法. 张建勇等^[8]结合 2-opt 法和等级替换策略设计了求解 VRPSDP 的一种混合遗传算法, 给出了该算法初始种群的两种生成规则, 阐述了违反约束条件的处理方法; Liu 等^[4]在求解带时间窗的 VRPSDP 时, 用遗传算法进行局部搜索, 用禁忌搜索进行路径分配的优化; 吴斌等^[9]提出了基于混沌理论的精英均值计算旋转角算法; Qu 等^[6]设计了两阶段启发式算法, 并将其用于车辆容量可调节的 VRPSDP.

自 VRPSDP 提出至今, 其模型的改进和求解算法虽然均已取得很大研究成果, 但大部分研究仍存在一些不足, 可归纳为两方面. 一方面, 绝大多数 VRPSDP 模型假设在计划期内(一般为一天)配送车

辆只能运行一次, 考虑多车次的 VRPSDP 模型在现有文献中极其罕见. 然而, 现实生活中, 物流公司为了节约配送成本和完成配送任务, 通常会启用较少的配送车辆, 同一辆车会多次往返于配送中心, 完成多条配送路线的配送任务. 另一方面, 绝大多数 VRPSDP 模型没有考虑货物的装卸时间和车辆的工作时间.

本文研究多车次同时送取货的车辆路径问题 (multi-trip vehicle routing problem with simultaneous deliveries and pickups, MTVRPSPD), 并在该问题中考虑了货物的装卸时间、车辆的工作时间和载重量限制. 与基本 VRPSDP 相比, 本文考虑的 MTVRPSPD 存在以下几个难点: 第一, 多车次的引入, 增加了问题的复杂度; 第二, 对车辆工作时间的限制, 加强了问题的约束. 当车辆工作时间接近最大限制时, 即使车辆的载重量仍然可以满足某些客户的需求, 但也必须返回配送中心. 对车辆工作时间的限制不仅会导致车辆总行程的增加, 而且会导致车辆数和配送路径数的增加, 使问题的优化难度加大^[3].

为了求解 MTVRPSPD, 本文在量子计算和蚁群算法的基础上, 吸收这两种方法的长处和优势, 克服它们的短处和缺陷, 进而提出求解该问题的量子蚁群算法. 本文的第二部分建立了 MTVRPSPD 问题的数学模型; 第三部分介绍了量子蚁群算法, 以及求解 MTVRPSPD 的基本流程; 第四部分进行了实例仿真, 采用量子蚁群算法求解 MTVRPSPD 问题, 获得了满意的结果.

2 MTVRPSPD 描述及其数学模型

2.1 问题描述

本文研究的 MTVRPSPD 是在所有客户点的位置和货物需求均已知的情况下, 完成所有客户的配送和取货服务, 并要求配送中心启用的配送车辆成

本和完成配送服务的行驶成本之和达到最小. 假设每个客户点可以同时收货和发货,也可以只收货或只发货;每辆车可以在一个工作日内为多条配送路线上的客户提供送货和取货服务. 此外,配送车辆和客户点还必须同时满足如下条件:

a. 服务约束,每辆车可以服务多个客户,但一个客户仅能由一辆车服务;

b. 配送中心约束,所有车辆由单一配送中心出发,配送完路径上所有的客户后返回配送中心;

c. 装载量约束,每条配送路径上车辆的装载量不能超过其最大载重量限制;

d. 最大工作时间限制,每辆车的工作时间(行驶时间和装卸货时间之和)不能超过其最大工作时间.

2.2 符号定义与数学模型

为了更好地描述 MTVRPSDP 的数学模型,对本文使用的符号作如下说明: m 为可供使用的最大车辆数; K 为运输车辆集合, $K = \{1, 2, \dots, m\}$; n 为客户总数; V_0 为所有客户的集合, $V_0 = \{1, 2, \dots, n\}$; V 为所有顶点的集合, $V = \{0\} \cup V_0$, 其中 0 代表配送中心; e_{ij} 为顶点 i 和顶点 j 之间的边; E 为所有边的集合, $E = \{e_{ij} = (i, j) | i, j \in V, i \neq j\}$; c_{ij} 为边 e_{ij} 的长度或车辆从顶点 i 到顶点 j 的旅行距离; v_k 为车辆 k 的平均行驶速度; t_{ijk} 为车辆 k 在边 e_{ij} 上的旅行时间, $t_{ijk} = \frac{c_{ij}}{v_k}$; f_k 为车辆 k 行驶单位距离的运行成本; h_k 为启用车辆 k 的成本; T_k 为车辆 k 在计划期内(一般为一天)的最大工作时间; G_k 为车辆 k 的最大负载能力; d_i 为客户 i 的送货需求量; g_i 为客户 i 的取货需求量; q_{ijk} 为车辆 k 访问完客户 i 后,紧接着访问客户 j 之前的负载量; S 为某条配送路线上的客户集, $S \subseteq V_0$.

决策变量:

$$x_{ijk} = \begin{cases} 1, & \text{车辆 } k \text{ 从客户 } i \text{ 到客户 } j, \text{ 且 } i \neq j; \\ 0, & \text{其他.} \end{cases}$$

由决策变量 x_{ijk} 的定义可知:若车辆 k ($\forall k \in K$) 未被启用执行配送和取货任务,则 $\sum_{j=1}^n x_{0jk} = \sum_{j=1}^n x_{j0k} =$

0; 否则, $\sum_{j=1}^n x_{0jk} = \sum_{j=1}^n x_{j0k} \geq 1$. 因此,车辆 k ($\forall k \in$

K) 的启用成本由 $h_k \min \left\{ 1, \sum_{j=1}^n x_{0jk} \right\}$ 决定.

MTVRPSDP 的目标是在满足上述约束条件的

基础上,确定完成所有送货和取货任务且期望成本最小的车辆路径. 可将 MTVRPSDP 表述为如下数学模型:

目标函数

$$\min z = \sum_{k=1}^m \left(h_k \min \left\{ 1, \sum_{j=1}^n x_{0jk} \right\} \right) + \sum_{k=1}^m \left(f_k \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ijk} \right) \quad (1)$$

约束条件

$$\sum_{k=1}^m \sum_{i=0}^n x_{ijk} = 1, \quad j \in V_0 \quad (2)$$

$$\sum_{i=0}^n x_{ijk} - \sum_{i=0}^n x_{jik} = 0, \quad j \in V_0, k \in K \quad (3)$$

$$\sum_{j=1}^n x_{0jk} = \sum_{j=1}^n x_{j0k}, \quad k \in K \quad (4)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1, \quad S \subseteq V_0, k \in K \quad (5)$$

$$\sum_{k=1}^m \sum_{i=0}^n q_{ijk} x_{ijk} - d_j = \sum_{k=1}^m \sum_{i=0}^n q_{jik} x_{jik} - g_j, \quad j \in V_0 \quad (6)$$

$$0 \leq q_{ijk} x_{ijk} \leq G_k, \quad i \in V, j \in V, k \in K \quad (7)$$

$$2st_{k0} \sum_{j=1}^n x_{0jk} + \sum_{i=0}^n \sum_{j=0}^n t_{ijk} x_{ijk} + \sum_{i=0}^n \sum_{j=1}^n st_{kj} x_{ijk} \leq T_k, \quad k \in K \quad (8)$$

$$x_{ijk} \in \{0, 1\}, \quad i \in V, j \in V, k \in K \quad (9)$$

其中:目标函数式(1)表示最小化所有车辆的启动成本与行驶路径成本之和;约束式(2)保证每个顾客都要被服务一次且仅被服务一次;式(3)保证一辆车服务客户 j , 则离开该客户的仍然是这辆车,即客户 j 只被同一辆车服务;式(4)保证每辆车从配送中心出发并最终返回到配送中心;式(5)排除子回路;式(6)表示车辆在经过客户 j 前后的载货量变化情况;式(7)保证每辆车在任意一条配送路段上的货物运输量不超过其最大载重量,且不为负;式(8)保证车辆 k 在计划期内的总工作时间不超过其最大工作时间 T_k , st_{ki} 表示车辆 k 在节点 i 的装卸时间;式(9)是决策变量 x_{ijk} 的取值范围.

3 MTVRPSDP 的量子蚁群算法

蚁群算法(ant colony optimization, ACO)是意大利学者 Dorigo 等^[10]通过模拟蚁群觅食行为提出的一种基于种群的模拟进化算法,其在求解组合优

化难题和连续优化问题中均取得了较好的结果.近年来,越来越多的学者利用ACO求解物流配送中的车辆调度和路径优化问题^[11].相比于其他智能优化算法,ACO具有较强鲁棒性、易与其他方法结合的优点.首先,ACO对初始路线的要求不高,即ACO的求解结果不依赖于初始路线的选择,而且在搜索过程中不需要进行人工调整;其次,ACO的参数数目少,设置简单.但是,ACO也有一些不足之处,如:容易陷入局部最优,搜索速度比较慢等.

蚂蚁状态的转移和路径上信息素的更新是ACO的重要组成部分.本文针对基本ACO收敛速度慢和易陷入局部最优等缺点,将量子进化算法中的量子比特和量子旋转门引入ACO,用以改进基本ACO算法中的状态转移概率,有效抑制算法的早熟现象,并加快算法的求解速度.本文将改进后的ACO算法称为量子蚁群算法(quantum ant colony optimization, QACO),并将其用于求解MTVRPSDP.

3.1 量子编码特性

量子计算的基本原理是使量子态所处叠加态的各个基态间的相对相位和概率幅度发生改变,使各个基态的发生概率产生变化,从而使其叠加态发生变化,具有并行、叠加、干涉等特征^[12].量子比特是量子计算中基本信息的存储单元,采用 $|0\rangle$ 和 $|1\rangle$ (“ $| \rangle$ ”称为Dirac记号,在量子力学中表示状态)来表示微观粒子的两种基本状态.单量子比特的任意状态都可以表示为这两个基本状态的线性组合,即除 $|0\rangle$ 和 $|1\rangle$ 之外,还可以是这两种状态之间的中间状态(叠加态): $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ (α 和 β 是一对复数,表示量子态的概率幅,即量子态 $|\varphi\rangle$ 以 $|\alpha|^2$ 的概率坍缩到 $|0\rangle$ 或以 $|\beta|^2$ 的概率坍缩到 $|1\rangle$,且满足 $|\alpha|^2 + |\beta|^2 = 1$).

一个量子比特同时可包含0和1的信息,那么一个长度为 m 的量子位可以表示 2^m 种不同的状态,有 m 个量子位的个体 j 的概率幅可表示为

$$p_j = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_m \\ \beta_1 & \beta_2 & \cdots & \beta_m \end{bmatrix} \quad (10)$$

其中, $|\alpha_i|^2 + |\beta_i|^2 = 1 (i=1,2,\dots,m)$.

在QACO中用量子比特来表示量子信息素,设种群大小为 n ,其量子信息素用量子位表示为 $p = (p_1, p_2, \dots, p_n)$,其中 $p_j (j=1,2,\dots,n)$ 如式(10)所示.

3.2 量子旋转门

量子比特相位的改变可以用量子旋转门实现,

量子旋转门的调整方式可定义为:

$$\begin{bmatrix} \alpha_i^{t+1} \\ \beta_i^{t+1} \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} \alpha_i^t \\ \beta_i^t \end{bmatrix} \quad (11)$$

式中: $[\alpha_i^t, \beta_i^t]^T (i=1,2,\dots,m)$ 为第 t 次迭代的第 t 个量子位的概率幅; θ_i 为第 i 个量子位的旋转角度,旋转角 θ_i 的大小直接影响算法的收敛速度和寻优能力.如果旋转角 θ_i 的幅值过大,会导致早熟;如果幅值过小,会减慢收敛速度.旋转角的大小和方向的确定是根据事先规定的调整策略,通过查表法得到^[13].

本文针对MTVRPSDP的求解和借鉴文献^[12]中量子旋转角的取值方法,设计了一种改进的 θ_i 取值方法,该方法能够根据当前量子比特概率幅和MTVRPSDP问题的目标函数值计算出相位的旋转角度,从而实现蚁群算法中种群的多样性,避免算法早熟收敛. θ_i 的取值定义如下:

$$\theta_i = s(\alpha_i \beta_i) (\theta_0 + \text{sign}(f(x_i) - f(b_i)) 0.07\pi) \quad (12)$$

式中, $s(\alpha_i \beta_i)$ 控制旋转角的旋转方向,若 $\alpha_i \beta_i \geq 0$, $s(\alpha_i \beta_i) = 1$,否则, $s(\alpha_i \beta_i) = -1$. $\text{sign}(f(x_i) - f(b_i))$ 为符号函数,自适应地调整旋转角的大小.当 $f(x_i) - f(b_i) > 0$ 时, $\text{sign}(f(x_i) - f(b_i)) = 1$;当 $f(x_i) - f(b_i) < 0$ 时, $\text{sign}(f(x_i) - f(b_i)) = -1$;当 $f(x_i) - f(b_i) = 0$ 时,则 $\text{sign}(f(x_i) - f(b_i)) = 0$. θ_0 为初始旋转角度, x_i 表示当前解, b_i 表示当前最优解, $f(x)$ 为目标函数.

3.3 状态转移规则

记蚂蚁 k 由客户点 i 出发访问下一个客户 j 的概率为 P_{ij}^k ,其定义为

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\sigma [\eta_{ij}]^\beta [u_j]^\mu [r_j]^\lambda [w_{ij}]^\xi}{\sum_{l \in N_i^k} [\tau_{il}]^\sigma [\eta_{il}]^\beta [u_l]^\mu [r_l]^\lambda [w_{il}]^\xi}, & j \in N_i^k \\ 0, & \text{其他} \end{cases} \quad (13)$$

式中: N_i^k 为蚂蚁 k 位于客户 i 时可选择的客户集合; τ_{ij} 为边 (i,j) 上的蚂蚁信息素轨迹浓度; $\eta_{ij} = \frac{1}{c_{ij}}$ 为边 (i,j) 的能见度,反映了蚂蚁对下一个客户的

预见性; $u_j = \frac{1}{|\alpha|^2}$ 为客户点 j 的量子信息强度, $|\alpha|^2$ 表示第 j 个量子位的量子坍缩到 $|0\rangle$ 的概率; $r_j = |d_j - g_j|$ 为客户点 j 的送取货物数量差值的绝对值,其值越大,越容易被选择; $w_{ij} = c_{i0} + c_{j0} - c_{ij}$ 为将客户 i 和客户 j 直接连接和两点分别与配送中心连接产生的节约量; $\alpha \geq 0, \beta \geq 0, \mu \geq 0, \lambda \geq 0$ 和

$\xi \geq 0$ 均为权衡因子,分别决定了 τ_{ij} , η_{ij} , u_j , r_j 和 w_{ij} 的相对重要性。

蚂蚁 k 由客户点 i 出发,采用伪随机规则确定所要访问的下一个客户:若 $q > q_0$ ($q_0 \in (0, 1)$ 是一个常数; q 为 $(0, 1)$ 区间的随机数),根据式(13)计算出的转移概率并按照轮盘赌的方法确定下一个访问的客户;否则,取转移概率最大的客户 j 为下一个访问的客户。

3.4 信息素更新

本文利用式(14)~(16)对客户 i 和 j 路径上的信息素进行更新:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \Delta\tau_{ij} \quad (14)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (15)$$

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k, & \text{蚂蚁 } k \text{ 在行驶中经过边}(i, j) \\ 0, & \text{其他} \end{cases} \quad (16)$$

式中: ρ ($0 < \rho < 1$) 为信息素挥发系数; $\Delta\tau_{ij}$ 为所有蚂蚁完成一次周游后,边 (i, j) 上的信息素变化量; $\Delta\tau_{ij}^k$ 为蚂蚁 k 在边 (i, j) 上释放的信息素量; Q 为所有蚂蚁完成一次周游后所释放的信息素总量; L_k 为蚂蚁 k 在本次周游中所经过的路径长度。

3.5 算法步骤

步骤 1 初始化各参数. 设定参数 $\alpha, \beta, \gamma, \lambda, \xi$ 等的值, 初始迭代次数 $t = 1$, 最大迭代次数为 $t_{\max}$; 初始化量子种群 Num , 量子位数目 m , 种群 $p(t) = (p_1^t, p_2^t, \dots, p_n^t)$, 其中, p_j^t ($j = 1, 2, \dots, n$) 为种群中第 t 次迭代的第 j 个个体:

$$p_j^t = \begin{bmatrix} \alpha_1^t & \alpha_2^t & \dots & \alpha_m^t \\ \beta_1^t & \beta_2^t & \dots & \beta_m^t \end{bmatrix}$$

式中: $t = 1$ 时, 所有的 α_i 和 β_i ($i = 1, 2, \dots, m$) 取值为 $1/\sqrt{2}$; 初始化信息素浓度 τ_{ij} 和信息素浓度增量 $\Delta\tau_{ij} = 0$; 初始化各蚂蚁已走的客户集 $tabu_k$, 以及候选客户集 N_i^k (包括配送中心); 初始化所有车辆已工作的时间为 0; 设置完成任务的蚂蚁数量 $k = 0$ 。

步骤 2 Num 只蚂蚁从配送中心出发, 完成配送路径的构建. 对蚂蚁 k , 按式(13)计算其转移概率, 并按照状态转移规则确定下一个服务的客户 j :
a. 若客户 j 满足配送车辆的载重量和工作时间的限制, 则将客户 j 加入蚂蚁 k 的禁忌表 $tabu_k$, 并更新 N_i^k ; b. 若客户 j 不满足车辆的载重量限制, 但满足其工作时间限制, 则车辆返回配送中心, 启动下一配送路线的配送任务; c. 若客户 j 满足车辆的载重量限制, 但不满足其工作时间限制, 则车辆返回配送中心, 配送中心启动下一辆车完成后续配送任务。

步骤 3 所有客户服务结束后, 蚂蚁 k 完成一次周游, 得到问题的一个可行解. 更新蚂蚁数 $k = k + 1$, 若 $k \leq Num$, 转步骤 2; 否则, 转步骤 4。

步骤 4 计算各蚂蚁的目标函数值 z_k ($k = 1, 2, \dots, Num$), 并记录当前最满意的解。

步骤 5 根据信息素更新规则, 利用式(14)~(16)对信息素进行更新。

步骤 6 应用量子旋转门对所有客户点的量子信息进行更新。

步骤 7 对各边 (i, j) 置 $\Delta\tau_{ij}^k = 0, t = t + 1$ 。

步骤 8 若迭代次数 $t < t_{\max}$, 则转步骤 2。

步骤 9 输出当前最优解。

4 算例分析

本文采用文献[14]的算例测试 QACO 的优化性能和 MTRVPSDP 模型(1)~(9)的可行性与实用性. 该算例为: 某第三方物流企业需要利用一定数量的车辆和配送人员为 100 个客户提供同时送货和取货的服务, 配送中心(用 0 表示)的位置坐标为 (34, 36) (单位 km), 配送中心的送货需求量 d_i 和取货需求量 g_i 均为 0 t; 100 个客户的位置坐标, 送货需求量和取货需求量如表 1 所示(见下页); 配送中心提供的配送车辆的车型相同, 平均运行速度均为 35 km/h, 最大装载容量均为 10 t, 最长工作时间均为 8 h。

利用 Matlab R2010a 对基本 ACO 和 QACO 算法进行编程实现, 2 种算法在 Intel Core i5 3.40 GHz, 4 GB 内存, 操作系统为 Windows7 的环境下运行。

为了确保算法之间的可比性, 使基本 ACO 和 QACO 在相同的起点上进行比较, 根据文献[14]对 ACO 和 QACO 的共同参数设定了相同的值: 蚂蚁数 $Num = 60$, 距离成本系数 $f_k = 10$ 元/km, 车辆启动成本系数 $h_k = 2\ 500$ 元, 客户点的装卸时间 $st_i = 0.1$ h, 配送中心的装货(卸货)时间 $st_0 = 0.1$ h, 信息素轨迹的相对重要性 $\alpha = 1$, 能见度的相对重要性 $\beta = 1$, 量子信息强度的相对重要性 $\mu = 2$, 送取货物差值的重要性 $\lambda = 2$, 节省量的重要性 $\xi = 1.5$, 信息素总量 $Q = 15$, 算法迭代次数 $t_{\max} = 200$. 量子蚁群算法中的初始旋转角度 $\theta_0 = 0.06\pi$. QACO 算法和基本 ACO 算法各随机运行 10 次, 求解结果如表 2 所示(见下页). 基本 ACO 算法求出的车辆数为 5, 车辆行驶成本为 12 047 元, 总成本为 24 547 元. QACO 算法求出的车辆数也为 5, 车辆行驶成本为 9 285.4

元,总成本为21 785.4元.QACO求解结果的标准差为0.67,小于ACO求解结果的标准差为2.29,这说明QACO的求解性能比ACO的求解性能稳定.虽然QACO与ACO求得相同的车辆数,但QACO求解的平均成本和总成本分别为22 129.8元和21 785.4

元,分别小于ACO求得的平均成本24 984元和总成本24 547元.这说明本文设计的QACO算法求解MTVRPSDP的求解结果优于基本ACO算法的求解结果.

表1 客户数据
Tab.1 Data of clients

客户编号	横纵坐标/km	退货量/t	送货量/t	客户编号	横纵坐标/km	退货量/t	送货量/t	客户编号	横纵坐标/km	退货量/t	送货量/t
1	42,7	0.08	0.42	35	65,20	0.18	0.42	69	21,24	2.45	0.35
2	24,12	0.25	0.25	36	45,30	1.13	0.57	70	17,34	0.15	0.15
3	23,3	0.09	0.61	37	35,40	1.40	0.20	71	12,24	0.65	0.65
4	11,14	1.41	0.39	38	41,37	1.44	0.16	72	24,58	0.79	1.11
5	6,38	0.25	1.35	39	64,42	0.59	0.31	73	27,69	0.39	0.61
6	2,48	0	0.10	40	40,60	1.40	0.70	74	15,77	0.18	0.72
7	8,56	0.39	2.31	41	31,52	1.61	1.09	75	62,77	1.61	0.39
8	13,52	0.90	2.70	42	35,69	1.17	1.13	76	49,73	1.68	0.82
9	6,68	0.26	2.74	43	53,52	1.08	0.02	77	67,5	0.19	2.31
10	47,47	1.30	0	44	65,55	1.18	0.22	78	56,39	2.51	1.09
11	41,49	0.84	0.16	45	63,65	0.78	0.02	79	37,47	0.47	0.13
12	35,17	0.34	0.36	46	2,60	0.02	0.48	80	37,56	0.33	0.17
13	55,45	1.06	0.24	47	20,20	0.80	0	81	57,68	0.26	0.24
14	55,20	0.69	1.21	48	5,5	1.60	0	82	47,16	0.85	1.65
15	15,30	1.30	1.30	49	60,12	0.62	2.48	83	44,17	0.35	0.55
16	25,30	0.25	0.05	50	40,25	0.56	0.34	84	46,13	0.23	0.57
17	20,50	0.20	0.30	51	15,19	0.08	0.02	85	49,11	0.40	1.40
18	10,43	0.21	0.69	52	22,22	0.20	0	86	49,42	1.11	0.19
19	55,60	1.47	0.13	53	18,24	1.65	0.55	87	53,43	1.14	0.26
20	30,60	0.80	0.80	54	26,27	2.60	0.10	88	61,52	0.26	0.04
21	20,65	0.37	0.83	55	25,24	1.92	0.08	89	57,48	1.94	0.36
22	50,35	1.33	0.57	56	22,27	0.90	0.20	90	56,37	0.40	0.20
23	30,25	1.92	0.38	57	25,21	1.01	0.19	91	55,54	2.55	0.05
24	15,10	1.33	0.67	58	19,21	0.90	0.10	92	15,47	0.51	1.09
25	30,5	0.13	0.67	59	20,26	0.69	0.21	93	14,37	0.42	0.68
26	10,20	0.95	0.95	60	18,18	1.70	0	94	11,31	0.25	0.45
27	5,30	0.03	0.17	61	49,58	0.84	0.16	95	16,22	2.98	1.12
28	20,40	0.60	0.60	62	27,43	0.57	0.33	96	4,18	0.78	2.72
29	15,60	0.43	1.28	63	37,31	1.17	0.23	97	28,18	1.67	0.93
30	45,65	0.62	0.28	64	57,29	0.92	0.88	98	26,52	0.45	0.45
31	45,20	0.49	0.61	65	63,23	0.07	0.13	99	26,35	1.11	0.39
32	45,10	0.40	1.40	66	53,12	0.14	0.46	100	31,67	0.14	0.16
33	55,5	0.26	2.64	67	32,12	0.26	0.44				
34	65,35	0.16	0.14	68	36,26	1.30	0.50				

表2 量子蚁群算法与蚁群算法效果比较

Tab.2 Comparison of the quantum-inspired ant colony algorithm with the basic ant colony algorithm

算法	成本均值/元	平均计算时间/s	标准差	满意解			
				总成本/元	车辆数	运行距离/km	行驶成本/元
ACO	24 984.0	206	2.29	24 547.0	5	1 204.7	12 047.0
QACO	22 129.8	224	0.67	21 785.4	5	928.54	9 285.4

图1为ACO和QACO算法求解满意解的总成本收敛图.由图1可知,ACO算法和QACO算法迭代100次已基本趋于收敛.虽然算法迭代开始时,ACO的计算结果优于QACO的计算结果,但迭代大约50次以后QACO的计算结果优于ACO的计算结果.此外,由于QACO中转移概率的计算和客户点量子信息的计算比ACO中的计算过程繁杂,导致QACO迭代200次的总计算时间224 s多于ACO的计算时间206 s,但QACO收敛到满意解的速度比ACO收敛到满意解的速度快.

表3给出了本文和文献[14]利用不同的模型和算法求解该算例的结果.文献[14]所建立的MTVRPSDP模型为二次规划模型,所采用的求解方法是允许不可行解的禁忌搜索法,模型和算法均与本文的模型和算法不同.将本文计算结果与文献[14]的计算结果进行对比,发现:虽然本文QACO求解式(1)~(9)获得的车辆数与文献[14]中利用允许不可行解的禁忌搜索法求解得到的车辆数相同,均为5辆车;但是,文献[14]共有10条配送路径,每辆车完成2条路线的配送任务,而本文算法求得的解共有12条配送路线,有2辆车的配送路线数为3,其余3辆车的配送路线数为2,不仅完成送货和取货

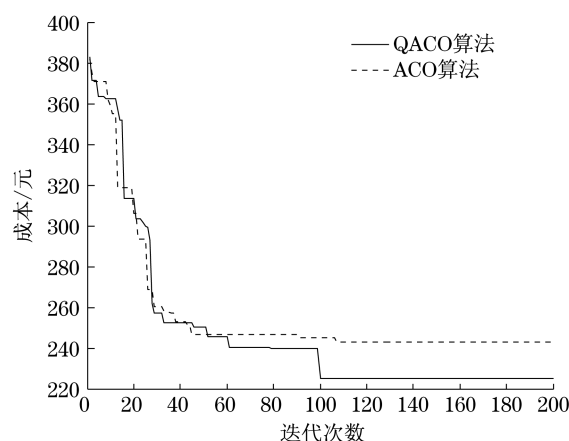


图1 QACO和ACO迭代趋势对比

Fig.1 Comparison between the iteration convergences of QACO and ACO

任务的总行驶距离928.54 km优于文献[14]的总行驶距离940.66 km,而且车辆工作总时间37.16 h优于文献[14]的工作总时间38.88 h.本文所计算的结果提高了车辆在单位时间内完成送货和取货的任务量,即提高了车辆的服务效率.这说明本文所建立的MTVRPSDP线性整数规划模型和设计的QACO算法在实际应用中均是可行的,且有一定的优越性,能够为所求解的问题找到较好的满意解.

表3 本文计算结果与文献[14]计算结果比较

Tab.3 Comparison of results obtained in the paper with the ones shown in literature [14]

比较	算法	模型	满意解					
			总成本/元	车辆数	配送路径数	总行驶距离/km	行驶成本/元	总工作时间/h
本文	量子蚁群算法	线性整数规划模型	21 785.4	5	12	928.54	9 285.4	37.16
文献[14]	允许不可行解的禁忌搜索算法	二次规划模型	21 906.6	5	10	940.66	9 406.6	38.88

QACO求得的配送路线图如图2所示(见下页),共12条配送路线.车辆1为2条配送路线的客户提供服务,2条配送路线分别为:0-2-24-4-26-96-48-51-95-53-58-59-0;0-99-28-70-93-94-27-5-7-8-0;车辆2为2条配送路线的客户提供服务,2条配送路线分别为:0-79-1-32-84-85-66-49-77-0,0-62-17-

20-40-30-76-81-75-45-44-88-80-11-0;车辆3为3条配送路线的客户提供服务,3条配送路线分别为:0-16-54-56-52-57-97-67-12-0,0-14-31-83-0,0-63-68-23-55-69-15-18-92-0;车辆4为3条配送路线的客户提供服务,3条配送路线分别为:0-36-22-78-90-86-0,0-41-98-72-21-73-100-42-0,

0-37-3-25-47-60-71-0; 车辆5为2条配送路线的客户提供服务, 2条配送路线分别为: 0-87-6-46-9-74-29-50-0, 0-38-10-61-19-91-43-89-13-39-34-64-65-35-0.

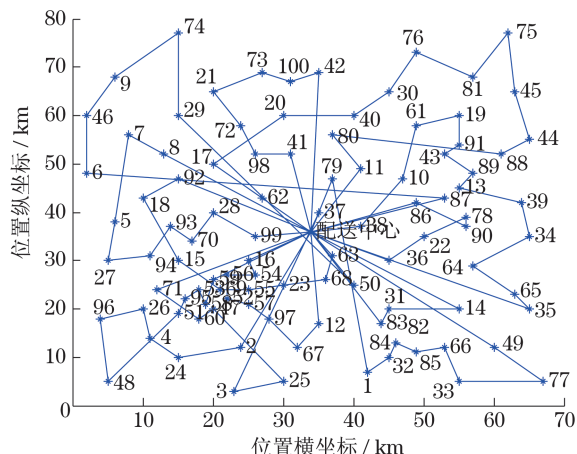


图2 配送方案路径图

Fig.2 Distribution routes

5 结论

MTVRPSDP 的研究对物流公司的配送服务具有十分重要的现实意义. 根据现实生活中配送车辆具有工作时间和载重量限制的特点, 本文建立了更符合实际应用且考虑了装卸货物时间的 MTVRPSDP 线性整数规划模型. 然后, 将量子计算中的量子比特和量子旋转门引入 ACO, 在每个客户点设置了量子信息, 用其改进蚂蚁的状态转移规则, 设计了求解该问题的 QACO 算法. 求解测试算例表明: 本文所设计的 MTVRPSDP 线性整数规划模型在实际应用中是可行和有效的; 此外, 本文所设计的 QACO 算法能够有效求解 MTVRPSDP 的线性整数规划模型, 具有较高的稳定性.

本文所设计的 MTVRPSDP 模型并未考虑客户对送货和取货服务的时间窗的限制和满意度, 也未将物流公司完成配送和取货任务所启动的车辆数、行驶路径总成本作为优化的多个目标. 建立并求解带有时间窗限制的多目标多车次同时取送货物的车辆路径问题模型将是作者下一步的研究工作.

参考文献:

[1] MIN H. The multiple vehicle routing problem with simultaneous delivery and pick-up points [J]. Transportation Research Part A: General, 1989, 23

(5): 377-386.

- [2] DETHLOFF J. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up[J]. OR-Spektrum, 2001, 23(1): 79-96.
- [3] 张涛, 余焯娅, 刘岚, 等. 同时送取货的随机旅行时间车辆路径问题方法[J]. 系统工程理论与实践, 2011, 31(10): 1912-1920.
- [4] LIU R, XIE X L, AUGUSTO V, et al. Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care[J]. European Journal of Operational Research, 2013, 230(3): 475-486.
- [5] MONTANÉ F A T, GALVÃO R D. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service [J]. Computers & Operations Research, 2006, 33(3): 595-619.
- [6] QU Y, BARD J F. The heterogeneous pickup and delivery problem with configurable vehicle capacity [J]. Transportation Research Part C: Emerging Technologies, 2013, 32: 1-20.
- [7] 田宇, 伍炜勤. 求解异车型同时集送问题的多属性标签算法[J]. 系统工程理论与实践, 2015, 35(1): 183-190.
- [8] 张建勇, 李军. 具有同时配送和回收需求的车辆路径问题的混合遗传算法[J]. 中国公路学报, 2006, 19(4): 118-122.
- [9] 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.
- [10] DORIGO M, BONABEAU E, THERAULAZ G. Ant algorithms and stigmergy [J]. Future Generation Computer Systems, 2000, 16(8): 851-871.
- [11] 李娅, 王东. 基于混沌扰动和邻域交换的蚁群算法求解车辆路径问题[J]. 计算机应用, 2012, 32(2): 444-447.
- [12] 何小锋, 马良. 带时间窗车辆路径问题的量子蚁群算法[J]. 系统工程理论与实践, 2013, 33(5): 1255-1261.
- [13] LI B B, WANG L. A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2007, 37(3): 576-591.
- [14] 李建, 达庆利, 何瑞银. 多车次同时集散货物路线问题研究[J]. 管理科学学报, 2010, 13(10): 1-7.

(编辑: 丁红艺)