文章编号:1671-6833(2020)04-0052-07

求解双目标 VRPTW 的改进混合蚁群算法

朱晓东,王 鼎

(郑州大学 电气工程学院,河南 郑州 450001)

摘 要:为了解决基本混合蚁群算法在求解大规模带时间窗车辆路径问题(VRPTW)时存在的问题,提出一种改进的双目标混合蚁群算法。首先在节点选择上使用周边选择策略提升选择效率,并提出一种首节点选择策略来加速算法收敛;其次在信息素叠加公式上增加了与车辆数有关的惩罚函数,使算法能够同时优化距离与车辆数两个目标;最后提出一种新的局部优化算法,通过将节点数较少的线路中的节点插入到其他线路来提升车辆利用率。通过该算法在 Solomon 标准数据集上的实验和对比,说明了改进的算法具有搜索能力强、收敛速度快、鲁棒性强等优点。

关键词:蚁群算法;车辆路径问题;时间窗;双目标

中图分类号:TP301.6 **文献标志码**:A **doi**:10.13705/j.issn.1671-6833.2020.04.004

0 引言

车辆路径问题[1](vehicle routing problem,VRP)是指使用一组车辆从配送中心出发给已知客户进行配送服务,在满足一定约束条件下,求解最短路径的线路。带时间窗的车辆路径问题(vehicle routing problem with time window,VRPTW)是 VRP 的拓展,它在 VRP 的基础上增加了服务时间的限制,即每个客户都有接受服务的时间范围,称为时间窗,车辆必须在时间窗内对客户进行服务。良好的路径规划不仅可以节省运输成本,提高配送效率,还可以提升服务质量,带来巨大的经济效益。

由于 VRPTW 是一个非确定性多项式难题(NP-Hard),精确算法无法在有效时间内完成求解,使用启发式算法求解大规模客户 VRPTW 问题受到专家学者的广泛关注,比如遗传算法[2-3]、差分进化算法[4]、粒子群优化算法[5]、布谷鸟搜索算法[6]等。

蚁群算法[7]在解决 VRPTW 问题上取得了众多成果,如 Yu 等[8]提出了蚁群算法和禁忌搜索混合的算法,该算法先使用蚁群算法得到一个局部最优解,然后使用一次禁忌搜索算法来跳出局部最优;Ding 等[9]对蚁群系统作了进一步改进,在计算转移概率时考虑了节点之间的节约值,同时采用 λ-interchange 作为局部优化方法,提升了算法的收敛速度;朱杰等[10]将模拟退火算法和蚁群算法结合,改进了蚁群算法中状态转移公式和信息素更新策略,提高算法搜索能力;刘云等[11]将单亲遗传算法和蚁群算法相结合,提升了传统蚁群算法的计算效率和收敛速度,并通过实际算例说明算法的实用性;柴获等[12]采用一种混合蚁群算法来解决双目标带时间窗车辆路径问题,提出了一种新的搜索空间构建方法,并改进了蚁群算法的状态转移公式;孙小军等[13]提出一种动态混合蚁群优化算法,该算法将蚁群算法和遗传算法进行融合,自适应控制蚁群算法参数;葛斌等[14]提出了一种动态混合改进蚁群算法,该算法在对客户分区的基础上,引入了交通拥堵因子来改进传统算法中的状态转移公式,并将挥发因子设置为随机变量,使算法更稳定地收敛到全局最优解。

上述文献通过对传统蚁群算法的改进,促进了蚁群算法在 VRPTW 问题上的应用及发展,但仍存在不足。如 Yu 等[8-10]提出的算法均是以最短路径为目标,但是在实际问题中还要考虑车辆数等多个目标,单目标优化无法满足实际需求。刘云等[11-12]提出了解决实际问题的多目标改进

收稿日期:2020-01-08;**修订日期**:2020-05-11

基金项目:国家自然科学基金资助项目(61876169,61922072)

作者简介:朱晓东(1970—),男,河南郑州人,郑州大学副教授,主要从事智能控制及智能信息处理方面的研究,E-mail:zhu_xd@zzu.edu.cn。

蚁群算法,但所解决问题中节点数较少,在解决大规模节点问题上存在搜索能力不足、收敛速度慢等缺点。针对这些问题,笔者提出一种用于解决大规模双目标 VRPTW 的改进混合蚁群算法。

1　VRPTW 数学模型

记 $G=(V,E)$ 为赋权图,V 为顶点集,$V=V_0\cup V_c$,其中 $V_0=\{0\}$ 表示配送中心,$V_c=\{1,2,\cdots,n\}$ 表示配送节点;E 为边集,$E=\{(i,j)\mid i,j\in V,i\neq j\}$ 表示节点 i 和 j 之间的有向线段。设 d_{ij} 表示节点 i、j 之间的距离,t_{ij} 表示车辆从 i 到 j 的时间。节点 i 的需求量为 d_i,接受服务的时间窗为 $[ET_i,LT_i]$,服务时长为 ST_i。$K=\{1,2,\cdots,m\}$ 表示车辆编号,每辆车载重恒定为 D,车辆 k 到达节点 i 的时间 T_{ik},若车辆 k 在时间窗开始之前到达,则需等待一段时间 w_{ik},$w_{ik}=\max(0,ET_i-T_{ik})$。定义变量:

$$x_{ijk}=\begin{cases}1, & \text{车辆 } k \text{ 从 } i \text{ 行驶到 } j;\\ 0, & \text{其他。}\end{cases}$$

$$y_{ki}=\begin{cases}1, & \text{节点 } i \text{ 由车辆 } k \text{ 服务;}\\ 0, & \text{其他。}\end{cases}$$

本文中 VRPTW 的双目标数学模型可表述为:

$$C_1=\min\sum_{i\in V}\sum_{j\in V}\sum_{k\in K}d_{ij}x_{ijk},\tag{1}$$

$$C_2=\min\sum_{j\in V}\sum_{k\in K}x_{0jk},\tag{2}$$

s.t:

$$\sum_{i\in V}d_iy_{ki}\leqslant D,\forall k\in K,\tag{3}$$

$$\sum_{i\in V_c}y_{ki}=1,\forall k\in K,\tag{4}$$

$$\sum_{i\in V}x_{ijk}=y_{kj},\forall j\in V,\forall k\in K,\tag{5}$$

$$\sum_{j\in V}x_{ijk}=y_{ki},\forall i\in V,\forall k\in K,\tag{6}$$

$$\sum_{j\in V_c}x_{0jk}=\sum_{j\in V_c}x_{0jk}=1,\forall k\in K,\tag{7}$$

$$T_{ik}<LT_i,\forall i\in V,\tag{8}$$

$$T_{ik}+w_{ik}+ST_i+t_{ij}<LT_j,\forall i,j\in V,\forall k\in K。\tag{9}$$

其中,式(1)、(2)为目标函数,表示以路径的最短距离和最小车辆数为目标;约束条件(3)表示每条线路上的客户需求量不能超过车辆容量限制;约束条件(4)表示每个客户只能被访问一次;约束条件(5)表示车辆在服务节点 j 之前只服务了一个节点 i;约束条件(6)表示车辆在服务节点 i 后只服务了一个节点 j,避免线路内部产生回路;约束条件(7)表示每辆车的起点和终点都必须是配送中心;式(8)~(9)为时间窗约束。

2　基本混合蚁群算法求解 VRPTW

2.1　可行解的构建

使用基本蚁群算法求解 VRPTW 时,首先令每只蚂蚁从配货中心出发,根据状态转移概率随机地从候选节点集合中选择下一节点来构建可行线路,若线路上无可行节点添加时,该蚂蚁返回配货中心,完成一条可行线路的构建。接着重复上述过程,继续从剩余节点中构建可行线路,直到所有节点都被访问,完成一个可行解的构建。

假设当前节点为 i,下一个节点的候选集合为 W_0,蚂蚁从节点 i 转移至节点 j 的转移概率为:

$$p_{ij}=\begin{cases}\dfrac{\tau_{ij}^{\alpha}\cdot\eta_{ij}^{\beta}}{\sum\limits_{j\in W_0}\tau_{ij}^{\alpha}\cdot\eta_{ij}^{\beta}}, & j\in W_0;\\ 0, & \text{其他,}\end{cases}\tag{10}$$

式中:p_{ij} 表示蚂蚁从节点 i 转移到节点 j 的概率;τ_{ij} 为 i、j 两点的信息素浓度;$\eta_{ij}=\dfrac{1}{d_{ij}}$ 为能见度;α、β 为启发因子。

当所有蚂蚁完成可行解的构建后,按照下式进行全局信息素更新。

$$\tau_{ij}^{\text{new}}=(1-\rho)\cdot\tau_{ij}^{\text{old}}+\Delta\tau_{ij},\tag{11}$$

$$\Delta\tau_{ij}=\begin{cases}\sum\limits_{u=1}^{m}\dfrac{Q}{dis(u)}, & (i,j)\in L(u);\\ 0, & \text{其他,}\end{cases}\tag{12}$$

式中:$\rho(0<\rho<1)$ 为常数,表示信息素的挥发速率;$\Delta\tau_{ij}$ 为信息素累计量;m 为蚂蚁数量;$L(u)$ 为第 u 只蚂蚁的路径,$dis(u)$ 表示第 u 只蚂蚁的总距离;Q 为常数,表示每只蚂蚁释放信息素的总量。

2.2　局部优化算法

车辆路径问题中常用的局部优化算法有 2-interchange 和 or-interchange。2-interchange 用于两条路径间的邻域搜索,该算法从两条路径分别选取 a 和 b 个节点进行交换来产生邻域解($a,b\in\{0,1,2\}$ 且 a 和 b 不同时为 0)。图 1 表示了 $a=1$、$b=1$ 时的一个样例:将路径 l 中的 1 个节点 x 与路径 k 中的 1 个节点 y 进行交换产生邻域解。

or-interchange 用于单条线路的邻域搜索,将线路内部两个节点交换位置来产生邻域解。图 2 表示将路径 l 中 x、z 两个节点交换产生邻域解。

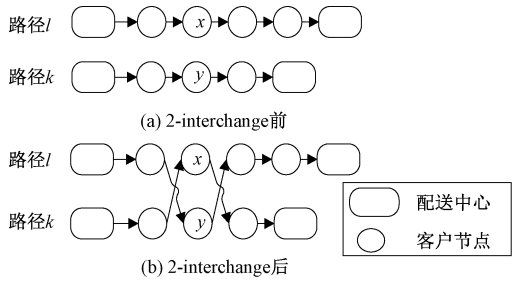

图1 $a=1$、$b=1$ 时 2-inerchange 变换

Figure 1 2-inerchange transformation when $a=1, b=1$

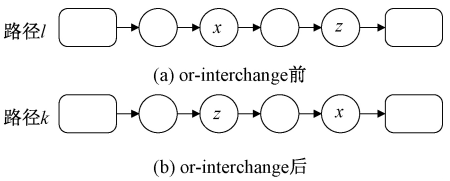

图2 or-interchange 变换过程

Figure 2 or-interchange transform process

3 改进的混合蚁群算法

在面对大规模 VRPTW 时,由于较多的节点数量和复杂的约束,上述基本混合蚁群算法收敛速度慢,搜索效率低,并且无法满足多目标的需求。笔者从候选节点的选择、转移概率和信息素叠加机制等方面对基本蚁群算法进行改进,并提出了一种新的局部优化方法来提升算法的搜索能力。

3.1 候选节点集合的改进

(1)周边搜索策略。在解决大规模节点问题时,基本蚁群算法产生的候选节点集合中节点数量较多,算法难以搜索到最优解。笔者结合 VRPTW 的问题特点,采用周边搜索策略来缩小候选节点搜索范围。该策略是从下一个可行节点的集合中,选择距离当前节点最近的若干节点作为候选节点集合 W,如图3所示。

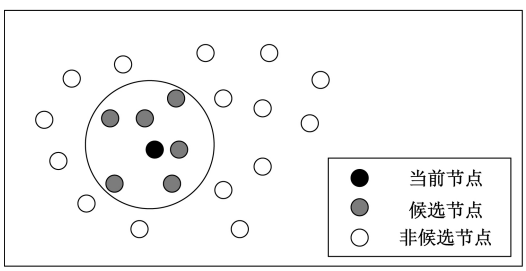

图3 周边选择策略

Figure 3 Peripheral selection strategy

(2)首节点的选择。在可行线路的构建过程中,线路上首节点的选择将会影响接下来整条路径的节点选择,对全局规划来说非常重要。面对大规模客户的 VRPTW 问题,线路上的首个节点相对于其他节点来说约束较少,可行节点的数量远多于路径中的节点,为了能够从众多可行节点中选择最合适的节点,笔者提出一种首节点选择策略来提升算法的收敛性。该策略可表述为:迭代初期,每条可行线路上的首个节点按概率从候选节点中随机选取,当迭代进度达到一定条件时,首节点为候选节点中转移概率最大的节点。线路的首节点用 i_1 表示,其选取规则如下:

$$i_1=\begin{cases}\arg\max p_{0i_1}, & \dfrac{it}{it_{\mathrm{m}}}>\varepsilon;\\ \text{random } i_1\in W, & \text{其他},\end{cases}\tag{13}$$

式中:p_{0i_1} 表示配货中心到 i_1 的转移概率;it 表示当前迭代次数;it_{m} 表示最大迭代次数;ε 表示迭代的进行程度的参数;W 为候选节点集合。

3.2 转移概率公式改进

在解决大规模复杂约束的车辆路径问题时,基本蚁群算法的转移概率公式中包含节点的特征信息较少,导致算法搜索能力不足,收敛速度慢。为了提升算法的收敛性,将节约算法[15]加入转移概率公式。节约算法首先将两个节点 i、j 分别分配到一条空余的线路上,再计算将两个节点合并到一条线路上后可节省的节约值 S_{ij}。式(14)为节约值计算公式。

$$S_{ij}=d_{0i}+d_{0j}-d_{ij},\tag{14}$$

式中:d_{0i}、d_{0j} 和 d_{ij} 分别表示配货中心到 i、j 节点的距离和 i、j 两节点的距离。改进后的状态转移公式如下:

$$p_{ij}=\begin{cases}\dfrac{S_{ij}\cdot\tau_{ij}^{\alpha}\cdot\eta_{ij}^{\beta}}{\sum\limits_{j\in W}S_{ij}\cdot\tau_{ij}^{\alpha}\cdot\eta_{ij}^{\beta}}, & j\in W;\\ 0, & \text{其他}。\end{cases}\tag{15}$$

3.3 信息素叠加公式的改进

基本蚁群算法在叠加信息素时仅考虑路径的总距离,未对车辆数进行优化。为了在优化距离的同时降低车辆数,本文算法对基本蚁群算法的信息素叠加公式进行了改进,如式(16)所示,使其能够完成双目标的优化。

$$\Delta\tau_{ij}=\begin{cases}\sum\limits_{u=1}^{m}\dfrac{Q}{dis(u)}\cdot f(k), & (i,j)\in L(u);\\ 0, & \text{其他}。\end{cases}\tag{16}$$

和式(12)相比,式(16)中增加了和车辆数相关的函数 $f(k)$。$f(k)$ 如式(17)所示。

$$f(k)=\frac{1+\max(0,V_{\mathrm{globalbest}}-V(k))}{1+\max(0,V(k)-\max(V_{\mathrm{localbest}},V_{\mathrm{globalbest}}))},\tag{17}$$

式中:$V(k)$ 为第 k 只蚂蚁求得可行解的车辆数;$V_{\text{globalbest}}$ 为当前迭代种群中的最小车辆数;$V_{\text{localbest}}$ 为全局最小车辆数。$f(k)$ 可以根据当前解的车辆数对它的信息素浓度进行奖励或惩罚:若当前迭代种群中的最小车辆数大于全局最小车辆数,则对 $V(k) > V_{\text{localbest}}$ 的解进行惩罚;若当前迭代种群中最小车辆数小于全局最小车辆数,那么对 $V(k) > V_{\text{globalbest}}$ 的解进行惩罚,对 $V(k) < V_{\text{globalbest}}$ 的解进行奖励。

由于蚁群算法中信息素叠加是一个正反馈过程,为了避免搜索停滞,笔者采用最大最小蚁群算法将信息素浓度限定于 $[\tau_{\max},\tau_{\min}]$,如式(18) 所示

$$\tau_{ij}=\begin{cases}\tau_{\max}, & \tau_{ij}>\tau_{\max};\\ \tau_{\text{initial}}, & \tau_{ij}<\tau_{\min},\end{cases} \tag{18}$$

式中:τ_{initial} 表示信息素的初始值。

3.4　改进的局部优化算法

笔者通过实验研究发现,在处理大规模节点和复杂约束的 VRPTW 问题时,基本蚁群算法构建的可行解中会产生客户数量比较少的线路,这将极大降低车辆的利用率,传统的局部优化算法对此类问题考虑不够。因此笔者提出了一种新的局部优化方法,将插入算法和 2-interchange 结合,在优化距离同时降低车辆数,提高车辆的利用效率。算法步骤如下。

Step 1　将当前解分为节点数量不小于 3 的线路集合 R_M 和节点数量小于3的线路集合 R_N,如图 4 所示。

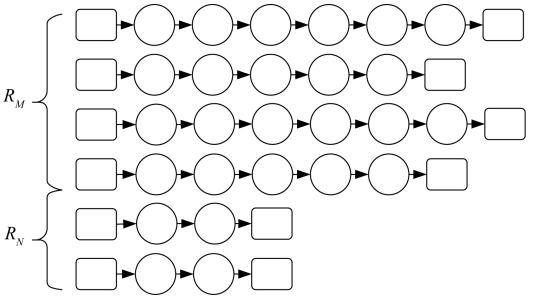

图 4　路线分类示意图

Figure 4　Schematic diagram of route classification

Step 2　使用 2-interchange 和 or-interchange 对 R_M 进行局部优化 z 次。迭代初期 z 取较大值可以加速解的收敛,随着迭代进行,算法产生更优邻域解的难度逐渐加大,通过 z 取值逐渐减小来提升算法效率。在本文算法中,z 初始值为 6,每经过 10 次迭代减小 1。

Step 3　设 $l(i)$ 为 R_N 中第 l 条线路上的第 i 个节点,初始化 $l=1$,$i=1$。

Step 4　将 $l(i)$ 插入到 R_M 中的所有节点之间。若 $l(i)$ 插入 R_M 中第 k 条线路的第 j 个节点后产生了可行解,根据式(19) 计算插入后的 $\Delta Cost_{likj}$。式(19) 中,$cost$ 为线路的总距离;l、k 为插入前的线路;l_{new}、k_{new} 为插入后的线路。直到遍历 R_M 中所有线路上的所有节点,计算所有可行解的 $\Delta Cost_{likj}$,跳转 Step 6。若 $l(i)$ 插入 R_M 的所有节点之间均无法产生可行解,则转到 Step 5。

$$\Delta Cost_{likj}=cost(l)+cost(k)-cost(l_{\text{new}})-cost(k_{\text{new}})。 \tag{19}$$

Step 5　将 $l(i)$ 与 R_M 中随机一个节点进行交换,要求交换后不改变解的可行性。然后重复 Step 4。值得注意的是,每个节点优化时本步骤仅执行一次,若仍无可行解产生,则跳转至 Step 7。

Step 6　将 $l(i)$ 插入到 $\Delta Cost_{likj}$ 最大的位置。更新插入后的线路。

Step 7　令 $i=i+1$,返回至 Step 4,继续对下一个节点进行优化,直到线路 l 上所有节点完成优化。

Step 8　令 $l=l+1$,转至 Step 2,继续对下一条线路进行优化,直到 R_N 中所有线路完成优化。

Step 9　将 R_N 中剩余节点使用 2-interchange 方法优化。

3.5　改进算法整体过程

改进算法整体流程图如图 5 所示。

4　实验与分析

笔者将改进的蚁群算法在 Solomon[16] 标准数据集上进行实验。Solomon 标准数据集有 R1、R2、C1、C2、RC1、RC2 共 6 类问题。这 6 类问题按节点分布方式可分为随机分布(R)、聚集分布(C)和混合分布(RC)。其中 R1、C1、RC1 问题中节点的时间窗较窄,车辆载重较小;R2、C2、RC2 问题中节点的时间窗较宽,车辆载重较大。Solomon 数据集场景多样具有代表性,因此其常被文献用于算法评估。

4.1　参数设置

笔者提出的改进混合蚁群算法中主要参数有式(15)中启发因子 α、β 和式(16)中每只蚂蚁信息素总量 Q。为了分析这些参数对算法的影响,笔者在 C101 测试问题上进行实验。每组实验均进行 5 次,取结果的平均值。实验最大迭代次数 $it_{\text{m}}=60$,信息素挥发速率 $\rho=0.08$。α、β 与结果的关系如图 6 所示。每只蚂蚁释放信息素总量 Q 与结果的关系如图 7 所示。

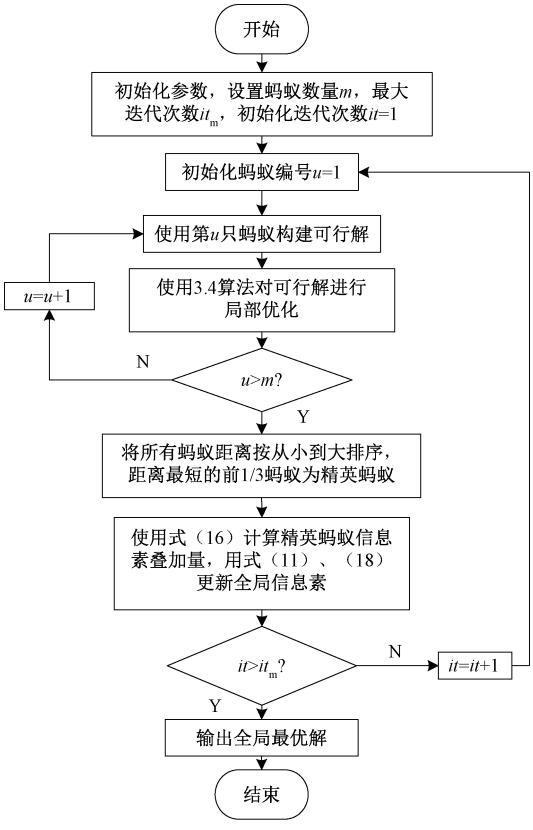

图 5 改进算法整体流程图

Figure 5 The overall flow chart of the improved algorithm

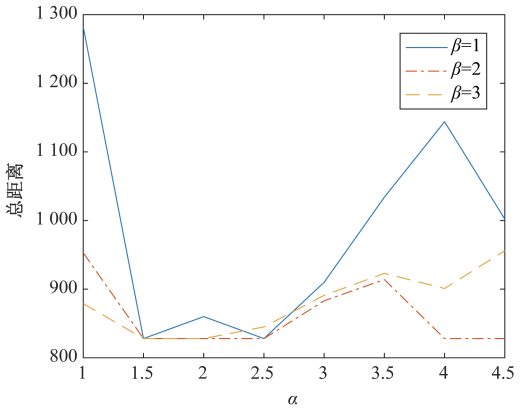

图 6 α、β 取值与总距离的关系

Figure 6 The relationship between the values of α, β and the total distance

从图 6 可以看出，当 $1.5<\alpha<2.5$、$\beta=2$ 时，算法取得较好结果。从图 7 可以看出，$60<Q<100$ 时，算法可以取得较好结果。

4.2 首节点选择策略的有效性

为了验证首节点选择策略的有效性，在标准数据集中 C101 测试问题上进行了相关实验。

首先对式(13)中参数 ε 的取值进行相关实验。若 ε 过小，算法会过早收敛，陷入局部最优；若 ε 较大，算法收敛速度慢，不能收敛于最优解。本实验中对每个 ε 取值都进行 5 次实验，结果取平均值。实验结果如图 8 所示。

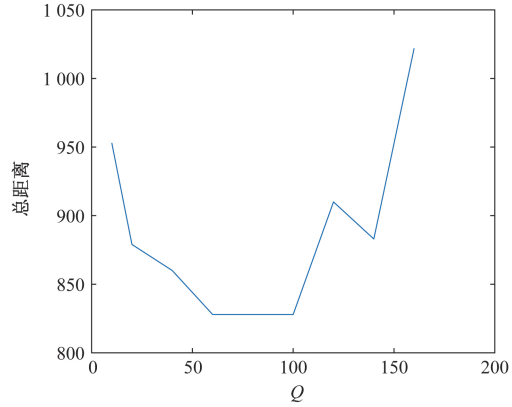

图 7 Q 取值与总距离的关系

Figure 7 The relationship between Q and total distance

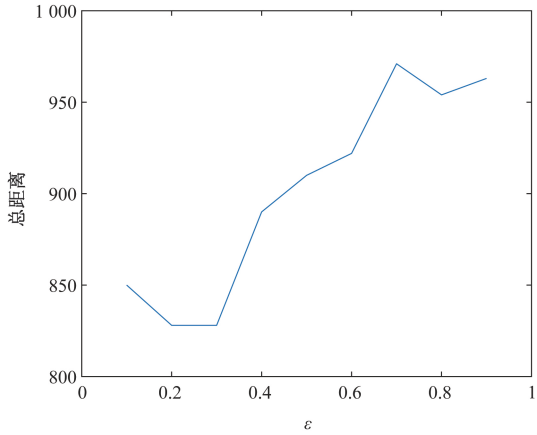

图 8 ε 取值与总距离关系

Figure 8 The relationship between ε and total distance

从图 8 中可以看出，当 $0.2<\varepsilon<0.3$ 时算法取得较好结果。

在相同条件下将无首节点选择策略的算法和加入首节点选择策略的算法进行对照试验。两者都运行 5 次，结果取平均值，实验结果如图 9 所示。

从图 9 中可以看出，虽然两者算法都收敛于最优解，但是加入首节点选择策略后算法的收敛速度明显变快，这也证明笔者提出的首节点选择策略可以提高算法的收敛速度。

4.3 标准数据集测试

为了说明本文算法具有适应性和有效性，笔者从标准数据集的每类问题中选取一个进行实验，将实验结果与基本蚁群算法和 Teymourian 等[6]提出的混合蚁群算法、Ghoseiri 等[17]提出的混合遗传算法进行比较。实验用 MATLAB 2016a 进行编程，在 Intel Core i5-5200 CPU，4 GB 内存，

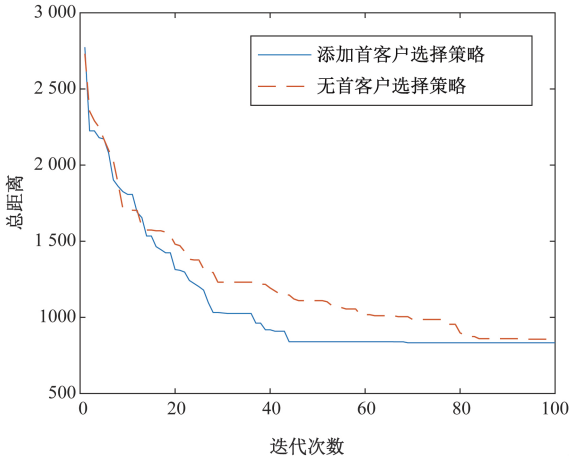

图 9　添加首节点选择策略和无首节点选择策略的比较

Figure 9　Comparisons of adding first node selection policy and no-first node selection policy

windows8.1 操作系统的计算机上运行。实验参数蚂蚁数量 $m=15$,最大迭代次数 $it_m=60$,$\rho=0.08$,根据参数设置的实验结果取 $\alpha=2$、$\beta=2$、$Q=80$。本文算法在所选问题上运行 3 次取最优结果,实验结果如表 1 所示。其中总距离为 *TD*,车辆数为 *VN*。为了便于比较算法平均值之间的差异,设 *res* 表示本文结果的平均值,*res'* 表示其他算法结果的平均值,Δ 表示结果之间的差距百分比,按式(20)计算。

$$\Delta=\frac{res-res'}{res}\times 100\%。\tag{20}$$

从表 1 中可以看出,在得出的 12 个结果中,笔者提出的改进算法在总距离和车辆数上都明显优于基本混合蚁群算法;和文献中算法相比,本文算法 75% 的结果取得了最优解,文献[6]算法

表 1　Solomon 数据集测试结果

Table 1　Solomon data set test results

测试问题	基本混合蚁群算法		文献[6]算法		文献[17]算法		本文改进算法	
	TD	*VN*	*TD*	*VN*	*TD*	*VN*	*TD*	*VN*
C102	930.9	11	828.9	10	828.9	**10**	**821.9**	**10**
R102	1 786.1	19	**1 491.0**	18	1 511.8	18	1 501.1	**17**
RC105	1 854.7	19	1 617.9	15	1 611.5	15	**1 606.3**	**14**
C201	841.5	5	**591.5**	**3**	**591.5**	**3**	**591.5**	**3**
R201	1 752.3	6	**1 214.2**	7	1 351.4	**4**	1 289.4	**4**
RC203	1 406.9	8	1 046.3	6	1 060.0	**4**	**1 045.6**	5
平均值	1 428.7	11.3	**1 131.6**	9.8	1 159.2	9.0	1 142.6	8.8
Δ	−25.0%	−28.3%	1.0%	−11.3%	−1.4%	−1.9%	—	—

41.6%的结果取得最优解,文献[17]算法 41.6%的结果取得最优解。从所有结果的平均值来看,与基本混合蚁群算法相比,本文算法在总距离上领先了 25.0%,在车辆数上领先了 28.3%;与文献[6]算法相比,本文算法在总距离上落后了 1.0%,但是在车辆数上领先了 11.3%;与文献[17]算法相比,本文算法在总距离上领先了 1.4%,在车辆数上领先了 1.9%。此外本文算法在不同类型测试问题上均采用相同参数进行实验,这也说明了本文算法具有较强的适应能力。

5　结论

笔者通过分析传统混合蚁群算法在求解大规模带时间窗路径规划问题(VRPTW)中存在的不足,提出了一种改进的双目标混合蚁群算法。该算法有针对性地采用了首节点选择策略和改进状态转移公式来加速收敛过程;考虑到路径距离与车辆数量两个优化目标,在信息素叠加过程增加了车辆数惩罚,同时改进了全局信息素更新策略。本文改进算法有效地提高了车辆利用率,扩大了搜索空间,能够在双目标下给出较优解。通过在 Solomon 标准数据集上的实验,验证了笔者所提算法的有效性。

参考文献:

[1] DANTZIG G B, RAMSER J H. The truck dispatching problem[J]. Management science, 1959, 6(1): 80-91.

[2] MOHAMMED M A, GANI M K A, HAMED R I, et al. Solving vehicle routing problem by using improved genetic algorithm for optimal solution[J]. Journal of computational science, 2017, 21:255-262.

[3] DE OLIVEIRA DA COSTA P R, MAUCERI S, CARROLL P, et al. A genetic algorithm for a green vehicle routing problem[J]. Electronic notes in discrete mathematics, 2018, 64:65-74.

[4] 汪慎文,杨锋,徐亮,等.离散差分进化算法求解共享单车调度问题[J].郑州大学学报(工学版),2019,40(4):48-53.

[5] 崔岩,张子祥,时新,等.考虑顾客时间紧迫度的生

鲜电商配送路径优化问题[J].郑州大学学报(工学版),2017,38(6):59-63.
[6] TEYMOURIAN E,KAYVANFAR V,KOMAKI G M,et al.Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem[J]. Information sciences, 2016, 334: 354-378.
[7] DORIGO M,MANIEZZO V,COLORNI A.Ant system: optimization by a colony of cooperating agents[J]. IEEE transactions on systems, man and cybernetics, part B (cybernetics), 1996,26(1):29-41.
[8] YU B,YANG Z Z,YAO B Z.A hybrid algorithm for vehicle routing problem with time windows[J]. Expert systems with applications, 2011,38(1):435-441.
[9] DING Q L,HU X P,SUN L J,et al.An improved ant colony optimization and its application to vehicle routing problem with time windows[J]. Neurocomputing, 2012,98:101-107.
[10] 朱杰,张培斯,张询影,等.基于改进蚁群算法的多时间窗车辆路径问题[J].计算机技术与发展,2019,29(1):102-105.
[11] 刘云,张惠珍.多目标带时间窗的车辆路径问题的单亲遗传混合蚁群算法[J].公路交通科技,2016,33(6):95-100,106.
[12] 柴获,何瑞春,苏江省,等.求解双目标带时间窗车辆路径问题的蚁群算法[J].交通运输系统工程与信息,2018,18(4):156-162.
[13] 孙小军,介科伟.求解带时间窗动态车辆路径问题的改进蚁群算法[J].大连理工大学学报,2018,58(5):539-546.
[14] 葛斌,韩江洪,魏臻,等.求解带时间窗车辆路径问题的动态混合蚁群优化算法[J].模式识别与人工智能,2015,28(7):641-650.
[15] CLARKE G,WRIGHT J W.Scheduling of vehicles from a central depot to a number of delivery points[J]. Operations research, 1964,12(4):568-581.
[16] SOLOMON M M.Algorithms for the vehicle routing and scheduling problems with time window constraints[J]. Operations research, 1987,35(2):254-265.
[17] GHOSEIRI K, GHANNADPOUR S F. Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm[J]. Applied soft computing, 2010,10(4):1096-1107.

An Improved Hybrid Ant Colony Algorithm for Bi-objective VRPTW

ZHU Xiaodong, WANG Ding

(School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: In order to solve the problems of basic hybrid ant colony algorithm in solving large-scale VRPTW, an improved bi-objective hybrid ant colony algorithm was proposed. Firstly, the peripheral selection strategy was used to improve the selection efficiency, and a first node selection strategy was proposed to accelerate the convergence of the algorithm. Secondly, a penalty function related to the number of vehicles was added to the pheromone superposition formula to optimize the number of vehicles while optimizing the distance. Finally, a new local optimization algorithm was proposed to improve vehicle utilization and expand the neighborhood solution by inserting nodes in routes with fewer nodes into other routes. Experiments and comparisons on Solomon benchmark problems showed that the improved algorithm had the advantages of strong search ability, fast convergence speed and strong robustness.

Key words: ant colony algorithm; vehicle routing problem; time window; bi-objective