

一种带时间窗车辆路径问题的混合蚁群算法*

黄震, 罗中良, 黄时慰

(惠州学院 计算机科学系, 广东惠州, 516007)

摘要: 针对带时间窗车辆路径问题求解时蚁群算法存在容易陷入局部最优, 而遗传算法初始种群的优劣对算法有效性存在直接影响, 提出一种混合蚁群优化算法。算法首先在蚁群算法的节点选择概率公式中引入时间窗因素, 以得到初始种群, 然后通过遗传算法的交叉算子和变异算子对初始种群中的较优路径进行交叉和变异操作, 从而得到更优的路径。通过 Matlab 环境下对文中混合算法进行仿真实验, 在车辆利用率和路径规划上效果明显, 表明了算法的高效性, 同时混合算法可以避免陷入局部最优。

关键词: 蚁群算法; 遗传算法; 车辆路径问题; 时间窗

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 0529-6579 (2015) 01-0041-06

Application Research of Hybrid ant Colony Algorithm in Vehicle Routing Problem with Time Windows

HUANG Zhen, LUO Zhongliang, HUANG Shiwei

(Department of Computer Science, Huizhou University, Huizhou 516007, China)

Abstract: A hybrid ant colony algorithm was proposed. Because, ant colony algorithm used to solve the vehicle routing problem with time windows (VRPTW) is easy to fall into local optimum, and the quality of initial population in genetic algorithm affects the effectiveness of the algorithm directly. Firstly, the algorithm introduces the factors of time windows into node selection probability formula of ant colony algorithm to get the initial population. Secondly, the crossover and the mutation were operated to get a better path for the initial population. Applying Matlab environment for hybrid algorithm simulation, the effects on the vehicle utilization and path planning is obvious. It shows the algorithm is efficient, and can avoid falling into local optimum.

Key words: ant colony algorithm; genetic algorithm; vehicle routing problem; time window

车辆路径问题 (Vehicle Routing Problem, VRP) 是一类经典的组合优化问题。一般指对一系列的客户点组织适当的行车路线, 使车辆有序地通过它们, 在满足一定的约束条件 (如货物需求量、车辆容量限制等) 下, 达到一定的目标 (如距离

最短、费用最少等), 带时间窗车辆路径问题 (Vehicle Routing Problem with Time Windows, VRPTW) 是在 VRP 的基础上要求在配送过程中按照客户要求在一定的时间窗内到达客户点, 即在不违背车辆容量限制、时间限制等约束条件的前提

* 收稿日期: 2014-06-22

基金项目: 广东省科技计划资助项目 (2012B010100038); 广东省高等学校教学质量与改革工程本科类资助项目 (粤高教函【2013】113号-113)、惠州市科技计划资助项目 (A512.0234)、全国大学生创新训练资助项目 (105771300)

作者简介: 黄震 (1980年生), 男; **研究方向:** 智能算法、无线传感器网络; **通讯作者:** 罗中良; E-mail: 195146501@qq.com

下,合理制定运输时的车辆配送路径方案,以尽可能小的成本满足处在不同地理位置的客户对货物运送和服务时间的要求^[1]。由于 VRPTW 带有服务时间的访问限制,VRPTW 比 VRP 更贴近实际应用,广泛地应用于交通和物流领域,由于 VRPTW 是 NP 难问题,主要都是使用启发式算法^[2-3]来求解,已有不少学者对该问题进行了相关的研究^[4-11]。其中,李全亮^[4]提出利用蚂蚁算法求解 VRPTW,从数值计算上探索了蚂蚁算法的优化能力,获得了满意的效果;李琳等^[5]将蚁群系统与最大最小蚂蚁系统相结合,并在状态转移规则中引入时间窗跨度与服务等待时间因素;何小锋等^[6]提出一种量子蚁群算法,增加量子比特启发因子、利用量子旋转门实现信息素更新,实验表明该算法求解 VRPTW 比蚁群算法效率更高;吴天羿等^[7]在初始种群的产生、交叉、变异等方面对遗传算法进行改进,在求解 VRPTW 中比遗传算法更有效。

本文首先使用蚁群算法求解 VRPTW,实验结果发现蚁群算法存在容易陷入局部最优的缺点,于是将蚁群算法和遗传算法相结合,通过蚁群算法产生初始种群,然后进行交叉和变异操作,实验结果表明混合算法不仅可以避免陷入局部最优,而且具有比较快的收敛速度和良好的全局寻优能力,能够有效的求解 VRPTW。

1 VRPTW 的数学模型

VRPTW 描述为^[12]:某仓库有 k 辆车,每辆车的最大载重量为 w ,需要为 n 个客户配送货物,其中仓库编号为 0,客户编号为 1 至 n ,每辆车从仓库出发为客户送货,最终回到仓库, $q_i (i = 1, 2, \dots, n)$ 为各客户的货物需求量, at_i 表示车辆到达客户 i 的时间, wt_i 表示车辆在客户 i 的等待时间, st_i 表示车辆在客户 i 的服务时间, t_{ij} 表示车辆从客户 i 到客户 j 的行驶时间, $[e_i, l_i]$ 是客户 i 的时间窗,即客户 i 的最早开始服务时间为左时间窗 e_i ,最迟开始服务时间为右时间窗 l_i , C_{ij} 为车辆从客户 i 到客户 j 的配送成本。

引入决策变量 X_{ijk} , 其取值如下:

$$X_{ijk} = \begin{cases} 1 & \text{车辆 } k \text{ 由 } i \text{ 到 } j \\ 0 & \text{否则} \end{cases} \quad (1)$$

VRPTW 的目标函数表示为:

$$\min p1K + p2 \sum_{k=1}^K \sum_{i=0}^n \sum_{j=0}^n C_{ij} x_{ijk} \quad (2)$$

一般情况下 VRPTW 的目标有 2 个,第一目标是配送的车辆数 K 最少,第二目标是总配送距离最短,所以需要在式 (2) 中将 $p1$ 和 $p2$ 的值设置为 $p1 \gg$

$p2$ 。

VRPTW 的约束条件如下:

$$\sum_{k=1}^K \sum_{i=0}^n x_{ijk} = 1 \quad \forall j \in \{1, 2, \dots, n\} \quad (3)$$

$$\sum_{k=1}^K \sum_{j=0}^n x_{ijk} = 1 \quad \forall i \in \{1, 2, \dots, n\} \quad (4)$$

$$\sum_{k=1}^K \sum_{j=1}^n x_{0jk} = \sum_{k=1}^K \sum_{i=1}^n x_{i0k} = K \quad (5)$$

$$\sum_{i=0}^n (q_i \sum_{j=0}^n x_{ijk}) \leq w \quad \forall k \in \{1, 2, \dots, K\} \quad (6)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \geq 1, \forall S \subseteq \{1, 2, \dots, n\}, \quad (7)$$

$$|S| \geq 2, \forall k \in \{1, 2, \dots, K\}$$

$$at_i \leq l_i, \forall i \in \{1, 2, \dots, n\} \quad (8)$$

$$e_i \leq at_i + wt_i \leq l_i, \forall i \in \{1, 2, \dots, n\} \quad (9)$$

$$at_i + wt_i + st_i + t_{ij} + (1 - x_{ijk})p3 \leq at_j$$

$$\forall i, j \in \{1, 2, \dots, n\}, \forall k \in \{1, 2, \dots, K\} \quad (10)$$

$$wt_i = \max\{0, e_i - at_i\}, \forall i \in \{1, 2, \dots, n\} \quad (11)$$

$$at_i \geq 0, \forall i \in \{1, 2, \dots, n\} \quad (12)$$

$$wt_i \geq 0, \forall i \in \{1, 2, \dots, n\} \quad (13)$$

$$st_i \geq 0, \forall i \in \{1, 2, \dots, n\} \quad (14)$$

其中式 (3)、(4) 约束每个客户只由唯一一辆车服务;式 (5) 是对仓库约束,即由仓库出发和回到仓库的车辆都是 K 辆;式 (6) 车辆的载重约束;式 (7) 为次回路消除条件,式 (8) 至式 (14) 是时间窗约束。

2 混合蚁群算法求解 VRPTW

2.1 求解 VRPTW 的编码方式

本文在求解 VRPTW 时采用整数编码方式^[13],需要编码的包括初始种群的染色体个体和每个染色体对应的解路径,本文采用 0 表示仓库 (有 K 辆车),1 至 n 表示客户编号,具体编码方式如下:

1) 每个染色体个体采用一个 n 维整型向量 $[r_1, r_2, \dots, r_n]$ 表示, $r_i \in \{1, 2, \dots, n\}$,每个 r_i 对应一个客户。

2) 解路径采用一个 $n + K + 1$ 维的整型向量 $[t_1, t_2, \dots, t_{(n+K+1)}]$ 表示, $t_j \in \{0, 1, \dots, n\}$,每个 t_j 对应仓库或客户。

每个染色体个体对应一个解路径,由染色体个体求解路径的基本原理是求出 K 条子路径,每条子路径对应一辆车的行驶回路,具体求解方法如下:

1) 将解路径向量的第一维分量 t_1 赋值为 0, 表示车辆由仓库出发;

2) 定义变量 Q (初始值为 0), 将染色体个体中的客户 r_1 的货物需求量加到 Q 中, 如果 Q 不超过车辆的最大载重量则计算 at_{r_1} (即从仓库到达客户 r_1 的时间), 如果 at_{r_1} 没有超过客户 r_1 的时间窗则将 r_1 加入到解向量中 (即将 r_1 赋值给解路径向量的第二维 t_2);

3) 按步骤 2 类似的方法继续计算染色体个体向量中 r_1 后面的每一个客户 r_i , 如果满足重量约束和时间窗约束则将 r_i 加入到解向量中, 如果 Q 超重或者 at_{r_i} 超过时间窗则不把 r_i 加入到解路径向量中, 而是将仓库编号 0 加入到解路径向量中 (表示车辆回到仓库), 这样在解路径向量中就构成了子路径 1;

4) 按照步骤 2、3 的方法遍历染色体个体向量中剩下的客户编号, 可以求出所有子路径。

2.2 求解 VRPTW 的混合蚁群算法

算法的流程如图 1 所示。

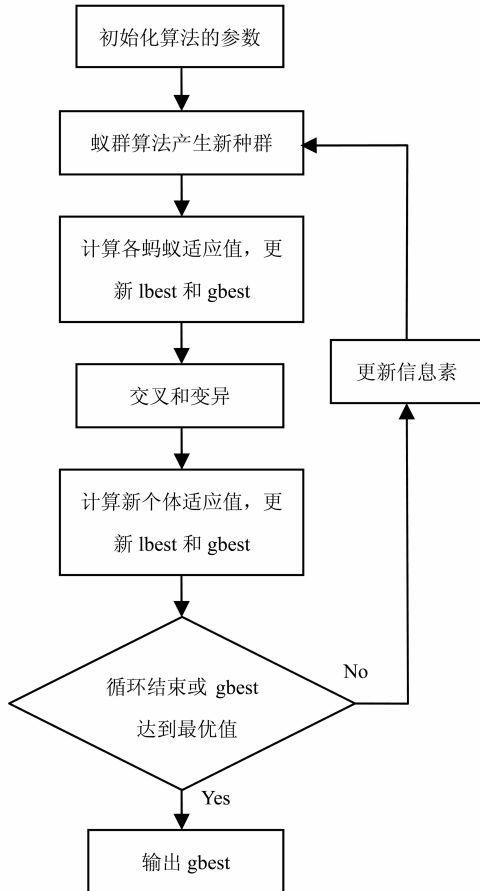


图 1 混合蚁群算法流程图

Fig. 1 Flow diagram of hybrid ant colony algorithm

2.2.1 蚁群算法产生新种群 首先采用的蚁群算法求解 VRPTW 的新种群, 具体步骤如下。

1) 设置蚂蚁数量 (一般与客户数相同), 每只蚂蚁的节点选择概率如式 (15) 所示

$$j = \begin{cases} \arg \max_{k \in \text{allowed}_i} (\tau_{ik})^\alpha (\eta_{ik})^\beta (T_{ik})^\gamma & q \leq q_0 \\ p_{ij}^k(t) = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta (T_{ij})^\gamma}{\sum_{s \in \text{allowed}_i} (\tau_{is})^\alpha (\eta_{is})^\beta (T_{is})^\gamma} & q > q_0 \end{cases} \quad (15)$$

其中 τ_{ik} 是指边 (i, k) 上的信息素, η_{ik} 是指与边 (i, k) 相关的启发信息函数, 一般取 $1/d(i, k)$, $d(i, k)$ 为节点 i 与 k 之间的距离; α 和 β 分别为信息素因子和启发因子, 分别影响信息素和启发信息函数的重要程度。

T_{ik} 是时间窗信息函数, γ 是时间窗信息因子, T_{ik} 计算方法如式 (16)

$$T_{ik} = \frac{1}{\text{wait}_j(l_j - e_j)} \quad (16)$$

wait_j 是车辆在节点 j 的等待时间, 当车辆到达节点 j 的时间 at_j 早于节点的左时间窗 e_j 时, 等待时间取值为 $e_j - at_j$, 若车辆到达时间晚于左时间窗 e_j 且早于右时间窗 l_j 则将等待时间设置为较小的数; $l_j - e_j$ 为节点 j 的时间窗跨度, 在等待时间相同的情况下时间窗跨度越小的节点应该优先选择, 由公式 (16) 可知, 等待时间越短、时间窗跨度越小的节点选择概率越大, 使得车辆配送的效率越高。

每只蚂蚁将已经走过的节点添加到禁忌表中, 禁忌表是记录每只蚂蚁目前已走过的客户节点 (为了不重复服务同一个客户), allowed_i 表示可选择的下一个客户节点的集合 (全部节点去掉禁忌表中节点的集合); $q_0 \in (0, 1)$ 是设定的参数, q 是 $(0, 1)$ 内均匀分布的随机变量, 当 q 小于或等于 q_0 时, j 取 allowed_i 中使 $(\tau_{ik})^\alpha (\eta_{ik})^\beta (T_{ik})^\gamma$ 求得最大的 k , 这样可以有一定概率取到使得边 (i, k) 信息素最高、距离节点 i 最短而且时间窗间隔时间最短的节点作为下一节点; 当 q 大于 q_0 时根据 $p_{ij}^k(t)$ 算出的概率按照轮盘赌的方法选择节点 j 。

2) 当蚂蚁从节点 i 选中节点 j 后, 应该减少边 (i, j) 的信息素, 这样可以增加蚂蚁选择其他节点的概率, 本文按照式 (17) 更新边 (i, j) 上的信息素

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij} \quad (17)$$

$$\Delta\tau_{ij} = Q/l_k$$

其中 $\rho \in [0, 1]$ 为信息素挥发系数, Q 是信息素大小 (通常取常数), l_k 是指本次迭代中第 k 只蚂蚁已

经走过的路径长度。

3) 每只蚂蚁按照步骤 1 选择完所有节点后即产生一个染色体个体, 所有蚂蚁产生的染色体个体构成初始群体。

2.2.2 计算适应值 产生了新种群后, 需要计算各染色体个体对应的适应值, 具体步骤如下。

1) 各染色体个体按照 2.1 的方法求解对应的解路径;

2) 解路径的子路径数量即配送需求的车辆数 K , 将各子路径的配送成本全部加起来求出总配送成本后, 根据公式 (2) 计算出染色体个体的适应值;

3) 比较各染色体对应的适应值, 更新当代局部最小值 $lbest$ 和全局最小值 $gbest$ 。

2.2.3 交叉和变异操作 遗传算法的交叉操作可以提高算法的全局搜索能力, 本文在蚁群算法产生的新种群中选取两个染色体个体作为父代进行交叉操作, 进行交叉的父代个体选择方法如下。

1) 选取当代局部最小值 $lbest$ 作为第一个父代个体;

2) 为了更有利于后代的进化效果, 第二个父代需要选择与 $lbest$ 有一定差异的个体^[14], 具体做法是先随机选择一个染色体个体作为父代 2, 然后计算两个父代的适应值 $F1$ 和 $F2$ 以及当前迭代适应值的最大值 $Fmax$ 和最小值 $Fmin$, 接着计算 $(F1 - F2)/(Fmax - Fmin)$ 的绝对值, 如果结果大于给定的阈值则开始交叉操作, 否则重新选择第二个父代。

按照上述方法选择了两个父代个体后, 开始交叉操作, 本文算法采用了部分匹配交叉算子, 交叉操作具体步骤如下。

1) 随机产生两个交叉位作为交叉区域;

2) 将两个父代的交叉区域分别加入对方的末尾;

3) 分别删除两个父代中与交叉区域相同的客户点即产生两个新的个体;

4) 计算新个体的适应值, 如果新个体更优则更新局部最小值 $lbest$ 和全局最小值 $gbest$ 。

例如, 父代个体 1 为 $X = [3\ 1\ 8\ 4\ 2\ 6\ 7\ 5]$, 父代个体 2 为 $Y = [2\ 8\ 5\ 7\ 3\ 4\ 6\ 1]$, 随机产生两个交叉位分别为 3 和 6, 取出 X 的第 3 至第 6 位 $[8\ 4\ 2\ 6]$ 作为交叉区域, Y 的第 3 至第 6 位 $[5\ 7\ 3\ 4]$ 作为交叉区域, 将交叉区域加入到对方末尾后两个个体分别成为 $[3\ 1\ 8\ 4\ 2\ 6\ 7\ 5\ 5\ 7\ 3\ 4]$ 和 $[2\ 8\ 5\ 7\ 3\ 4\ 6\ 1\ 8\ 4\ 2\ 6]$, 分别删除相同客户点后产生两个新个体为 $[1$

$8\ 2\ 6\ 5\ 7\ 3\ 4]$ 和 $[5\ 7\ 3\ 1\ 8\ 4\ 2\ 6]$ 。

变异操作相对交叉操作更简单, 一般只需要将染色体个体中的某一位与另一位交换即可, 通过变异可以提高算法的局部搜索能力, 本文算法的变异操作的步骤如下。

1) 选取当代局部最小值 $lbest$ 作为变异的个体;

2) 在 $lbest$ 的两条不同子路径上分别产生 1 个变异位, 计算两个变异位交换后的个体适应值, 如果适应值更优则保留变异后的个体。

3) 如果变异后的个体更优则更新局部最小值 $lbest$ 和全局最小值 $gbest$ 。

2.2.4 全局更新信息素 当一次迭代结束后, 需要更新每条边的信息素, 信息素按照式 (18) 的方法更新

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (18)$$

其中 $\Delta\tau_{ij}^k(t)$ 为在本次迭代中第 k 只蚂蚁对边 (i, j) 的信息素增量, $\Delta\tau_{ij}(t)$ 则为本次迭代中的所有蚂蚁对边 (i, j) 的信息素增量之和。

$\Delta\tau_{ij}^k(t)$ 有三种计算方法, 分别是蚁周系统 (ant-cycle system)、蚁量系统 (ant-quantity system) 和蚁密系统 (ant-density system)。本文使用的是蚁周系统, 按照式 (19) 的方法计算

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & \text{若第 } k \text{ 只蚂蚁选择了 } (i, j) \\ 0 & \text{否则} \end{cases} \quad (19)$$

其中信息素大小 Q 为常数, L_k 为第 k 只蚂蚁在本次迭代的路径总长度。

3 仿真实验

3.1 实验的参数设置

本文使用 Matlab 对算法进行仿真实验, 其中蚁群算法的各参数设置为: $\alpha = 1, \beta = 5, \gamma = 2, \rho = 0.1, q_0 = 0.7, Q = 100$; 遗传算法的交叉概率设置为 0.9, 阈值设置为 0.2, 变异概率设置为 0.2, 求解目标函数时 $p1$ 设置为 10 000, $p2$ 设置为 1。

3.2 实验及结果分析

3.2.1 实验 1 实验 1 采用 VRPTW BENCHMARK PROBLEMS 测试集的数据, 该测试集的数据分为 C 类 (客户的地理位置集中分布)、R 类 (客户的地理位置随机分布) 和 RC 类 (客户的地理位置集中和随机相混合) 3 种数据集, 本实验选取 3 类数据

中有 25 个客户的 C101 - 25、R101 - 25、RC101 - 25 和有 100 个客户的 C101 - 100、R101 - 100、RC101 - 100 共 6 组数据，6 组数据的理想解如表 1 所示。

表 1 测试集的理想解

Table 1 Optimal Solutions of the test problems

数据集	车辆数	路径长度/km
C101 - 25	3	191.8
C101 - 100	10	827.3
R101 - 25	8	617.1
R101 - 100	20	1637.7
RC101 - 25	4	461.1
RC101 - 100	15	1 619.8

本文分别使用蚁群算法和本文算法进行 10 次仿真实验，25 个客户的数据迭代次数为 400 次，100 个客户的数据迭代次数为 800 次，蚁群算法和本文算法求出的最好解的结果如表 2 所示。

表 2 实验 1 的仿真结果

Table 2 Simulation results of the first experiment

数据集	蚁群算法		本文算法	
	车辆数	路径长度/km	车辆数	路径长度/km
C101 - 25	4	211.8	3	191.8
C101 - 100	10	869.8	10	828.1
R101 - 25	8	635.4	8	618.3
R101 - 100	23	1746.6	20	1644.8
RC101 - 25	6	537.1	4	471.6
RC101 - 100	19	1 820.1	15	1698.1

由表 2 可知，本文算法在 6 组数据中无论在车辆数还是在配送的路径长度都明显优于蚁群算法的结果。在 6 组数据中，本文算法的都得到了理想解的车辆数，误差率均为 0，而路径长度的误差率分别是 0、0.1%、0.19%、0.43%、2.23% 和 4.61%，结果都非常接近理想解，其中 C101 - 25 得到了理想解，C101 - 25 的配送路线如图 2 所示，图 2 中的 X 坐标和 Y 坐标是仓库和客户位置的坐标，路线图由 3 辆车配送，路径长度为 191.813 6，由 3 条子路径构成，分别是 0 - 5 - 3 - 7 - 8 - 10 - 11 - 9 - 6 - 4 - 2 - 1 - 0、0 - 20 - 24 - 25 - 23 - 22 - 21 - 0、0 - 13 - 17 - 18 - 19 - 15 - 16 - 14 - 12 - 0，其中 0 表示仓库号，1 至 25 表示客户号。

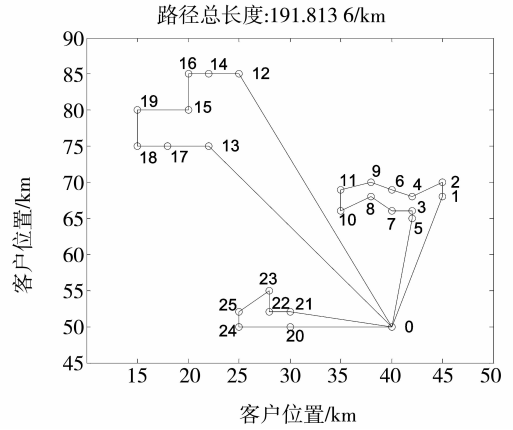


图 2 C101 - 25 的车辆配送路线图

Fig. 2 Vehicle distribution routing for C101 - 25

3.2.2 实验 2 为了更好的说明本文算法求解 VRPTW 的有效性，实验 2 采用文献 [5] 中的数据，实验数据包括 1 个仓库和 20 个客户，分布在边长为 200 km 的正方形区域，仓库的坐标为 145 km，130 km，每辆车的载重量为 8 t，车辆的行驶速度为 60 km/h，忽略车辆在客户点的服务时间，20 个客户的具体数据如表 3 所示。

表 3 客户的具体数据

Table 3 Data of customers

客户编号	横坐标/km	纵坐标/km	需求量/t	左时间窗/h	右时间窗/h
1	128.0	85.0	0.1	0.5	5.5
2	184.0	34.0	0.4	1.0	8.0
3	154.0	166.0	1.2	0.0	5.5
4	189.0	152.0	1.5	0.5	4.5
5	155.0	116.0	0.8	0.0	5.0
6	39.0	106.0	1.3	1.0	7.5
7	106.0	76.0	1.7	0.5	7.0
8	86.0	84.0	0.6	0.5	3.0
9	125.0	21.0	1.2	1.0	5.0
10	138.0	52.0	0.4	0.5	4.0
11	67.0	169.0	0.9	1.0	5.0
12	148.0	26.0	1.3	1.0	8.0
13	18.0	87.0	1.3	1.0	5.0
14	171.0	110.0	1.9	0.0	10.5
15	7.4	10.0	1.7	1.5	3.5
16	2.0	28.0	1.1	2.0	10.0
17	119.0	198.0	1.5	0.5	6.5
18	132.0	151.0	1.6	0.0	5.5
19	64.0	5.6	1.7	1.0	7.5
20	96.0	148.0	1.5	0.5	8.5

文献 [5] 得到的最好解是 3 辆车进行配送，车辆利用率分别是 96.25%、100% 和 100%，配送路径总长度为 1 266.01 km，本实验对上述数据进行

仿真实验 10 次, 实验的迭代次数 400 次, 实验结果如表 4 所示。

表 4 实验 2 的仿真结果

Table 4 Simulation results of the second experiment

	配送车辆数	路径总长度/km
最好解	3	1 249.27
最差解	3	1 264.65
平均解	3	1 258.14

由表 4 可知, 本文算法得到的最差解都比文献 [5] 的最好解的路径总长度更短, 最好解的路径总长度为 1 249.27 km, 对应的 3 条子路径分别是: 0-7-8-15-16-13-20-0、0-5-14-4-3-17-11-0、0-18-1-10-2-12-9-19-6-0, 车辆利用率分别是 98.75%、97.5% 和 100%, 实验表明本文算法的结果在同样使用 3 辆车配送的基础上, 车辆的使用率更加平均, 路径的总长度则更短。

4 结 论

本文将蚁群算法和遗传算法相结合来求解 VRPTW, 介绍了求解 VRPTW 的编码和解码方法、产生新种群的方法、交叉和变异以及全局更新信息素的方法。进行了两个仿真实验, 实验 1 验证了本文算法在求解 VRPTW 时比基本蚁群算法更加有效, 实验 2 表明本文算法比文献 [5] 的结果更优, 说明本文算法是求解 VRPTW 的有效算法。

参考文献:

- [1] 蒋波. 基于遗传算法的带时间窗车辆路径优化问题研究[D]. 北京: 北京交通大学, 2010.
- [2] 罗中良, 易明珠. 最优化问题的蚁群混合差分进化算

法研究[J]. 中山大学学报: 自然科学版, 2008, 47(3): 37-41

- [3] 罗中良, 易明珠, 唐华, 等. 开 Y-开 Δ 变压器相序调整的蚁群优化方法及收敛性研究[J]. 中山大学学报: 自然科学版, 2007, 46(2): 29-32
- [4] 李全亮. 蚂蚁算法在带时间窗车辆路径问题中的应用研究[J]. 数学的实践与认识, 2006, 36(10): 173-178.
- [5] 李琳, 刘士新, 唐加福. 改进的蚁群算法求解带时间窗的车辆路径问题[J]. 控制与决策, 2010, 25(9): 1378-1382.
- [6] 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.
- [7] 吴天羿, 许继恒, 刘建永, 等. 求解有硬时间窗车辆路径问题的改进遗传算法[J]. 系统工程与电子技术, 2014, 36(4): 708-712.
- [8] 袁建清. 求解动态车辆调度问题的混合禁忌搜索算法[J]. 计算机应用与软件, 2012, 29(4): 148-150.
- [9] 李进, 傅培华. 基于能耗的带时间窗车辆路径问题建模与仿真[J]. 系统仿真学报, 2013, 25(6): 1147-1154.
- [10] 王飞. 带时间窗车辆调度问题的改进粒子群算法[J]. 计算机工程与应用, 2014, 50(6): 226-229.
- [11] 戚铭尧, 张金金, 任丽. 基于时空聚类的带时间窗车辆路径规划算法[J]. 计算机科学, 2014, 41(6): 218-221.
- [12] 潘立军. 带时间窗车辆路径问题及其算法研究[D]. 长沙: 中南大学, 2012.
- [13] 黄震. 混合量子粒子群算法求解车辆路径问题[J]. 计算机工程与应用, 2013, 49(24): 219-223.
- [14] 周艳聪, 孙晓晨, 余伟翔. 基于改进遗传算法的物流配送路径优化研究[J]. 计算机工程与科学, 2012, 34(10): 118-122.