

文章编号:1007-5321(2019)01-0016-06

DOI:10.13190/j.jbupt.2018-111

基于动态聚类混合拓扑结构粒子群算法的 PDVRPTF

杨福兴¹, 胡智超¹, 孔继利²

(1. 北京邮电大学 自动化学院, 北京 100876; 2. 北京邮电大学 现代邮政学院, 北京 100876)

摘要: 经典物流配送模型的目标、约束条件不够全面,在实际应用中存在一定缺陷,对此,构建了时间窗和油耗取送一体化的物流配送路径优化模型(PDVRPTF). 设计了一种基于 k -medoids 动态聚类混合拓扑结构粒子群算法,解决了经典粒子群算法在求解此类模型时容易陷入局部最优解的问题. 仿真结果表明,改进型粒子群算法能很好地跳出局部最优解,并快速收敛于全局最优解,且该算法可有效求解物流配送路径优化的问题.

关键词: 时间窗; 取送一体化; 油耗; 动态聚类; 混合拓扑结构; 粒子群算法

中图分类号: TP391.9; U116.2

文献标志码: A

PDVRPTF Based on Dynamic Clustering Hybrid Topological Structure Particle Swarm Optimization

YANG Fu-xing¹, HU Zhi-chao¹, KONG Ji-li²

(1. School of Automation, Beijing University of Posts and Telecommunications, Beijing 100876, China;

2. School of Modern Post, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Aiming at the problem that the classic logistics distribution model considers the target, the constraints are not comprehensive enough and there are certain defects in the practical application, a integrated pickup and distribution vehicle routing problem on the basis of the classical model considering time window and fuel consumption (PDVRPTF) is constructed. Hybrid topological structure of particle swarm optimization based on k -medoids dynamic clustering is designed, which solves the problem that classical particle swarm optimization is easy to fall into local optimal solution when solving such models. The simulation results show that the improved particle swarm optimization can jump out of the local optimal solution quickly and converge to the global optimal solution quickly, which solve the logistics distribution path optimization problem effectively.

Key words: time window; integrated pickup and distribution; fuel consumption; dynamic clustering; hybrid topological structure; particle swarm optimization

经典物流配送模型的目标和约束条件不够全面,在实际应用中存在一定缺陷,对此,提出了时间窗和油耗取送一体化物流配送路径优化模型(PDVRPTF),integrated pickup and distribution vehicle rou-

ting problems considering time window and fuel consumption),并将其用于解决实际问题,这样很大程度上减少了车辆的空载率和油耗,不但能够为企业降低成本,也能减少燃油资源的消耗和污染气体的

收稿日期: 2018-06-14

基金项目: 国家自然科学基金项目(71772010); 北京邮电大学青年科研创新计划专项-人才项目(2017RC26)

作者简介: 杨福兴(1964—),男,教授,博士生导师.

通信作者: 胡智超(1993—),男,硕士生, E-mail:hzc199308@163.com.

排放,有利于绿色物流的发展.

对于求解路径优化问题,粒子群优化算法实现简单,计算效率高,在搜索前期收敛速度较快,但后期容易陷入局部最优解. 针对这一问题,国内外很多学者提出了很多改进方式,如加入收敛因子^[1]、采用不同的拓扑结构^[2]和加入惯性因子^[3]等. 笔者将 k -medoids 聚类分析算法与粒子群结合,聚类之后采用 all-ring 型拓扑结构进行信息交流. 利用 Rastrigin 函数验证算法的可行性,并利用 Solomon 标准数据集测试模型的实用性.

1 PDVRPTF 问题模型

1.1 问题描述

对 PDVRPTF 问题的基本描述如下:假设有 n 个客户,每个客户的位置和需求已知,配送中心有 K 辆额定车载量为 Q 的车. 车辆配送优化模型要求从配送中心选出 $k(k=1,2,\dots,K)$ 辆车服务 n 个有送货或取货需求的客户,要求使用尽量少的车辆为所有的客户服务,每辆车在为客客户取送货途中车载量不能超过额定车载量 Q . 在满足用户要求下,制定合理的车辆配送路线,使路程、时间和油耗等综合成本达到最少.

1.2 模型重要约束分析

在建立 PDVRPTF 问题模型之前,需要对以下几个约束条件分别进行讨论.

1.2.1 时间窗约束

每个客户点都有规定的时间窗 $[E_i, L_i]$,其中接受服务的最早时间为 E_i ,最晚时间为 L_i ,且每个点的服务时间为 S_i .

如果车辆 k 到达 i 点的时间 A_i^k 小于 E_i ,则车辆 k 要等待的时间为 W_i^k , $W_i^k = \max\{0, (E_i - A_i^k)\}$;当车辆 k 到达 i 点的时间 A_i^k 在 $[E_i, L_i]$ 区间,则直接服务 i 客户,不用付出等待时间;车辆 k 到达 i 点的时间 A_i^k 大于 L_i ,则其延迟的时间为 T_i^k , $T_i^k = \max\{0, (A_i^k - L_i)\}$.

车辆 k 离开 i 的时间 D_i^k 为

$$D_i^k = \begin{cases} A_i^k + W_i^k + S_i, & A_i^k < E_i \\ A_i^k + S_i, & A_i^k \geq E_i \end{cases} \quad (1)$$

1.2.2 额定车载量的取送一体化

车辆 k 的额定车载量约束为 Q ,取送一体化的服务方式需要保证车辆在路径的每一个点的车载量都不大于 Q . 设用户 i 的送货量为 c_i ,取货量为 p_i ,

若车辆 k 的行驶路径为 U_0^k ,该行驶路径中车辆 k 需要服务的客户总数为 U^k ,则车辆 k 需要满足以下约束:

$$\sum_{i=1}^{U^k} c_i \leq Q \quad (2)$$

$$\sum_{i=1}^{U^k} p_i \leq Q \quad (3)$$

$$\sum_{i=1}^m p_i + \sum_{i=m+1}^{U^k} c_i \leq Q, m = 1, 2, \dots, U^k \quad (4)$$

式(2)表示保证路径 U_0^k 所有客户的送货量之和不超过额定车载量 Q ;式(3)表示保证路径 U_0^k 的全部取货量之和不超过额定车载量 Q ;式(4)表示保证车辆 k 在用户 m 处卸货并完成装货后的车载量不超过额定车载量 Q .

1.2.3 油耗约束

车辆的油耗不仅与车辆行驶的路程有关,而且与车辆发动机负荷率、行驶阻力、车辆的行驶状况、保养情况等因素相关. 汽车等速行驶百公里油耗^[4]为

$$R_s = \frac{Pb_0}{1.02v\rho g} = \frac{C_0 F_0 b_0}{\tau} \quad (5)$$

其中: P 为汽车行驶过程中遇到的阻力功率, b_0 为燃油消耗率, v 为车辆的行驶速度, ρ 为燃油的密度, g 为重力加速度, C_0 为常数, F_0 为车行驶途中的阻力, τ 为传动效率. 由式(5)可知,汽车百公里燃油消耗量正比于等速行驶时的行驶阻力和燃油消耗率.

本模型假设车辆的行驶状况、保养情况和路面的摩擦等因素一定,汽车的载重量越大,所遇到的摩擦阻力越大,燃油消耗量也就越大. 笔者仅考虑车辆的行驶里程和车辆的车载量对油耗的影响,且假设油耗和车辆的行驶里程、载量存在一定的线性关系,因此可利用“载重估计法”对油耗进行计算. 若 r^* 为车辆满载时行驶单位距离的油耗, r_0 为车辆空载时行驶单位距离的油耗,则车辆 k 在对路线 U_0^k 中第 m 个客户进行服务后车辆载重量 q_m 的计算公式为

$$q_m = \sum_{i=1}^m p_i + \sum_{i=m+1}^{U^k} c_i \quad (6)$$

车辆在车载量为 q_m 时行驶单位距离的油耗 $r(q_m)$ 计算公式为

$$r(q_m) = r_0 + (r^* - r_0) \frac{q_m}{Q} \quad (7)$$

在车载量为 q_m 时,车辆从点 i 行驶到点 j 的路程为 d_{ij} ,消耗的油耗 R_{ij} 计算公式为

$$R_{ij} = r(q_m)d_{ij} = \left[r_0 + (r^* - r_0) \frac{q_m}{Q} \right] d_{ij} \quad (8)$$

将式(8)引入 PDVRPTF 模型,将引导车辆先服务距离近、配送量大、货量小的客户,以降低车辆的油耗,使得车辆 k 的行走路径更加合理。

1.3 PDVRPTF 模型构建

目标函数:

$$\begin{aligned} \min Z = & c_1 \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^{N+1} (x_{ij}^k d_{ij}) + \\ & c_2 \sum_{k=1}^K \sum_{i=1}^N (Y_i^k W_i^k) + c_3 \sum_{k=1}^K \sum_{i=1}^N (Y_i^k T_i^k) + \\ & c_4 \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^{N+1} R_{ij}^k \end{aligned} \quad (9)$$

除 1.2 节中所给出的约束条件(1)~(8)之外,该模型的其他约束条件如下。

$$x_{ij}^k = \begin{cases} 1, & \text{车辆 } k \text{ 从路线 } U_0^k \text{ 的 } i \text{ 行驶到 } j \\ 0, & \text{其他 } i \neq j, k \in K \end{cases} \quad (10)$$

$$Y_i^k = \begin{cases} 1, & \text{车辆 } k \text{ 服务路线 } U_0^k \text{ 的客户 } i \\ 0, & \text{其他 } k \in K \end{cases} \quad (11)$$

$$D_i^k > A_i^k > D_{i-1}^k \quad (12)$$

$$P_{ij}^k = \frac{d_{ij}^k}{v} \quad (13)$$

$$A_i^k = D_{i-1}^k + P_{(i-1)i}^k \quad (14)$$

$$\sum_{i=1}^N x_{ih}^k - \sum_{j=1}^{N+1} x_{hj}^k = 0, h \in N, k \in K \quad (15)$$

$$\sum_{k=1}^K Y_i^k = 1, i = 1, 2, \dots, N \quad (16)$$

$$\sum_i x_{ij}^k = Y_j^k, j = 0, 1, 2, \dots, N, \forall k \quad (17)$$

$$\sum_{j=1}^N x_{ij}^k = Y_i^k, i = 0, 1, 2, \dots, N, \forall k \quad (18)$$

式(9)是模型的目标函数,即在考虑车辆行走总路程、等待时间、延迟时间、油耗等因素的情况下,使总成本达到最小,其中 T_i^k 表示车辆 k 在用户 i 处的延迟时间, d_{ij} 表示从 i 行驶到 j 的路程, c_1 、 c_2 、 c_3 、 c_4 分别表示车辆行驶单位距离、等待单位时间、延迟单位时间、单位油耗所花费的成本;式(10)、式(11)为车辆 k 经过客户点的记录函数;式(12)表示车达到客户 i 的时间小于离开客户 i 的时间,且大于离开客户 $(i-1)$ 的时间;式(13)中 P_{ij}^k 表示车辆从客户 i 到客户 j 的行驶的时间,其中 v 为行驶速度, $d_{(i,j)}^k$ 表示车辆 k 从 i 行驶到 j 的行驶路程;式(14)表示车 k 到达

某一点的时间为上一点的离开时间与路上消耗时间之和;式(15)表示到达某一点的车 k 只能从该点出发;式(16)~(18)为车辆 k 到达某一点的唯一性约束,即每一个点仅由一辆车服务。

2 基于混合拓扑结构 k -medoids 动态聚类的粒子群算法

在粒子群优化算法(PSO, particle swarm optimization)中,粒子的状态包括 2 个向量:速度和位置。设搜索空间为 d 维,第 i 个粒子的速度向量 $V_i = [v_{i1}, v_{i2}, \dots, v_{id}]$,位置向量 $X_i = [x_{i1}, x_{i2}, \dots, x_{id}]$,第 i 个粒子迄今搜索到的最优解 $p_b = [p_{i1}, p_{i2}, \dots, p_{id}]$,所有粒子目前搜索的全局最优解为 $g_b = [g_{i1}, g_{i2}, \dots, g_{id}]$,粒子的速度更新公式如式(19)所示,位置更新公式如式(20)所示。

$$V_{i+1} = \omega V_i + b_1 \text{rand}(p_b - X_i) + b_2 \text{rand}(g_b - X_i) \quad (19)$$

$$X_{i+1} = X_i + V_{i+1} \quad (20)$$

其中: ω 为惯性权重,表示粒子上一位置的速度对当前位置搜索速度的影响程度; b_1 、 b_2 为加速系数,分别表示粒子受个体认知和社会模式影响程度;rand 是 0~1 之间的随机数。

针对取送结合的考虑油耗和时间窗的车辆配送路径优化模型,笔者设计改进型粒子群算法,用于求解物流配送车辆路径优化问题。

2.1 动态种群分割策略

粒子群算法中粒子在不断搜索迭代的过程中收敛于全局极值 g_{bi} 和个体极值 p_{bi}^t 之间,考虑到算法的粒子具有聚集性^[5],因此将聚类算法与粒子群算法有效结合。基于粒子的位置信息,以粒子间的距离作为度量,以各个粒子到各自聚类中心的距离之和最小化为目标,对种群进行 k -medoids 聚类。本文中 k -medoids 聚类利用 k -means++ 选取 k 个粒子作为聚类中心 (a_1, a_2, \dots, a_k) ,这样选取的聚类中心粒子距离较远,差异性较大,解决 k -means 算法中初始粒子选择不当导致簇效果不佳或收敛速度慢的问题;由于初选的聚类中心距离远,可保持簇间的差异性,有利于跳出局部最优解。 $d^2(x, y)$ 表示 2 个不同粒子之间欧几里得距离的平方,此处 x 、 y 分别表示 2 个不同的粒子,计算公式如式(21)所示,聚类迭代使得簇内误差平方和(E)最小,计算公式如式(22)所示。

$$d^2(x, y) = \sum_{j=1}^d (x_j - y_j)^2 = \|x - y\|_2^2 \quad (21)$$

$$E = \sum_{i=1}^n \sum_{j=1}^k w^{(i,j)} = \|x^{(i)} - u^{(j)}\|_2^2 \quad (22)$$

其中 $u^{(j)}$ 为簇 j 的聚类中心点, 如果 $x^{(i)}$ 属于簇 j , 则 $w^{(i,j)} = 1$, 否则 $w^{(i,j)} = 0$.

基于动态种群分割策略的聚类步骤如下:

步骤 1 初始化空集合 M , 储存选定的 k 个聚类粒子点, 从初始粒子中选第一个中心点 $u^{(j)}$, 并加入集合 M , 对集合 M 外的任一粒子 $x^{(i)}$, 通过计算找到与其平方距离最小的粒子 $d^2(x^{(i)}, M)$, 并使用加权概率分布 $\frac{d^2(u^{(p)}, M)}{\sum_i d^2(x^{(i)}, M)}$ 随机选下一个中心点

$u^{(p)}$, 重复直到选定 k 个中心点;

步骤 2 根据 $d^2(x, y)$ 将每个粒子划分到距离最近的中心点 $u^{(j)}$ ($j \in \{1, 2, \dots, k\}$) 所代表的簇中, 用各簇的中心点替代原来的中心点;

步骤 3 重复步骤 2 的迭代过程, 直到各簇的中心点位置不变或达到预定迭代次数, 则聚类算法终止并输出聚类簇.

2.2 基于动态邻域的粒子速度

在粒子聚类成簇之后, 计算粒子的适应度, 在簇 j 中选取适应度值最优的粒子作为该簇的最优解, 并标记为 H_j , 比较簇 j 相邻簇最优解的大小, 并将邻簇最优解记为 L_b . 对簇 j 中所有的粒子 i 而言, 邻簇最优解的粒子记为 $L_{b,i}$, $L_{b,i} = (L_{b,i1}, L_{b,i2}, \dots, L_{b,id})$. 考虑到邻域最优解、自身最优解和全局最优解对粒子信息的影响, 改进后的粒子群算法速度更新公式如式(23)所示. 位置更新公式仍采用经典的位置更新公式, 见式(20).

$$V_{i+1} = \omega V_i + b_1 \text{rand}(p_b - X_i) + b_2 \text{rand}(g_b - X_i) + b_3 \text{rand}(L_{b,i} - X_i) \quad (23)$$

这种改进后的速度更新公式引进了邻簇最优解对粒子的影响, 解决了粒子群算法中粒子朝着初始粒子的全局最优 g_b 聚集而陷入局部最优解的问题.

2.3 拓扑结构

拓扑结构对算法的性能和精确度有很大影响, 目前研究的种群拓扑结构主要有全局、局部、4 类和金字塔.

考虑对粒子聚类后采用 all-ring 混合型拓扑结构, 聚类簇内采用 all 型拓扑结构, 粒子间相互连接, 聚类簇之间采用 ring 型拓扑结构连接. all-ring 拓扑结构增加了邻域最优解的影响, 解决了所有粒子仅

向当前种群最优粒子靠近而容易陷入局部最优解的问题, 且在提升求解精度的同时, 算法能保持较快的收敛速度.

2.4 自适应调整策略

定义 1 将第 i 簇记为 A_i , C_i 表示第 i 簇粒子的个数, f_{ij} 表示第 i 簇中第 j 个粒子的适应度的值; 对于目标函数求极大值的问题, 适应度函数取目标函数的相反数; 第 i 簇适应度平均值 F_i 的计算公式为: $F_i =$

$$\frac{\sum_{j=1}^{C_i} f_{ij}}{C_i};$$

由于粒子总共被划分为 k 簇, 则整个种群的适应度平均值 B 的计算公式为: $B = \frac{\sum_{i=1}^k \sum_{j=1}^{C_i} f_{ij}}{\sum_{i=1}^k C_i}$.

定义 2 设 ω_{\max} 为算法的最大惯性权重, ω_{\min} 为算法的最小惯性权重, 算法的惯性权重 ω 根据 ω_{\max} , ω_{\min} 和迭代次数进行调整.

粒子群算法在不同的搜索时期, 粒子处于不同的解位置. 粒子处于较好位置, 应该赋予较小的惯性权重, 增强该粒子周围搜索力度; 粒子处于较差位置, 应该赋予较大的惯性权重. 聚类后的簇之间的粒子存在较大的差异性, 根据每个簇的适应度平均值和种群适应度平均值的比较, 赋予粒子不同的惯性权重.

若 $F_i < B$, 说明 i 簇粒子处于较差的位置, 应该赋予较大的惯性权重, 以提高全局搜索能力, 防止陷入局部最优解. T_{\max} 表示最大迭代次数, l 表示当前迭代次数, ω 的变化方式为

$$\omega_i^{l+1} = \omega_i^l + (\omega_{\max} - \omega_{\min}) \left(\frac{T_{\max} - l}{T_{\max}} \right) \quad (24)$$

若 $F_i \geq B$, 说明该簇粒子处于较好的位置, 应该赋予较小的惯性权重, 以提高局部搜索能力, ω 变化方式为

$$\omega_i^{l+1} = \omega_i^l - (\omega_{\max} - \omega_{\min}) \left(\frac{T_{\max} - l}{T_{\max}} \right) \quad (25)$$

2.5 算法流程

改进型粒子群算法具体实现步骤如下.

1) 初始化粒子位置 x 、速度 v_0 , 初始种群包括粒子个数 n 、最大迭代次数 T_{\max} 、最大惯性权重 ω_{\max} 和最小惯性权重 ω_{\min} , 加速系数 b_1 、 b_2 、 b_3 和初始惯性权重 ω ;

2) 采用 2.1 中的聚类方法对粒子进行聚类, 确定 k 个簇, 第 i 个簇记为 A_i ;

3) 计算第 i 个粒子的适应度值 f_i , 记录当前各个体搜索到的最优 p_{bi} 和全局最优 g_b , 将第 i 簇中的最优粒子记为本簇的最优值 $Z_i, i = 1, 2, \dots, k$;

4) 计算每个簇的适应度平均值 F_i 和全局适应度平均值 B , 判断 F_i 和 B 之间的关系, 根据式(24)或式(25)调整惯性权重 ω ;

5) 对簇 i , 比较 Z_{i-1}, Z_{i+1} , 将较大的值作为当前簇的 L_b , 采用式(20)、式(23)调整粒子的位置和速度;

6) 计算粒子的适应度值, 并记录当前粒子的最优值 g_b , 判断是否达到最大迭代次数 T_{max} 或搜索到满意解, 满足条件则停止迭代; 否则重复步骤 2) ~ 6) 进入下一轮迭代。

3 仿真测试分析与验证

3.1 Rastrigin 函数实验

选定 PSO 和笔者设计的基于混合拓扑结构 k -medoids 动态聚类的 PSO (KPSO), 运用 Matlab 建立仿真模型, 求解 Rastrigin 函数。

Matlab 仿真求解函数迭代图如图 1 所示。

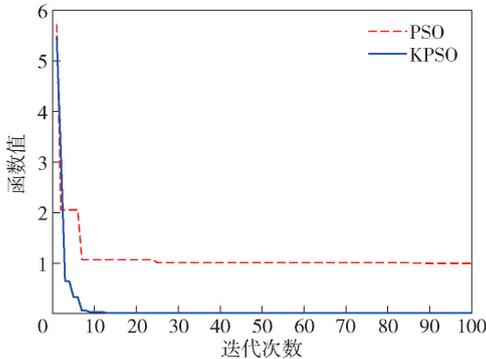


图 1 Rastrigin 函数求解迭代图

由图 1 可以看到经典 PSO 在 Iteration 为 7 时收敛, 但粒子在适应度值为 1 附近陷入局部最优解, KPSO 在 Iteration 为 9 时收敛到最优解附近, 具有更好的跳出局部最优解的能力。

图 2 和图 3 分别为改进前后的粒子群算法迭代到 Iteration 为 10 时粒子分布图。

可以看出, KPSO 粒子朝着最优值方向聚集, 粒子间具有较大的差异性, 而 PSO 中几个种群主要聚集在坐标轴的第 2 象限。陷入局部最优解中, 难以进行进一步搜索, 这也显示了 PSO 容易陷入局部最

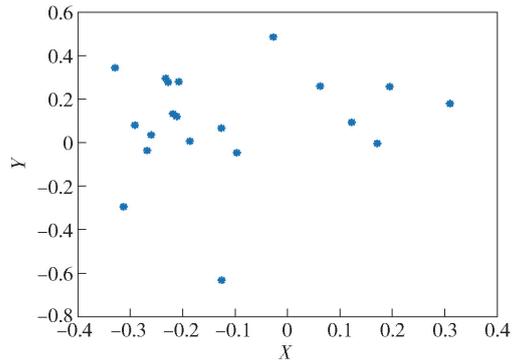


图 2 PSO 算法粒子聚集图

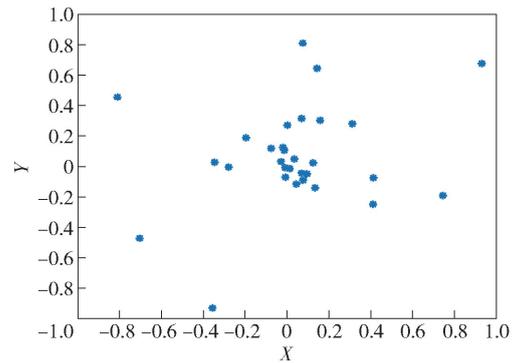


图 3 KPSO 算法粒子聚集图

优的缺陷。

3.2 物流配送路径优化模型求解

仿真采用 Solomon 标准数据集中 R101 的数据集作为实验数据, 笔者按比例随机选取 60% 用户有同时取货送货的需求, 对于取货需求将采取随机生成的方法添加。

其仿真参数为: 最大迭代次数 $T_{max} = 100$, 惯性权重采用经典的线性调整, 即最大惯性权重 $\omega_{max} = 0.9$, 最小惯性权重 $\omega_{min} = 0.4$, 初始惯性权重 $\omega = 0.9$, 粒子数 $n = 50$, 加速系数采用经典粒子群中的加速系数, $b_1 = 2, b_2 = 2$, 邻域加速系数 $b_3 = 1$, 聚类簇 $k = 3$, 假设 $c_1 = 0.3, c_2 = 0.1, c_3 = 0.3$, 单位油耗成本 $c_4 = 0.3$, 车辆满载时行驶单位距离的油耗 $r^* = 1$, 车辆空载时行驶单位距离的油耗 $r_0 = 0.5$, 车辆的运行速度 $v = 5 \text{ m/s}^2$ 。利用 Matlab 仿真求解, 迭代收敛图如图 4 所示。

由图 4 可以看出 PSO 在 Iteration 为 10 时陷入局部最优解, KPSO 在 Iteration 为 46 时跳出了局部最优解, 且能搜索到更优解, 对解决物流调度问题具有更大的优势。

由上面的仿真算例可以看出, 改进后的粒子群算法具有很好的跳出局部最优解的能力, 并能够较

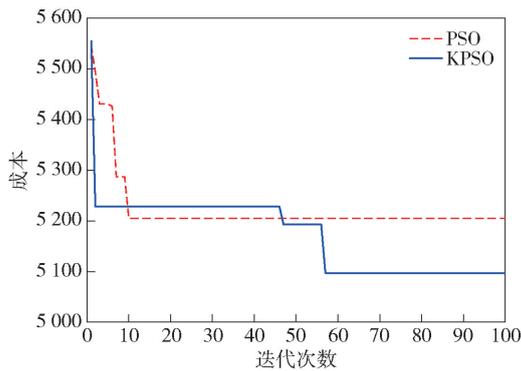


图 4 配送模型迭代收敛图

好地解决 PDVRPTF 的问题.

4 结束语

针对传统的车辆路径优化模型的缺点和经典粒子群算法容易陷入局部最优的不足,详细分析了油耗、时间窗、取送一体化等约束条件,并在此基础上构建相应的 PDVRPTF 数学模型.同时基于 k -medoids 对粒子群算法中的粒子动态聚类,采用 all-ring 型拓扑结构设计改进粒子群算法,并分别用经典粒子群算法和改进后粒子群算法求解 Rastrigin 函数和 Solomon 改进数据集的车辆调度问题.测试结果表明,改进后的粒子群算法能够很好地跳出局部

最优解,可以较好地优化多目标的车辆调度问题.

参考文献:

- [1] Shi Y H, Eberhart R. A modified particle swarm optimizer[C] // Proceedings of the IEEE Conference on Evolutionary Computation, Anchorage: IEEE, 1998: 69-73.
- [2] Xie X F, Zhang W J, Yang Z L. Adaptive particle swarm optimization on individual level[C] // International Conference on Signal Processing. Beijing: IEEE, 2002: 1215-1218.
- [3] 王俊伟,汪定伟. 粒子群算法中惯性权重的实验与分析[J]. 系统工程学报, 2005, 20(2): 194-198.
Wang Junwei, Wang Dingwei. Experiments and analysis on inertia weight in particle swarm optimization[J]. Journal of Systems Engineering, 2005, 20(2): 194-198.
- [4] 余志生. 汽车理论 [M]. 第 3 版. 北京: 机械工业出版社, 2000: 42-45.
- [5] 杨韬, 邵良杉. 采用改进的 k 均值聚类分析策略的粒子群算法[J]. 计算机工程与应用, 2009, 45(12): 52-54.
Yang Tao, Shao Liangshan. Hybrid particle swarm optimization employing improved k -means clustering analysis strategy[J]. Computer Engineering and Applications, 2009, 45(12): 52-54.