

基于改进蚁群算法的装配线 VRPTD 问题研究

刘 凯, 牛江川, 申永军, 韩彦军

(石家庄铁道大学 机械工程学院, 河北 石家庄 050043)

摘要:采用改进蚁群算法求解了装配线物料配送的 VRPTD 问题(带最后期限时间窗的车辆路径问题)。通过信息素动态更新设计,使改进蚁群算法具有自适应性,克服了传统蚁群算法在遍历寻优过程中容易出现停滞和陷入局部最优解的缺点。通过进一步对启发函数可见度进行改进设计,提高了算法的全局搜索能力。仿真结果表明,改进蚁群算法可以很好地求解装配线 VRPTD 问题,这对实际应用有一定的参考价值。

关键词:改进蚁群算法;装配线物料配送;带最后期限时间窗的车辆路径问题

中图分类号:TH165.1 **文献标志码:**A **文章编号:**2095-0373(2017)03-0055-07

0 引言

在准时化生产模式下,需要对装配线上的各类物料实施多品种、小批量的准时配送。在正确的时间,将正确的物料配送至正确的工位是保证装配线顺利进行的关键^[1]。

针对物流配送车辆路径问题(VRP),许多学者根据实际情况提出了解决方案:杨斯淇^[2]针对生产车间 VRP 问题,建立了牵引车配送路线问题的数学模型,结合使用遗传算法和插入法,对传统的遗传算法做了改进,简化了求解过程;任星球^[1]建立了带缓存区配送方式下的牵引车配送路径模型,设计混合量子进化算法解决装配线 VRP 问题。针对装配线物料配送带最后期限时间窗的车辆路径问题(VRPTD)的研究,徐智慧等人^[3]建立了基于灾后物资供应的 VRPTD 模型,采用遗传算法求解,对应急物资的调度有指导意义;赵思敏^[4]考虑粮食应急物流道路阻塞与时间的相关函数,建立以满意度最大为目标的 VRPTD 模型,通过遗传算法与蚁群算法的演示程序将两个算法进行对比,结果表明相同的前提条件下,蚁群算法在求解速度、最优解质量和算法稳定性等方面优于遗传算法。吴隽等^[5]针对物流配送中的带时间窗的车辆路径问题(VRPTW),提出了一种改进的最大最小蚁群算法,引入了局部搜索策略二次优化(2-opt),算法寻优过程得到了优化;Du et al^[6]提出了贪婪策略与蚁群算法相结合的自适应蚁群算法,效果优于传统的自适应蚁群算法。

本文采用改进蚁群算法对装配线 VRPTD 问题进行了优化研究,通过信息素动态更新、启发函数 η_{ij} 的改进设计,使算法具有自适应性,在一定程度上克服了传统蚁群算法在遍历寻优过程中容易出现停滞和陷入局部最优解的缺点。

1 牵引车物料配送的 VRPTD 数学模型

已知缓存区、工位和道路条件,在每个工位段需求和服务时间窗已知的条件下,对 m 辆额定载重量为 Q (单位 kg)的运输车辆, n 个工位,确定运输车辆的最佳配送路线。设 n 个工位集合 $V = \{v_1, v_2, \dots, v_n\}$, V 中两个工位集合 $D = \{D_{ij} | v_i, v_j \in V\}$, 工位 i, j 坐标分别为 (x_i, y_i) 、 (x_j, y_j) , 两工位间距离为

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

收稿日期:2017-03-26 责任编辑:车轩玉 DOI:10.13319/j.cnki.sjztdxxxbzrb.2017.03.11

基金项目:河北省自然科学基金(F2013210109);河北省高等学校创新团队领军人才培养计划(LJRC018);河北省教育厅自然科学基金(QN2014151)

作者简介:刘凯(1988-),男,硕士研究生,主要从事企业信息化与协调管理技术的研究。E-mail: lk1256229856@126.com

刘 凯,牛江川,申永军,等.基于改进蚁群算法的装配线 VRPTD 问题研究[J].石家庄铁道大学学报:自然科学版,2017,30(3):55-61.

模型满足以下约束条件:

(1) 物料配送不考虑车间内通行路线的限制,各个工位间的距离为理想的直线距离。

(2) 每个工位的需求量均不超过运输车辆的额定载重量,即 $\max(q_i) \leq Q$, 其中第 i 个工位的需求量为 $q_i (i=1, \dots, n)$, q_i 的单位为 kg。

(3) 每一个工位所需的物料只能由一台车配送完成,每辆车可以配送多个工位,且所有车辆配送路线均起止于缓存区。

(4) 每个工位都有最后服务期限的时间窗 $[0, b_i]$, 物料配送必须在此时间窗内完成,否则将受到惩罚。

在建立数学模型之前,需要确定缓存区使用车辆数,使用文献[1]中的公式来确定需要的车辆数 m

$$m = \sum_{i=1}^n \frac{q_i}{aQ}, a \in (0, 1) \quad (2)$$

式中, $q_i (i=1, \dots, n)$ 为第 i 个工位的需求量; Q 为车辆的额定载重量; a 取 0.85, 约束条件越多, 车辆装卸物料越复杂, a 值越小^[7]。

引入 0-1 变量, y_{ik} 表示工位 i 所需要的物料由运输车 k 配送。当工位 i 由运输车 k 配送时 $y_{ik} = 1$, 否则 $y_{ik} = 0$ 。

在满足上述约束条件下, 确定运输车辆的最佳配送路线, 使总行驶距离最小, 确定总行驶距离为目标函数 f , 数学模型如下

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

式中, 当运输车 k 从 i 行驶到 j 时 $x_{ijk} = 1$, 否则为 0。并满足以下约束条件

$$\sum_{k=1}^m y_{0k} = m \quad (4)$$

式中, 下标 0 表示缓存区, 式(4)保证了有 m 辆运输车从缓存区出发, 进行物料配送。

$$\sum_{k=1}^m y_{ik} = 1, \quad i = 1, \dots, n \quad (5)$$

式(5)保证了每一个工位都有一辆车负责物料配送。

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

$$\sum_{j=0}^n \sum_{k=1}^m x_{jik} = 1, \quad i = 0, \dots, n \quad (7)$$

式(6)、式(7)保证每个工位只有一辆车执行物料配送任务。

$$\sum_{i=1}^n q_i y_{ik} \leq Q, \quad k = 1, \dots, m \quad (8)$$

式中, Q 为每辆车的额定载重量。车辆的运送总质量应不超过车辆的最大载重, 式(8)保证每条路径上车辆的装载量约束。

$$\sum_{i=0}^n x_{ipk} - \sum_{j=0}^n x_{pj k} = 0, \quad k = 1, \dots, m; p = 1, \dots, n \quad (9)$$

式中, p 为 n 个工位中任意一个工位, 式(9)保证每辆车到达某工位后仍从该工位离开。

$$s_{ik} + t_{ij} - K(1 - x_{ijk}) \leq x_{jk}, \quad \forall i, j \in (1, \dots, n); k = 1, \dots, m \quad (10)$$

式中, s_{ik} 为运输车 k 在 s_{ik} 时刻配送物料至工位 i , $s_{ik} \in [0, b_i]$; K 为很大的正数。式(10)保证车辆不早于 $s_{ik} + t_{ij}$ 到达工位 j 。

$$t_{ij} = \frac{d_{ij}}{u}, \quad i = 1, \dots, n; \quad j = 1, \dots, n \quad (11)$$

式中, u 为运输车的恒定速度; t_{ij} 为运输车从工位 i 到工位 j 的行驶时间。

2 VRPTD 问题蚁群算法的设计与改进

2.1 蚁群算法设计

设蚂蚁的编号为 $k(k=1, \dots, n)$, 蚂蚁数量为 n , 蚂蚁数量的确定需要在算法全局搜索能力和收敛速度之间做出权衡, n 值增大, 会增强全局搜索能力, 但代价是算法的收敛速度下降; 而如果 n 值较小, 蚂蚁经过的路径上的信息素浓度会越来越低, 极易使算法产生过早停滞现象, 仿真实验显示, 选取蚂蚁数量应与工位数量相近^[8], 取蚂蚁数量为工位数量。 t 时刻在工位 (i, j) 上的信息素浓度 $\tau_{ij}(t)$, 在算法初始时, $\Delta\tau_{ij}(0)=0$, 即初始时刻各条路径上的信息素均相同, 各条路径上的信息素浓度决定蚂蚁的转移方向^[9], 蚂蚁在 t 时刻由 i 转移到 j 的概率 $P_{ij}^k(t)$ 如下

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{s \in \text{allowed}_k} [\tau_{is}(t)]^\alpha [\eta_{is}(t)]^\beta}, & j \in \text{allowed}_k \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

式中, 信息启发因子 α 和期望启发因子 β 分别表示信息素和期望值的相对重要程度, α 值越大, 表示信息素浓度在状态转移中起的作用越大, 同样 β 值越大, 表示启发函数在状态转移中起的作用越大, 即蚂蚁会以较大的概率转移到距离较短的工位。

$$\text{allowed}_k = \{V - \text{Tabu}_k\} \quad (13)$$

式中, allowed_k 为蚂蚁 k 下一步能选择的工位集合; $V = \{v_1, v_2, \dots, v_n\}$ 为工位集合。开始时, allowed_k 中有 $(n-1)$ 个元素, 随着迭代次数的增加, allowed_k 中的元素逐渐减少, 直至为空, 即表示此时蚂蚁已遍历完集合中所有工位。禁忌表 Tabu_k 存储并记录蚂蚁 k 走过的工位。

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (14)$$

式(14)为启发函数, 表示蚂蚁从工位 i 转移到工位 j 的期望程度, η_{ij} 越大, $P_{ij}^k(t)$ 越大。

蚂蚁每完成一次循环, 需要考虑信息素的挥发, 蚂蚁在释放信息素的同时, 各个工位路径上的信息素也在逐渐消失, 需要实时更新各个工位路径上的信息素^[10]。

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^n \Delta\tau_{ij}^k \quad (15)$$

式中, $\tau_{ij}(t+1)$ 为在 $(t+1)$ 时刻工位 (i, j) 上的信息素浓度; $\Delta\tau_{ij}^k$ 为蚂蚁 k 从工位 i 到工位 j 路径上释放的信息素浓度, $\sum_{k=1}^n \Delta\tau_{ij}^k$ 为所有蚂蚁从工位 i 到工位 j 释放的信息素浓度之和; ρ 为信息素挥发系数, $(1-\rho)$ 为信息素残留因子, 为了防止信息素无限积累, $\rho \in (0, 1]$ 。

2.2 算法改进

2.2.1 信息素动态更新改进

蚁群算法中信息素挥发系数的大小直接关系到蚁群算法的全局搜索能力和收敛速度^[11]。当问题规模较大时, 由于信息素挥发系数的存在, 从未被搜索到的路径上的信息素浓度会逐渐减小到接近于 0。当信息素挥发系数过小时, 路径上的信息素浓度增大会使以前搜索过的路径再次被选择的可能性增大, 从而降低了算法的全局搜索能力, 增大信息素挥发系数虽然可以提高算法的全局搜索能力, 但又会使算法的搜索速度降低^[8]。因此, 通过改变信息素挥发系数, 提出了一种信息素挥发系数自适应变化的蚁群算法, 希望通过自适应的改变信息素挥发系数提高算法的全局搜索能力。

当求得的最优解在连续若干代内没有明显改进时, 按式(16)对信息素挥发系数进行更新。

$$\rho = \begin{cases} 0.95\rho, & 0.95\rho \geq \rho_{\min} \\ \rho_{\min}, & \text{otherwise} \end{cases} \quad (16)$$

式中, 信息素挥发系数 ρ 应在初始给定最大值和最小值之间取值; 参数 0.95 是信息素挥发系数更新系数, 每次下降 5%, 直到小于信息素挥发系数的最小值 $\rho_{\min}$, $\rho_{\min}$ 可以防止因信息素挥发系数过小而降低算法的收敛速度。

本算法与基本蚁群算法相比取消了信息素的局部更新即二次优化 2-opt 操作,而是进行信息素的全局更新以此来提高算法的收敛速度。

2.2.2 启发函数的改进

由式(14)可知,两工位间的距离越小,可见度 η_{ij} 越大,状态转移概率 $P_{ij}^k(t)$ 越大,蚂蚁从工位 i 到工位 j 的期望度也越高。如果将每个工位的接受服务的时间窗考虑在内,改进后的启发函数如下

$$\eta_{ij} = \frac{1}{d_{ij} \{b_i - (s_{jk} - t_{ij})\}} \quad (17)$$

改进后的可见度,有助于提高算法解决带最后期限时间窗的车辆路径问题的全局性。

2.3 算法实现流程

基于上述设计,改进后的蚁群算法求解 VRPTD 问题步骤如下,其算法流程如图 1 所示。

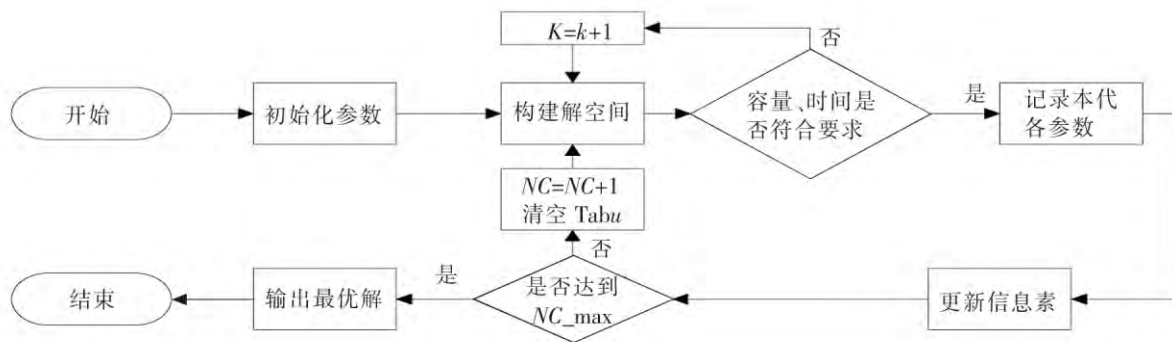


图 1 改进蚁群算法流程图

(1) 初始化参数。开始计算之前,需要对相关参数进行初始化,如蚂蚁数目 n 、车辆数目 m 、重要度系数 α 、能见度系数 β 、权重系数 γ 、挥发系数 ρ 、最小挥发系数 $\rho_{\min}$ 、最大迭代次数 $NC_{\max}$ 、车辆载重量 Q 、车辆装载量 $load_w$ 等。

(2) 构建解空间。将未服务的工位放入 $V = \{v_1, v_2, \dots, v_n\}$ 中,对于蚂蚁 $k(k=1, \dots, n)$ 按照公式(12)计算下一个要服务的工位,直到所有蚂蚁访问完所有工位。

(3) 容量和时间判断。设置蚂蚁的禁忌表索引号 $k=1$,每服务一个工位,判断容量和时间是否符合要求,符合则将服务过的工位置于当前 $tabu(k)$ 中,不符合则跳转到第 2 步。

(4) 更新信息素。计算各个蚂蚁走过的路径长度,计算当前迭代次数中的最短路径。同时根据公式(15)、公式(16)、公式(16)对各个工位路径上的信息素进行更新。

(5) 判断终止条件。当迭代计数器 $NC < NC_{\max}$ 时, $NC = NC + 1$,清空禁忌表,跳转到第 2 步。否则,程序结束,输出最优解。

3 算法测试及结果分析

3.1 算法测试

为了测试改进后的蚁群算法求解 VRPTD 问题的效果,分别用基本蚁群算法和改进后的蚁群算法求解 25 个客户点的 Solomon 基准实例^[12]。Solomon 例库有 C 型、R 型、RC 型 3 种,它们之间的区别在于客户点坐标的设置方式和时间窗的设置不同。C 型例库按照结构性方式来设置客户坐标,R 型例库则是以均匀分布的方式来设置客户坐标,而 RC 型例库则是按以上两种例库混合的方式来设置客户坐标^[13-14]。

采用 Matlab 对改进蚁群算法进行测试,各个参数设置如下:蚂蚁数目 $n=25$,车辆数目 m 依具体情况而定,实例 C101、R101、RC101 分别对应的车辆数目为 3、2、15,重要度系数 $\alpha=3$,能见度系数 $\beta=2$,权重系数 $\gamma=2$,挥发系数 $\rho=0.5$,最小挥发系数 $\rho_{\min}=0.05$,最大迭代次数 $NC_{\max}=100$,车辆载重量 $Q=$

200,车辆装载量 $load_w=0$ 。测试结果如表 1 所示。

表 1 Solomon C 型、R 型、RC 数据集测试结果对比

实例	基本蚁群算法		改进蚁群算法	
	运行时间/s	最短距离/m	运行时间/s	最短距离/m
C101	16.380 0	175.464 5	16.162 0	174.832 0
R101	17.550 0	184.251 1	16.895 0	184.059 0
RC101	26.927 0	770.846 2	26.754 0	770.484 7

3.2 算例实现

由于 Solomon 测试例库中,C 型数据集、R 型数据集、RC 型数据集每个工位处的服务时间分别为 90、10、10,不是随机变化的。故采用如下算例,进一步对改进算法的有效性进行验证。具体算例数据如表 2 所示。

表 2 某装配线工位点数据

编号	横坐标/m	纵坐标/m	需求量/kg	截至时间/min	服务时间/min
1	330.4	231.2	0	0	0
2	363.9	131.5	90.0	290.0	90.0
3	417.7	224.4	40.0	240.0	40.0
4	371.2	139.9	60.0	260.0	60.0
5	348.8	153.5	70.0	270.0	70.0
6	332.6	156.6	70.0	270.0	70.0
7	323.8	129.9	40.0	240.0	40.0
8	419.6	104.4	20.0	220.0	20.0
9	431.2	79.0	40.0	240.0	40.0
10	438.6	17.0	70.0	270.0	70.0
11	300.7	197.0	60.0	260.0	60.0
12	256.2	175.6	40.0	240.0	40.0
13	278.8	149.1	40.0	340.0	40.0
14	238.1	167.6	40.0	340.0	40.0
15	133.2	69.5	20.0	320.0	20.0
16	371.5	167.8	80.0	380.0	80.0
17	391.8	217.9	50.0	350.0	50.0
18	406.1	237.0	30.0	230.0	30.0
19	378.0	221.2	50.0	250.0	50.0
20	367.6	257.8	50.0	250.0	50.0
21	302.9	183.8	50.0	350.0	50.0
22	426.3	293.1	50.0	350.0	50.0
23	342.9	190.8	60.0	260.0	60.0
24	350.7	237.6	70.0	270.0	70.0
25	339.4	264.3	50.0	250.0	50.0
26	343.9	320.1	40.0	340.0	40.0
27	293.5	324.0	40.0	240.0	40.0
28	314.0	355.0	60.0	260.0	60.0
29	254.5	235.7	70.0	270.0	70.0
30	277.8	282.6	50.0	250.0	50.0
31	237.0	297.5	30.0	230.0	30.0

蚂蚁数目 $n=31$, 车辆数目 $m=9$, 算法其它初始设置与上例相同。经过仿真分析, 分析结果如表 3 所示。

表 3 算例仿真结果对比

算法	最短距离/m	运行时间/s	访问工位段顺序	
基本蚁群算法	2 708.500	27.160	车 1:1-24-20-25-1	车 5:1-29-31-27-28-1
			车 2:1-11-21-6-1	车 6:1-30-26-22-19-1
			车 3:1-23-16-4-1	车 7:1-3-18-17-13-12-1
			车 4:1-2-5-7-1	车 8:1-14-15-10-9-8-1
改进蚁群算法	2 590.300	27.004	车 1:1-24-20-19-1	车 5:1-25-26-28-27-1
			车 2:1-11-21-13-12-1	车 6:1-16-17-3-18-1
			车 3:1-2-4-7-1	车 7:1-8-9-10-15-14-1
			车 4:1-23-5-6-1	车 8:1-29-31-30-22-1

采用基本蚁群算法, 迭代过程中最短距离、平均距离变化和路径规划结果, 如图 2 和图 3 所示。

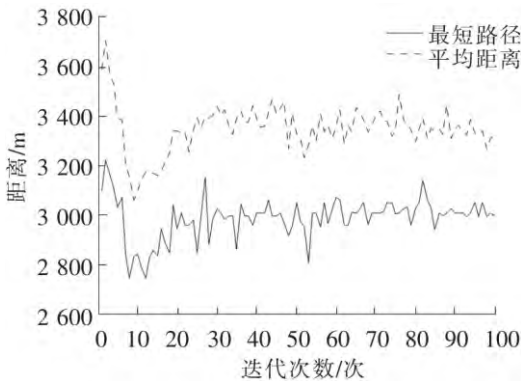


图 2 基本蚁群算法最短距离、平均距离变化

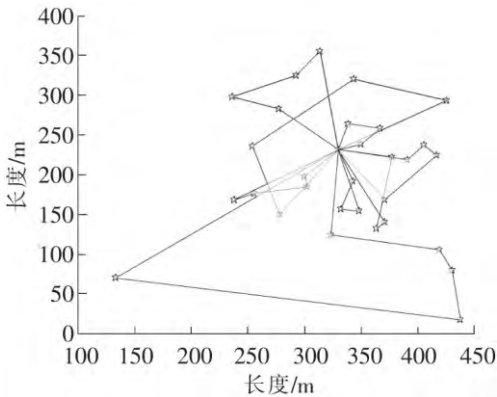


图 3 基本蚁群算法路径规划结果

由图 2 可以看出, 传统的蚁群算法求解 VRPTD 问题时, 最终求得的最短距离在 2 700 m 左右, 在第 10 次迭代的时候, 找到了最优解, 求解效率较高。由图 3 可以看出, 基本蚁群算法求得的路径规划结果有重复的路径出现, 求解结果还有优化的空间。

采用改进蚁群算法, 迭代过程中最短距离、平均距离变化和路径规划结果, 如图 4 和图 5 所示。

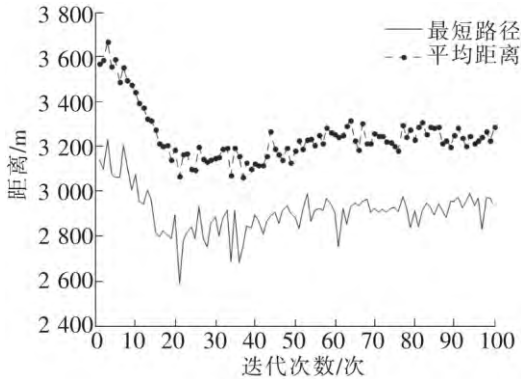


图 4 改进蚁群算法最短距离、平均距离变化

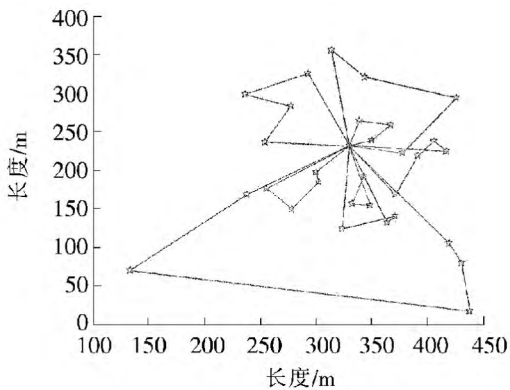


图 5 改进蚁群算法路径规划结果

由图 4 可以看出, 改进后的蚁群算法求得的最短距离在 2 600 m 左右, 缩短了近 100 m, 第 20 次迭代的时候找到了最优解, 寻优效率没有改进前高, 但寻优结果较改进前好。由图 5 可以看出, 改进后蚁群算法求得的路径规划结果避免了重复路径的出现, 进一步优化了求解结果。

算法改进前后最短路径对比, 如图 6 所示。

由图6知,改进后的蚁群算法求解结果优于基本蚁群算法求解结果。迭代初期,改进后的蚁群算法寻优效率并没有传统蚁群算法高,但是随着迭代次数的增加,改进后的蚁群算法由于对当前最优路径上信息素的充分利用,扩大了解的搜索空间,提高了算法的全局遍历性,并没有陷入局部最优解的死循环。

4 结论

通过对传统蚁群算法信息素更新和可见度设计方面进行改进,求解装配线物料配送优化问题。并通过实例对算法改进前后进行了对比分析,分析结果表明,改进后的蚁群算法在求解装配线 VRPTD 问题时不仅对解决蚁群算法的停滞问题非常有效,而且提高了运算精度,增大了收敛到全局最优解的可能性,有效地解决搜索停滞、早熟问题,对生产实际有一定的指导意义。

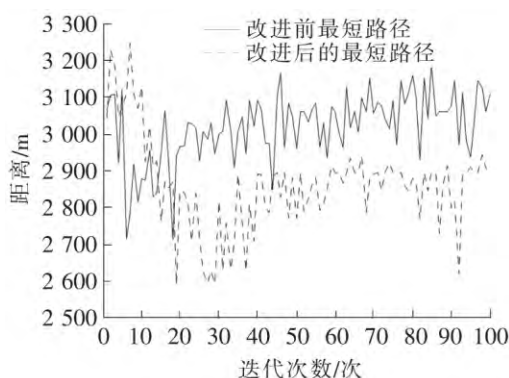


图6 最短路径前后对比

参考文献

- [1]任星球. 制造企业装配线物料准时配送优化研究[D]. 杭州:浙江工业大学,2011.
- [2]杨斯淇. 基于遗传算法的制造企业生产物流牵引车配送路线优化研究[D]. 长春:吉林大学,2008.
- [3]Zhihui X, Zhongning F. Study on emergency response of vehicle routing based on multi-commodity dispatch VRPTD model[C]//Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference on. IEEE, 2011: 568-571.
- [4]赵思敏. 粮食应急物流系统的网络构建及路径优化[D]. 武汉:武汉理工大学,2011.
- [5]吴隽,陈定方,李文锋,等. 基于改进蚁群算法的有时间窗车辆路径优化[J]. 湖北工业大学学报,2008(3):9-12.
- [6]Du D P, Zu Y R. Ubiquitous Computing Application and Wireless Sensor [M]. Berlin: Springer Netherlands, 2015: 663-670.
- [7]邢文训,谢金星. 现代优化计算方法[M]. 北京:清华大学出版社,1999.
- [8]石华瑞. 改进的蚁群算法在实际 VRP 中的应用研究[D]. 济南:山东大学,2012:33-34.
- [9]Yi Y, Lin X, Sheng K, et al. Intelligent computing theories and methodologies[C]. Berlin:[s. n.], 2015:11-22.
- [10]段海滨. 蚁群算法原理及应用[M]. 北京:科学出版社,2005.
- [11]吴小菁. 基于挥发系数的自适应蚁群算法[J]. 福建金融管理干部学院学报,2010,1(1):54-58.
- [12]Solomon Marius M. Algorithms for the vehicle routing and scheduling problem with time window constraints[J]. Operations Research, 1987, 35(2): 254-265.
- [13]Abdulkader M M S, Gajpal Y, Elmekawy T Y. Hybridized ant colony algorithm for the multi compartment vehicle routing problem[J]. Applied Soft Computing, 2015, 37:196-203.
- [14]刘凯,牛江川,申永军,等. 基于遗传粒子群算法的堆垛机作业路径优化[J]. 石家庄铁道大学学报:自然科学版,2016, 29(2):67-71.

Study of Assembly Line VRPTD Problem Based on Improved Ant Colony Algorithm

Liu Kai, Niu Jiangchuan, Shen Yongjun, Han Yanjun

(School of Mechanical Engineering, Shijiazhuang Tiedao University, Shijiazhuang 050043, China)

Abstract: The distribution of materials for VRPTD(assembly lines with the deadline time windows vehicle routing problem) is studied by improved ant colony algorithm. By pheromone dynamically updated design, the improved ant algorithm is adaptive, which overcomes the shortcomings of the traditional ant colony optimization algorithm that traversal process is prone to stagnation and falling into local optima. The design of the heuristic function of the visibility, which based on further improvement, had improved the global search capability. The simulation results show that the algorithm can solve the problem of assembly line VRPTD, which has a certain reference value for practical application.

Key words: improved ant colony algorithm; assembly line material feeding; VRPTD