

文章编号:1006-2467(2021)10-1291-12

DOI: 10.16183/j.cnki.jsjtu.2019.176

基于级联的改进差分进化算法的 仓储多订单分批优化

陈广锋, 余立潮

(东华大学 机械学院, 上海 201620)

摘要: 为了提高仓储车间货物调度的柔性和响应效率,提出一种级联的改进差分进化算法,构建以拣货小车运行时间、货架稳定性及货位的存货能力为资源条件的货位分配和以每批订单中每一货物分配到对应分区的最优货位的最大完工时间为条件的订单重新分批分配的两级目标模型.将拉格朗日插值算法融进标准差分进化算法得到改进算法,分别求解两级目标模型,并将两级求解过程级联,完成级联关系的差分进化算法求解多订单分批分配问题.改进的差分进化算法在自适应地调整差分进化参数基础上,结合拉格朗日插值优化差分进化算法的局部搜索能力,运用局部和全局切换因子动态调整进化方向,提高算法的收敛性能.将改进的差分进化算法应用于求解多订单分批分配问题,实验结果证明,改进的算法优化结果明显优于粒子群优化算法、遗传算法及标准差分进化算法,减少了每一批订单的最大完工时间,有效地均衡工作负载.

关键词: 订单分批分配优化;货架稳定性;最大完工时间;拉格朗日插值;差分进化

中图分类号: TP 18

文献标志码: A

Multi-Order Batch Optimization of Warehouse Based on Cascaded Improved Differential Evolution Algorithm

CHEN Guangfeng, YU Lichao

(School of Mechanical Engineering, Donghua University, Shanghai 201620, China)

Abstract: In order to improve the flexibility and response efficiency of warehouse dispatching, a cascaded improved differential evolution algorithm is proposed to construct the allocation of goods with the picking trolley running time, shelf stability, and inventory capacity as resource conditions. The maximum completion time for each item in the batch order assigned to the optimal location of the corresponding partition is the two-level target model that is re-batch-allocated for the conditional order. The Lagrangian interpolation algorithm is integrated into the improved algorithm of the standard differential evolution algorithm to solve the two-level target model, and the two-level solution process is cascaded to complete the cascaded differential evolution algorithm to solve the multi-order batch allocation problem. Based on the adaptive adjustment of differential evolution parameters, the improved differential evolution algorithm combines the local search ability of Lagrangian interpolation to optimize the differential evolution algorithm, and uses local and global switching factors to dynamically adjust the evolution direction and improve the convergence performance of the algorithm. The improved differential evolution algorithm is

收稿日期:2019-06-23

基金项目:国家重点研发计划资助项目(2017YFB1304000)

作者简介:陈广锋(1976-),男,上海市人,副教授,主要研究方向为智能检测与控制.

通信作者:余立潮,男,硕士生;E-mail:15216879773@163.com.

applied to solve the problem of multi-order batch allocation. The experimental results show that the improved algorithm optimization results are better than the particle swarm optimization algorithm, the genetic algorithm, and the standard differential evolution algorithm, which reduces the maximum completion time of each batch of orders and effectively balance the workload.

Key words: order batch allocation optimization; shelf stability; maximum completion time; Lagrangian difference; differential evolution

仓储货物多订单的执行操作显著影响着供应链性能,亚马逊和沃尔玛等公司不断提高仓库拣货的柔性作业效率^[1-2].提高仓库拣货的柔性作业效率的关键步骤包括订单合理分配、批处理、排序及拣选最优小车路径.仓库订单拣选是一项劳动密集型活动,占用高达大约61%的仓库成本,而其中大约59%的拣货时间用于行程优化^[3].因此,智能优化的方法需要从多个方面优化订单履行,从而最小化订单周期时间.

已有文献研究各种情况下的订单拣选问题^[4].Çelk等^[5]运用精确方法,以总距离和时间最短为目标函数,结合鱼形过道的仓库布局解决订单拣选问题.Matusiak等^[6]以总距离与时间的比值最小为目标函数提出的模拟退火(SA)算法用于批处理.Chen等^[7]为订单批处理提出了非线性混合的整数规划模型(MIP).在模型的基础上,以最小化订单的总延迟时间为目标函数,提出了混合遗传(GA)和蚁群优化(ACO)算法.后来Chen等^[8]使用混合粒子群优化(PSO)和ACO方法解决订单批处理问题.Henn^[9]借助最大限度地减少总延迟时间模型研究可变邻域下降(VND)和变邻域搜索(VNS)算法在批量处理的应用.Öncan^[10]也引入MIP并在每个迭代中运用禁忌搜索作为局部搜索算法.Pan等^[11]引入订单的分批和工作均衡问题,运用分组GA研究订单的分批效果.Chen等^[12]制定出一个创新的订单批处理方法解决仓库中的拥堵问题,他们使用ACO获取每个小车的初始路线,然后使用一组方法规则试图修改路线以缓解拥堵.Cortés等^[13]将库存约束考虑在内,采用禁忌搜索(TS)算法来解决拣选小车路径寻优问题,他们将TS分别与GA和SA的结果进行了对比,发现TS比其他方法具有更好的效果.Henn等^[14]采用几种元启发式算法建立总延迟最小化数学模型解决批量处理问题,与标准的启发式算法相比,他们的方法可以改进解决方案.Menéndez等^[15]提出了基于VNS改进算法解决订单批处理问题,并通过广泛的测试显示了该方法的性能.Žulj等^[16]提出基于自适应邻域搜索算法和禁忌搜索算

法,建立最小化总的行程和时间数学模型解决订单分配问题.陆汉东等^[17]提出基于禁忌搜索算法分批调度柔性作业车间的模型.

上述研究从仓库的布局、使用的方法、考虑的特殊约束以及所使用的优化目标等方面考虑订单拣选相关问题,大多数研究都集中在建立行程的总距离或者时间最小化数学模型解决订单拣选问题,只有少数学者将完工时间作为优化目标融进订单分批问题.Scholz等^[18]的研究将订单分配、订单批处理及拣选小车路径寻优同时考虑在内,使完工时间尽量减少,达到总拖期较少的目标.然而,每批订单拣选的批次序列不会改变总体完工时间,不同的批次序列可能导致不同的拖期级别,并不能保证拣货小车之间的工作量均衡.

订单分配、订单批次、拣选小车路线及可用拣选小车数量会影响总拣货完工时间.虽然最小化行驶的总距离可以降低拣选成本,但并不能保证拣货小车之间的工作量均衡.

针对上述问题,考虑最小化完工时间有利于拣选小车之间的负载均衡,因此,考虑拣选仓库中的最小完工时间是必不可少的.近年来,众多研究人员对使用差分进化(DE)解决优化问题越来越感兴趣.由Storn等^[19]提出的DE算法是一种有效的局部搜索算法,各国学者将其用于各种优化问题.本文提出基于拉格朗日插值混合差分进化算法,建立每批订单最小化最大完工时间的数学模型,用拉格朗日插值提高DE算法的局部搜索能力,通过自适应控制DE算法的进化参数,使算法避免早熟收敛.此外,设计出DE算法局部和全局搜索自动切换因子,可以显著改善算法在收敛性和准确性方面的表现.

1 问题模型

仓储环境下的多订单分配优化内容可以描述如下:订单 $o \in \{1, 2, \dots, O\}$,将每个订单 o 含有客户要求数目不确定的货物 $n \in \{1, 2, \dots, N\}$ 与具有 i 排, j 列, k 层的仓储货物拣选库(图1)在 $r \in \{1, 2, \dots, R\}$ 的拣选小车的运输下进行交互.图1所示仓储货

物拣选库研究的仓库呈矩形,所有通道都是平行的,黑格子表示所需拣选的货物货位,白格子表示无需拣选货物货位.每一件货物可以选择任意一批入库或者出库订单,每一批入库或者出库的不同订单都可以随机组合,每一批入库或者出库订单可以选择任何一辆拣选小车,每一辆拣选小车可以选择任意

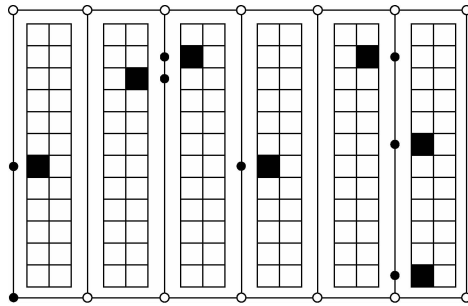


图1 仓储货物拣选库

Fig. 1 Warehouse goods picking library

一条到达目的位置的路径.

拣货操作从拣选小车挑选分配给它们的订单开始,每批包含1个或多个订单.图2所示为多个订单和多辆拣选小车的分配优化问题,其中每个格子代表完成某个订单若干个物料对应分区拣选小车所消耗的最大完工时间,图中, q 为不同的物料分区; t 为某一分区完成拣选的最大完工时间.假设每个订单对应的物料分区最优仓储货位已经求得,订单一起做批处理,每批都分配给第 r 辆拣选小车, t_{\max} 是所有批次中最大的完工时间.假设拣选小车的行进速度恒定,不受过道拥挤的影响.当前研究的问题可以表述如下:通过一定数量的拣选小车完成一组非空的客户订单的目标货位分配,批次 p 被分配到 m 个订单,最小化 p 中 m 个订单每个分区的最大完工时间,从而最小化所有批次各分区的最大完工时间,其中每个客户订单包含一组具有仓储中待确定位置的物料若干.

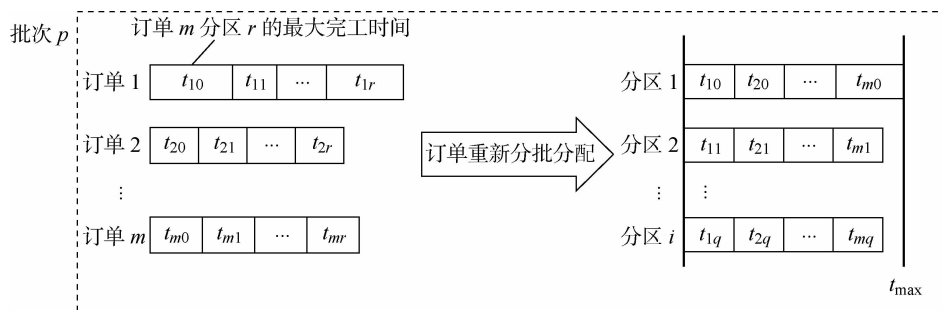


图2 多订单分批分配优化

Fig. 2 Allocation optimization of multi-order batch

为了最小化最大完工时间,建立对应数学模型如下:

$$\min t_{\max} \quad (1)$$

$$\text{s. t. } \sum_{b=1}^B \sum_{r=1}^R x_{bo}^r = 1, \quad \forall o \in \{1, 2, \dots, O\} \quad (2)$$

$$\sum_{z=1}^Z \sum_{b=1}^B \sum_{r=1}^R Y_{bn}^{rz} = 1, \quad \forall n \in \{1, 2, \dots, N\} \quad (3)$$

$$\sum_{o=1}^O x_{bo}^r \leq C_{\text{order}} \quad (4)$$

$$\forall b \in \{1, 2, \dots, B\}, \quad r \in \{1, 2, \dots, R\}$$

$$\sum_{o=1}^O \sum_{n=1}^N x_{bo}^r q_{on} \leq C_{\text{unit}} \quad (5)$$

$$\forall b \in \{1, 2, \dots, B\}, \quad r \in \{1, 2, \dots, R\}$$

$$\delta = \frac{\sum_{i \in \sum_{n \in p_i^z} Y_{bn}^{rz} q_{on}} \|M^2 - p_i^z\|_2}{\sum_{n \in p_i^z} Y_{bn}^{rz} q_{on}} \quad (6)$$

$$\forall b \in \{1, 2, \dots, B\}, \quad r \in \{1, 2, \dots, R\},$$

$$z \in \{1, 2, \dots, Z\}$$

$$S = \sum_{z=1}^Z \sqrt{\frac{(\|M^2 - p_i^z\|_2 - \delta)^2}{\sum_{n \in p_i^z} Y_{bn}^{rz} q_{on}}} \quad (7)$$

$$\forall b \in \{1, 2, \dots, B\}, \quad \forall r \in \{1, 2, \dots, R\}$$

$$C = \begin{cases} \sqrt{\frac{\sum_{o=1}^O \sum_{n=1}^N x_{bo}^r q_{on}}{C_{\text{unit}}}}, & \frac{\sum_{o=1}^O \sum_{n=1}^N x_{bo}^r q_{on}}{C_{\text{unit}}} \geq \eta \\ \zeta, & \text{否则} \end{cases} \quad (8)$$

$$\forall b \in \{1, 2, \dots, B\}, \quad r \in \{1, 2, \dots, R\}$$

$$t_{bn}^z = \max_{z \in Z} \frac{f \sqrt{\|p_i^z - p_{\text{start}}\|_2}}{v} \quad (9)$$

$$\forall b \in \{1, 2, \dots, B\}, \quad r \in \{1, 2, \dots, R\}$$

$$\omega_1 t_{bn}^z Y_{bn}^{rz} x_{bo}^r + \omega_2 (C + S) \leq M T_b^{rz} \quad (10)$$

$$\forall b \in \{1, 2, \dots, B\}, \quad r \in \{1, 2, \dots, R\}$$

$$\omega_1 + \omega_2 = 1 \quad (11)$$

$$S_r^z \geq \sum_{b=1}^B T_b^{rz}, \quad \forall r \in \{1, 2, \dots, R\} \quad (12)$$

$$t_{\max} \geq S_r^z, \quad \forall r \in \{1, 2, \dots, R\} \quad (13)$$

式中:

$$x_{bo}^r = \begin{cases} 1, & \text{订单 } o \text{ 被分配到拣选小车 } r \text{ 的第 } b \text{ 批次} \\ 0, & \text{否则} \end{cases}$$

$$Y_{bm}^z =$$

$$\begin{cases} 1, & \text{货物 } n \text{ 被拣选小车 } r \text{ 的第 } b \text{ 批次分配到分区 } z \\ 0, & \text{否则} \end{cases}$$

C_{order} 为批次 p 中拣选小车 r 所能承载的最大订单数量; $b \in \{1, 2, \dots, B\}$ 为订单批次索引号; q_{om} 为订单 o 中货物 n 的数量; C_{unit} 为仓储货物拣选库货位的承受能力; ζ 为物料集中度均值; M 为足够大的数; ζ 为其他阈值范围的常数补偿系数; t_{bm}^z 为分区 z 批次 b 的 m 个订单所用时间; p_i^z 为物料所在分区的位置索引; $z \in \{1, 2, \dots, Z\}$ 为仓储货物拣选库分区索引号; S 为物料在空间集中度的标准差; C 为补偿系数; η 为阈值常数; f 为货物辗转频率; p_{start} 为拣货起始地点; v 为拣选小车行驶速度; ω_1, ω_2 分别为权重系数; T_b^z 拣选小车 r 完成分区 z 批次 b 订单所用时间; S_r^z 为分区 z 每辆拣选小车 r 的完工时间; $p_i^z \in \{1, 2, \dots, P\}$ 为仓储货物的拣选库分区 z 货位 k .

如式(1)所示,模型是以最小化最大完工时间为目标函数,目标函数的解决是以式(10)的惩罚条件为前提;约束(2)和(3)保证在可行的解集中所有订单和所有物料都能被拣选小车选中;约束(4)表示批次 b 中拣选小车 r 所能承载的最大订单 o 的数量;约束(5)为仓储货物拣选库货位容量约束并确保批次 b 中拣选小车 r 承载合理的物料重量;约束(6)~(8)为货架稳定性和货位的存货能力;约束(9)为拣货小车 r 在满足物料周转率条件下将物料运送到目标货位的等价运行时间;式(10)表示保证在满足约束(6)~(9)条件下物料的最优货位分配最优模型;约束(11)为权重因子;约束(12)和(13)为每辆拣选小车的完工时间以及其最大完工时间。

2 融合拉格朗日插值算法的改进差分进化(LGDE)算法

差分进化算法主要包括4个步骤:种群初始化、变异、交叉及选择。

2.1 标准差分进化

在所求解空间内, $\{x_{\text{NP}}(0) \mid x_{u,d}^l \leq x_u(0) \leq x_{u,d}^U, u \in (1, 2, \dots, \text{NP}), d \in (1, 2, \dots, D)\}$ (NP为种群的大小, D 为基因的长度)随机均匀地产生初始种群,产生的种群为

$$x_{u,v}(0) = x_{u,v}^l + \text{rand}(0, 1)(x_{u,v}^U - x_{u,v}^l) \quad (14)$$

式中: $x_{u,v}(0)$ 为种群中第0代的第 u 条“染色体”以

及第 v 个基因; $\text{rand}(0, 1)$ 为均匀生成 $0 \sim 1$ 的随机数。

差分进化算法通过随机选取种群中的两个个体进行向量缩放与变异个体进行合成:

$$v_i(g+1) = x_{r_1}(g) + F[x_{r_2}(g) - x_{r_3}(g)] \quad (15)$$

$$i \neq r_1 \neq r_2 \neq r_3$$

式中: $v_i(g+1)$ 为变异个体; F 为缩放因子; $x_i(g)$ 为第 g 代的第 i 个个体; r_1, r_2 及 r_3 分别为第 r 个个体的随机3个基因位。

对 $x_i(g)$ 及其变异个体 $v_i(g+1)$ 进行交叉操作产生新的个体 $u_{i,j}(g+1)$:

$$u_{i,j}(g+1) = \begin{cases} v_{i,j}(g+1), & \text{rand}(0, 1) \leq \text{CR 或 } j \leq j_{\text{rand}} \\ x_{i,j}(g), & \text{否则} \end{cases} \quad (16)$$

式中: CR 为交叉概率; $j_{\text{rand}} \in (1, 2, \dots, D)$ 为随机整数。

采用贪婪算法选择进入下一代种群个体:

$$x_i(g+1) = \begin{cases} u_i(g+1), & f(u_i(g+1)) \leq f(x_i(g)) \\ x_i(g), & \text{否则} \end{cases} \quad (17)$$

式中: f 为目标评价函数。为了提高算法的收敛速度,保持较高的种群多样性,选择从种群中随机选择4个不同个体修正式(15):

$$v_i(g+1) = x_{\text{gbest}}(g) + F(x_{r_1}(g) - x_{r_2}(g) + x_{r_3}(g) - x_{r_4}(g)) \quad (18)$$

$$i \neq r_1 \neq r_2 \neq r_3 \neq r_4$$

式中: $x_{\text{gbest}}(g)$ 为最优个体染色体; r_4 为第 r 个个体的第4个随机基因位。

2.2 时变交叉概率

由式(16)可知,如果 CR 越大,则选中个体 $v_{i,j}(g+1)$ 的几率较大,有利于局部搜索和加速收敛速率;反之,选中 $x_i(g)$ 的几率大,有利于保持种群的多样性和全局搜索。良好的搜索策略应该是在搜索的初始阶段保持种群多样性,进行全局搜索,而在搜索的后期应加强局部搜索能力,以提高算法的精度。因此,采用下式将 CR 应用于自适应差分进化(ADE)算法和本文算法。

$$l = e^{1 - \frac{t_m}{t_m + 1 - T}} \quad (19)$$

$$\text{CR} = \text{CR}_0 2^l$$

$$\text{CR} = 1/[1 + e^{-(t-t_0)}] \quad (20)$$

式中: l 为概率系数; t_m 为最大进化代数; T 为当前进化代数; CR_0 为交叉概率算子; t' 为当前消耗时间; t_0 为消耗时间参数。

2.3 拉格朗日插值

拉格朗日插值是一种在已知多个坐标点的情况

下得到通过坐标点的近似函数的建模方法^[20]. 采用拉格朗日插值构造近似函数加强差分进化算法的局部搜索能力. 已知坐标点 $(x_{i,0}(g), f(x_{i,0}(g)))$, $(x_{i,1}(g), f(x_{i,1}(g)))$, \dots , $(x_{i,D}(g), f(x_{i,D}(g)))$, 其中 $x_{i,j}(g)$ 为第 g 代种群个体, $f(x_{i,j}(g))$ 为第 g 代种群个体的适应度. 因此得到 l 次拉格朗日插值多项式:

$$L_l(x_{i,j}(g)) = \sum_{l=0}^D \left[\prod_{\substack{m'=0 \\ m' \neq l}}^D \frac{x_{i,j}(g) - x_{i,m'}(g)}{x_{i,l}(g) - x_{i,m'}(g)} f(x_{i,l}(g)) \right] \quad (21)$$

式中: m' 为索引参数. 本文取 $l \in \{0, 1, 2\}$, 代入式(21)得到抛物插值, 将其简化成二次函数模型:

$$L_2(x_{i,j}(g)) = h'x_{i,j}^2(g) + p'x_{i,j}(g) + q' \quad (22)$$

式中: h' 、 p' 及 q' 为方程式的系数, 计算过程分别为 $h' =$

$$-\frac{l_1 l_1^1 f(x_{i,0}(g)) + l_2 l_2^2 f(x_{i,1}(g)) + l_0 l_0^0 f(x_{i,2}(g))}{l_0 l_1 l_2}$$

$$p' =$$

$$-\frac{l_1 f(x_{i,0}(g)) + l_2 f(x_{i,1}(g)) + l_0 f(x_{i,2}(g))}{l_0 l_1 l_2}$$

$$q' = q_0 + q_1$$

$$q_0 = -[l_2(l_1 + l_1^1)(l_2 + l_2^2)f(x_{i,0}(g)) + l_0(l_0 + l_0^0)(l_2 + l_2^2)f(x_{i,1}(g))]/2l_0 l_1 l_2$$

$$q_1 = -\frac{l_1(l_0 + l_0^0)(l_1 + l_1^1)f(x_{i,2}(g))}{2l_0 l_1 l_2}$$

$l_0, l_1, l_2, l_0^0, l_1^1$ 及 l_2^2 由下式计算:

$$l_0 = x_{i,0}(g) - x_{i,1}(g)$$

$$l_1 = x_{i,1}(g) - x_{i,2}(g)$$

$$l_2 = x_{i,2}(g) - x_{i,0}(g)$$

$$l_0^0 = x_{i,0}(g) + x_{i,1}(g)$$

$$l_1^1 = x_{i,1}(g) + x_{i,2}(g)$$

$$l_2^2 = x_{i,2}(g) + x_{i,0}(g)$$

为了利用拉格朗日插值优化差分进化算法搜索局部最优值, 取出最优个体染色体的一个基因位和变异最优个体染色体的一个基因位作为插值点, 得到两个插值点: $x_{i,0}(g)$ 和 $x_{i,1}(g)$, 则 $x_{i,2}(g)$ 可由下式得到:

$$x_{i,2}(g) = \begin{cases} -\frac{p}{2h} & h > 0 \\ \frac{l_0^0}{2} + \text{rand}(0,1)l_1, & h = 0 \text{ 或 } p = 0 \\ \frac{l_0^0}{2} + l, & \text{否则} \end{cases} \quad (23)$$

$$l = \text{rand}(0,1) \left(\frac{l_0^0}{2} - \min \left\{ \frac{l_0^0 + l_0}{2}, \frac{l_1^1 + l_1}{2} \right\} \right) +$$

$$\text{rand}(0,1) \left(\frac{l_0^0}{2} - \max \left\{ \frac{l_0^0 + l_0}{2}, \frac{l_1^1 + l_1}{2} \right\} \right) \quad (24)$$

算法1 拉格朗日插值算法流程

步骤1 输入当前迭代阶段的 $x_{\text{gbest}}(g)$ 和该最优染色体对应的最优目标函数值 $f(x_{\text{gbest}}(g))$, 遍历最优染色体每一位基因位 h , 通过随机数随机生成 $r_1 \in D, r_2 \in D$, 替代 $x_{\text{gbest}}(g)$ 的第 h 位基因, 得到新的染色体 $x_{r_1}(g), x_{r_2}(g)$;

步骤2 计算新的染色体 $x_{r_1}(g), x_{r_2}(g)$ 对应的函数值 $f(x_{r_1}(g)), f(x_{r_2}(g))$;

步骤3 将步骤1和2计算得到的染色体 $x_{\text{gbest}}(g), x_{r_1}(g)$ 及 $x_{r_2}(g)$ 和对应的函数值 $f(x_{\text{gbest}}(g)), f(x_{r_1}(g))$ 及 $f(x_{r_2}(g))$ 代入式(22)及 p', q', h' 的计算式进行计算, 根据式(23)、(24)将得到染色体新的基因位 $x_{i,2}(g)$, 对染色体新的基因位 $x_{i,2}(g)$ 进行界限判断, 根据实际染色体的编码设计求得相应的最优染色体;

步骤4 判断染色体基因位遍历是否结束, 如果没有结束, 转步骤1, 否则, 结束算法.

2.4 自适应调整差分进化算法搜索方向

为了保持种群的多样性与算法局部搜索能力的平衡, 采用全局收敛算子 λ_g 和局部收敛算子 λ_L 评估算法的进化方向:

$$\lambda_g = \frac{f(x_{\text{gbest}}(g)) - f(x_{\text{gbest}}(g+1))}{f(x_{\text{gbest}}(g))} \quad (25)$$

$$\lambda_L = \frac{f(x_{\text{Lbest}}(g)) - f(x_{\text{Lbest}}(g+1))}{f(x_{\text{Lbest}}(g))} \quad (26)$$

提出的自适应调整差分进化算法搜索方向是通过一个动态调整的阈值因子(DF)控制:

$$\text{DF} =$$

$$\begin{cases} \min \left\{ \frac{1}{1 + e^{-(t-t_0)}}, \text{DF}_{\max} \right\}, & \lambda_g f(x_{\text{gbest}}(g+1)) > 0 \\ \max \{ \text{DF}_{\min} + (\text{DF}_{\max} - \text{DF}_{\min})e^{-2\frac{t}{T}}, \text{DF}_{\min} \}, & \text{否则} \end{cases} \quad (27)$$

$$\text{DF} =$$

$$\max \{ \text{DF}_{\min} + (\text{DF}_{\max} - \text{DF}_{\min})e^{-2\frac{t}{T}}, \text{DF}_{\min} \} \quad (28)$$

$$\lambda_g > \lambda_L$$

式中: $\text{DF}_{\max}, \text{DF}_{\min} \in (0, 1)$. 为避免所提算法在进化过程全局无法收敛, 采用式(27)控制算法进入局部搜索区域, 式(28)则避免算法进入早熟阶段. 当 λ_g 较大时, 算法进入全局搜索, 反之进入局部搜索.

2.5 算法流程

2.5.1 LGDE 算法 LGDE 算法流程如图3所示, 具体步骤如下.

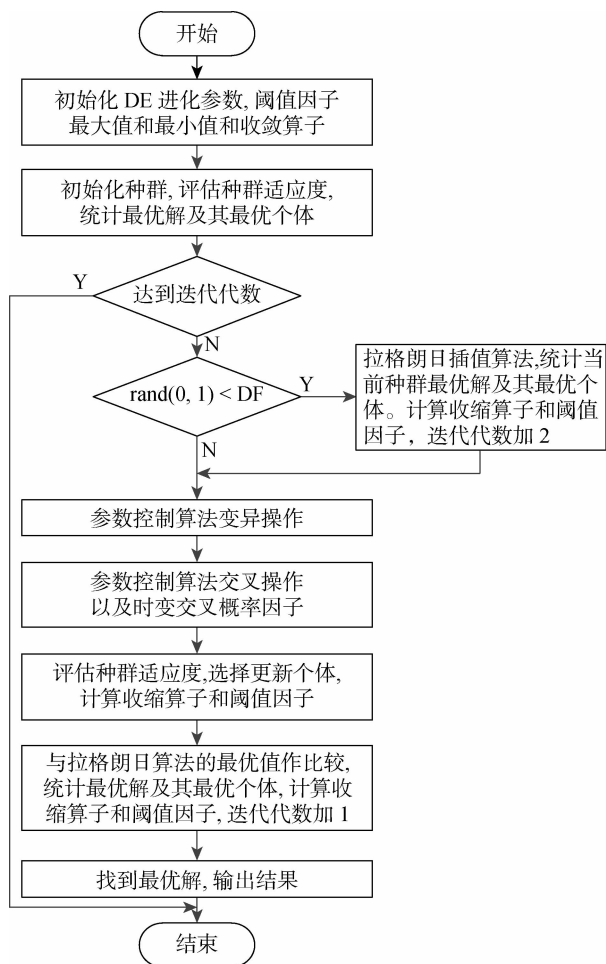


图 3 LGDE 算法流程图

Fig. 3 Flow chart of LGDE algorithm

步骤 1 设置算法的种群大小 NP , 最大进化代数为 $\max G$ 以及其他进化参数. 初始化 DF 和收缩算子. 初始化种群, 将要优化的货位或者时间存入数组, 并初步统计种群的最优个体 $x_{gbest}(g)$ 以及最优解 $f(x_{gbest}(g))$;

步骤 2 进入算法主循环, 判断结果是否达到 $\max G$, 如果是则退出主循环, 否则进入步骤 3;

步骤 3 生成一个 $0 \sim 1$ 之间的随机数 $\text{rand}(0, 1)$ 与 DF 初值比较, 如果 $\text{rand}(0, 1) < DF$ 则进入步骤 4, 否则进入步骤 5;

步骤 4 根据算法 1 计算出的 $f(x_{i,2}(g))$ 和统计出目标函数 $f(x_{i,0}(g))$ 、 $f(x_{i,1}(g))$ 及 $f(x_{i,2}(g))$ 的最优解及其最优个体, 按照式 (26)、(27) 更新 DF

和 λ_L , 并将迭代代数加 2;

步骤 5 根据式 (16)、(18) 及 (19) 进行变异和时变交叉操作, 按照式 (17) 进行选择操作;

步骤 6 统计当前代数种群的最优值与最优个体, 并与步骤 4 所得结果进行比较, 更新全局最优解和最优个体. 按照式 (25)、(28) 更新 DF 和 λ_g , 迭代代数加 1, 转至步骤 2;

步骤 7 得到最优值和最优个体.

2.5.2 多订单分批分配优化 多订单分批分配优化过程如图 4 所示, 具体步骤如下.

步骤 1 货位分配阶段采用图 5 所示单批次个体编码, 每一个基因位是仓储区域可行的仓储货位编号, 初始编码将采用 $0 \sim 1$ 的随机数编码, 表示货位的选中与否. 订单分配阶段采用如图 6 所示的多批次个体编码, 根据编码规则选择相应的订单编号为同一批次. 运用 LGDE 算法计算出每个分区相应订单货物对应的货位时间向量表, 进入步骤 2;

步骤 2 将基于订单编号的染色体输入到 LGDE 算法模型中, 算法模型将进行每批次每个分区的最大完工时间最优分配, 求解出每个订单的批次分配情况;

步骤 3 判断是否满足终止条件, 如果满足则完成订单分批分配, 结束分配, 否则转到步骤 2.

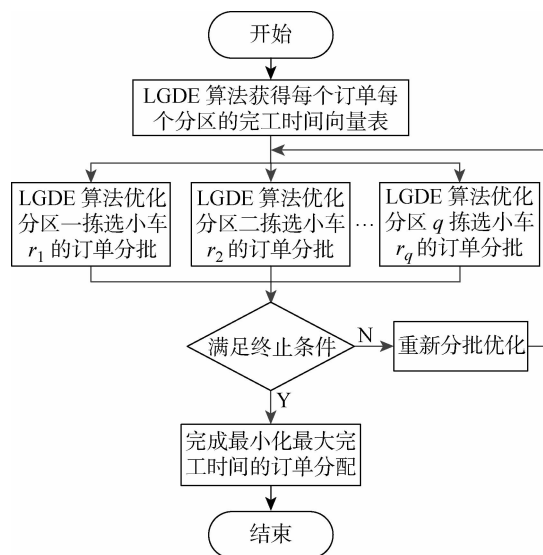


图 4 多订单分批分配优化

Fig. 4 Allocation optimization of multi-order batch

货位编号	14	21	10	3	30	21	25	31	40	24	22	31	42
初始编号	1	0	0	1	1	1	0	0	1	0	1	0	1
拓扑编码	14	无	无	3	30	21	无	无	40	无	22	无	42

图 5 单批次个体编码

Fig. 5 Single batch individual code

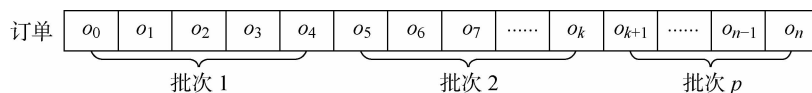


图 6 多批次个体编码

Fig. 6 Multi-batch individual code

3 算法实例验证

为了验证所提算法的有效性,本文将采用具有图 1 所示的货物拣选库结构进行实验.将货架从底层依次向上编号,所提算法的种群个体编码将采用 0~1 的随机数编码,同时为每种货物划定可行的存储货位.用传统的启发式算法如遗传算法、粒子群优

化算法进行对比实验,同时与标准的差分进化算法以及改进的自适应差分进化算法进行对比分析.实验计算机基本配置参数为: Intel(R)Pentium(R)/CPUG2020@2.90 GHz/6 GB,操作系统为 Windows 7,实验算法采用 Python 2.7 版本编程语言进行编写,所提算法参数设置如表 1 所示,表中 CR_{init} 为交叉概率的初始值.

表 1 改进差分进化算法参数设置

Tab. 1 Parameter settings of improved differential evolution algorithm

算法参数	种群规模	迭代代数	缩放因子	交叉概率	DF_{max}	DF_{min}	CR_{init}
货位分配	30	500	0.5	式(19)	0.20	0.05	0.18
订单分配	35	200	0.5	式(19),(20)	0.20	0.05	0.20

在差分进化算法参数中,交叉概率和缩放因子两个参数对算法的性能影响较大,经过多次仿真实验,分析不同的参数组合对算法性能的影响,改进的算法采用式(19)和(20)的变交叉概率因子有利于提高算法的性能.经过大量实验,将缩放因子取值在 0.5 左右取得较好实验效果,局部和全局调整因子在 0.04~0.3 时,算法具有较好的收敛性能.对于对比实验的算法中,本文将采用多次实验确定算法的参数.遗传算法中的变异因子和交叉因子会对算法产生重要影响,在粒子群算法的惯性权重表明粒子对当前的粒子速度的继承情况,其对算法的性能和效率会产生影响,自适应差分进化算法中的交叉概率因子将采用式(19)进行时变操作.参考文献[11]设定的遗传算法参数,经过多次实验将遗传算法的交叉因子设为 $p_c = 0.6$,变异因子设为 $p_m = 0.02$,参考文献[8]粒子群优化算法的求解过程并将惯性权重设置为 $\omega = 0.5$,学习因子设置为 $c_1 = c_2 = 2$.

3.1 货位分配实验分析

最小化每批订单的拣选小车的最大完工时间可以分解为最优各个分区的货位分配时间和最小化每批订单各分区的拣选小车的最大完工时间两个子问题,解决因多订单路径不同导致的订单完成顺序不同的问题.

为验证最优各个分区的货位分配时间子问题的解决效果,采用一定数量的货物样例进行实验,货物种类编号和数量如表 2 所示,实验结果如图 7 所示.图中 J 为目标函数值,算法测试的种群个体为 1 行,

63 列的数组矩阵.

表 3 是在拣货小车运行时间较短、货架稳定性较优及货位满足最大存货的能力等约束条件下各种算法目标函数值的计算结果.表中, N_t 为迭代代数; t_{CPU} 为 CPU 消耗时间.

表 2 某一订单批次货物种类编号和数量

Tab. 2 Number and quantity of the goods of an order batch

货物种类编号	数量
1	35
15	34
18	33
20	41
21	39
24	34
25	47
26	45
30	30

表 3 各个算法优化结果对比

Tab. 3 Comparison of optimization results of each algorithm

算法	最优值	N_t	t_{CPU}/s
GA	8.66	93	95.72
PSO	6.70	15	113.59
ADE	6.58	300	118.74
DE	4.58	350	110.68
LGDE	4.47	118	68.35

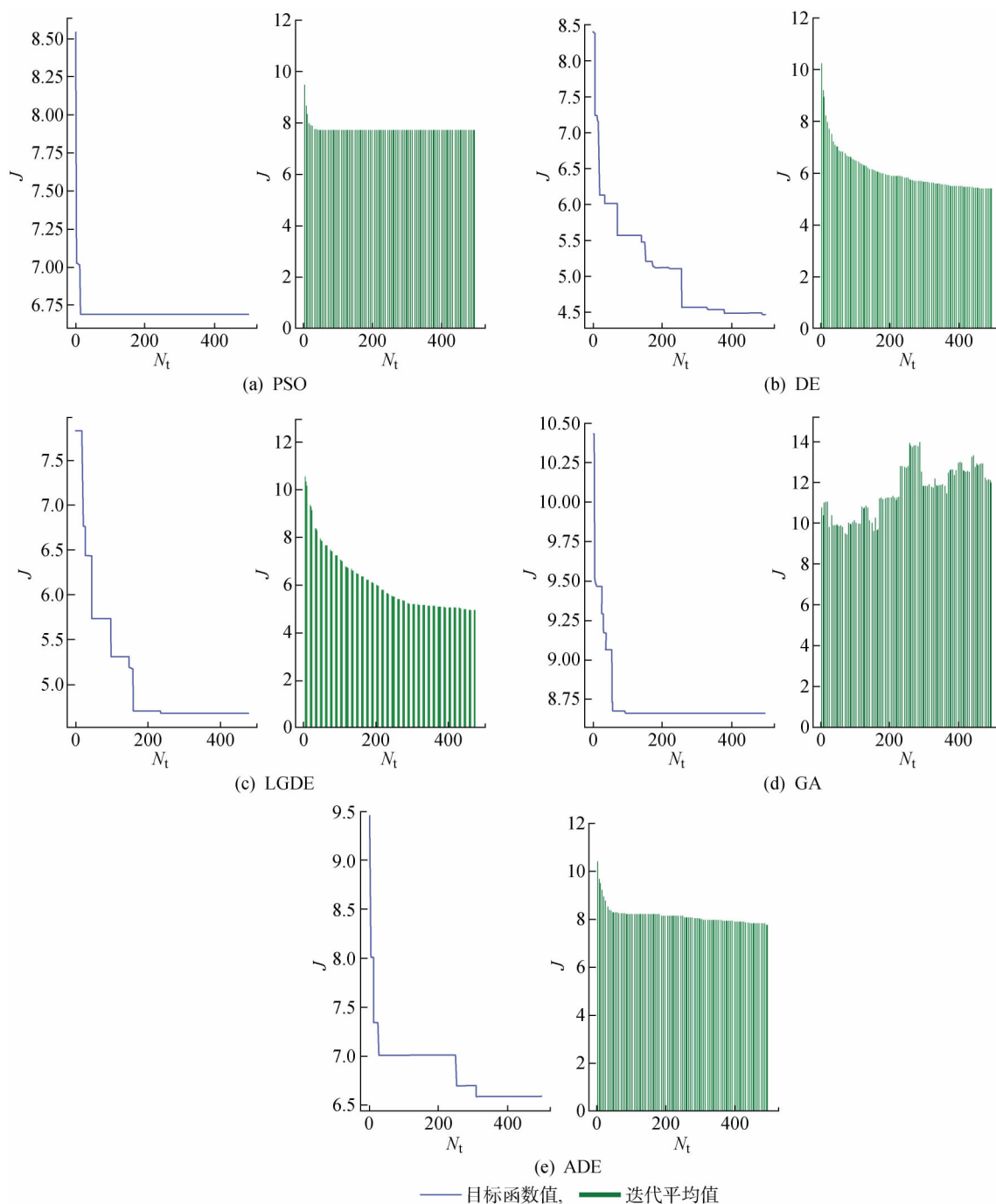


图7 各个算法目标函数值和迭代平均值迭代情况

Fig. 7 Objective function value of each algorithm and the iterative case of each generation

从图7和表3的结果可知,迭代过程每一代种群个体的平均值都有比较大的数值震荡,在5种算法中,遗传算法耗时最长,且其最优值和迭代代数较小,CPU消耗时间较长.标准的粒子群优化算法出现局部陷入最优的情况,解决问题的全局搜索的能力明显不如其他4种算法,最优值仅比遗传算法好,但CPU消耗时间却比ADE算法和DE算法短,逼近所提算法的时间耗时间.差分进化算法的在局部搜索方面具有较强的表现能力,在最优值上均能达

到较优的结果.在自适应差分进化算法中,扩大了种群的搜索能力,最优值上不如DE算法.对比以上5种算法,所提算法在各种表现上均出现较优结果,CPU消耗时间明显低于DE算法,局部搜索能力表现最佳.

表4和5为50次迭代目标函数平均值 \bar{J} 和CPU时间的平均值 \bar{t}_{CPU} ,图8所示为50次迭代目标函数值和CPU消耗时间变化情况.由表4、5及图8可以看出,本文所提算法的目标函数值和CPU消

表 4 各个算法运行 50 次目标函数值平均值

Tab. 4 Average value of objective function values of each algorithm running 50 times

算法	\bar{J}
LGDE	4.02
PSO	6.43
DE	4.03
ADE	5.23

耗时间的平均值在总体上具有较优效果,收敛的最优目标函数值相较于 PSO 算法降低了 37.34%,同时 CPU 消耗时间节省了 31.62%。由于自适应差分进化算法增加了种群的扰动度,导致局部搜索能力

下降。虽然改进的差分进化算法相对于标准差分进化算法最优目标函数值只降低了 0.017%,但是 CPU 消耗时间节省了 32.12%,所以,改进的算法在加速算法的收敛性上具有显著的效果。

表 5 各个算法运行 50 次 CPU 消耗时间的平均值

Tab. 5 Average value of CPU consumption time of each algorithm running 50 times

算法	$\bar{t}_{\text{CPU}}/\text{s}$
HLIDE	89.42
PSO	130.76
DE	131.71
ADE	138.15

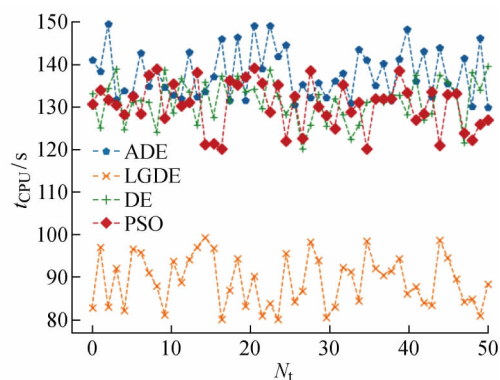
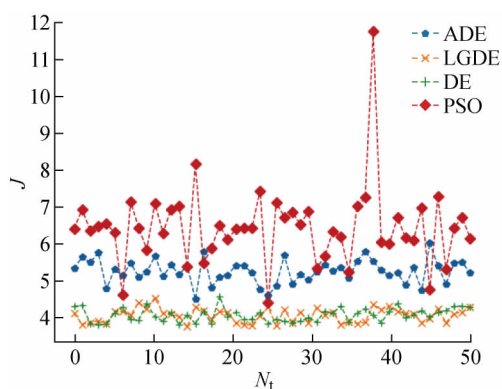


图 8 各个算法运行 50 次目标函数值和 CPU 消耗时间

Fig. 8 Objective function value and CPU consumption time of each algorithm running 50 times

3.2 多批次多订单分配实验分析

为验证多批次多订单分配的实验效果,采用 LGDE 算法对模型进行求解,模型以每一批次各个分区的最大完工时间为目标,最后保证所有批次各个分区工作负载均衡。分别采用遗传算法、粒子群优化算法、标准差分进化算法、自适应差分进化算法对模型进行对比实验分析,实验中采用数量为 100, 150, 200 及 250 的 4 组订单,每组订单分别进行多批分批实验,5 种算法求得的实验结果如图 9 和 10 所示。图中: C' 为订单数量; N_r 为小车数量。

图 9(a)的 f_{best} 为目标函数最优值,图 9(b)的 \bar{T} 表示每一批次时间平均值,图 9(c)的 t_{CPU} 表示 CPU 时间,图 9(d)的 T_{max} 表示最大完工时间,图 9(e)的 T_{min} 表示最小完工时间,图 9(f)的 $f_{\text{best, LGDE}}$ 的表示 LGDE 算法目标函数最优值。分析实验结果得知, LGDE 算法的目标函数值较其他优化算法取得较好的实验结果。LGDE 算法能够在合理的 CPU 时间内找到较优的目标函数值,显示出算法良好的性能, PSO 在 CPU 消耗时间和最小完工时间比其它算法

较优,但是在最大完工时间上效果较差,导致工作负载不均衡。如图 9(d)、9(f)所示,5 种算法中 LGDE 算法能够有效地实现每一批次最大完工时间最小化,并能保证目标函数值在可预测的趋势中不发生较大的偏差,说明 LGDE 算法的有效性。

为了更加清晰地分析 LGDE 算法对多批次多订单分配情况,将 LGDE 算法的实验结果以三维曲面的形式表示,三维曲面的形式可以更加直观的看到不同订单不同拣选小车数量对实验模型的影响,如图 10 所示。

图 10(a)表示 LGDE 算法的 CPU 损耗时间 ($T_{\text{CPU, LGDE}}$) 与拣选小车数量和订单拣选数量的关系。图 10(b)表示 LGDE 算法的最大完工时间与拣选小车数量和订单拣选数量的关系。由图 10 可知, LGDE 算法得到的实验结果具有可预测的效果,随着拣选小车数量和订单拣选数量的增加, CPU 损耗时间和最大完工时间的趋势是在增加的,即使在某些情况下,略微有实验偏差,实验结果依然可以清楚地显示出 LGDE 算法的可靠性。

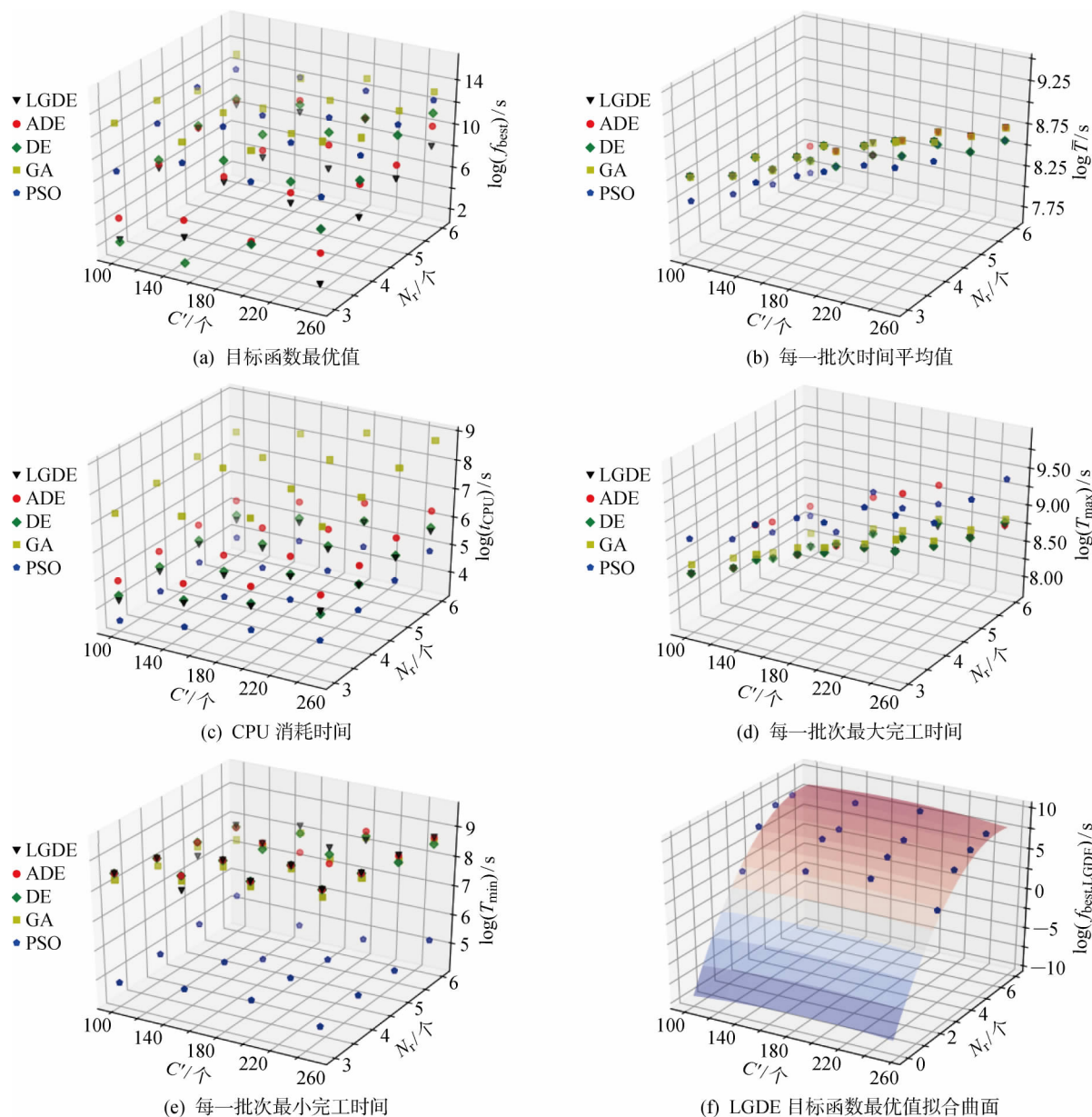


图9 各种算法实验结果

Fig. 9 Experimental results of each algorithm

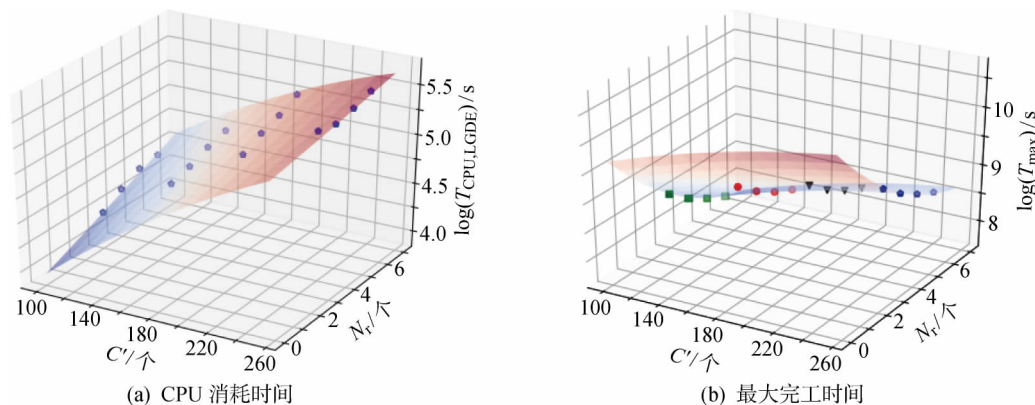


图10 LGDE 算法实验结果拟合曲面

Fig. 10 Fitting curves of experimental results of LGDE algorithm

图 11 所示为在不同订单不同拣选小车对应的每批次最大和最小完工时间的差值的三维和二维图,图中 \bar{f} 为工作均衡目标值.最小化最大完工时间有利于拣选小车的工作量均衡.可以看出,LGDE 算法总体上在 5 种算法中具有较好地工作负载均衡

效果,在拣选小车数量较少的情况下 DE 算法比 LGDE 算法表现较优,但随着拣选小车数量的增加,LGDE 算法的工作均衡能力就明显优于其它算法,因此,在经济成本合理的情况下,可用该模型预测拣选小车的数量.

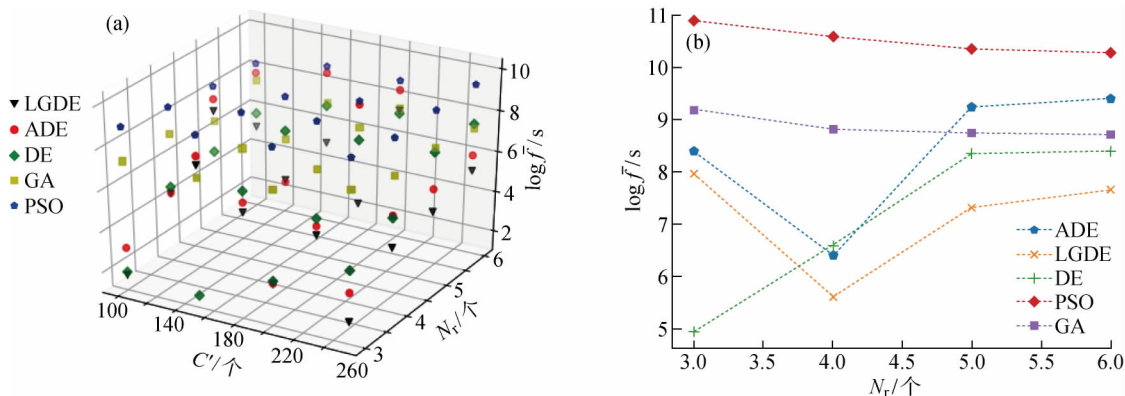


图 11 不同订单不同拣选小车对应的每批次最大和最小完工时间的差值

Fig. 11 Difference between maximum and minimum completion time for each batch of different picking carts for different orders

4 结语

提出一种融合拉格朗日插值算法的改进差分进化算法求解仓储多订单分批优化,构建了以拣货小车运行时间、货架稳定性及货位的存货能力为资源条件的货位分配和以每批订单中每一货物分配到对应分区的最优货位的最大完工时间为条件的订单重新分批分配的两级目标模型.通过实验验证了改进的差分进化算法比传统的启发式算法在解决订单分批问题上具有较佳的效果,相对于传统的启发式算法,可以得到更可靠的最大完工时间,有利于拣选小车更好地均衡工作量,为企业分配订单提供有效的参考模型.同时,改进的差分进化算法的局部寻优表现出更佳的全局搜索和局部搜索能力,避免算法早熟和无法收敛.在实际数据量较大的情况下,改进的差分进化算法求解出的仓储分批优化消耗计算机的运行时间也在合理的范围内.为了更好地改善算法的性能,分布式计算结构的设计以及加速算法是下一步的研究方向.

参考文献:

- [1] RANDALL W S, GIBSON B J, CLIFFORD DEFEE C, *et al.* Retail supply chain management: Key priorities and practices[J]. **The International Journal of Logistics Management**, 2011, 22(3): 390-402.
- [2] ARDJMAND E, SANEI BAJGIRAN O, RAHMAN S, *et al.* A multi-objective model for order cartoniza-

tion and fulfillment center assignment in the e-tail/retail industry[J]. **Transportation Research Part E: Logistics and Transportation Review**, 2018, 115: 16-34.

- [3] ARDJMAND E, SHAKERI H, SINGH M, *et al.* Minimizing order picking makespan with multiple pickers in a wave picking warehouse[J]. **International Journal of Production Economics**, 2018, 206: 169-183.
- [4] CERGIPOZAN Ç, TASAN A S. Order batching operations: An overview of classification, solution techniques, and future research[J]. **Journal of Intelligent Manufacturing**, 2019, 30(1): 335-349.
- [5] ÇELİK M, SÜRAL H. Order picking under random and turnover-based storage policies in fishbone aisle warehouses[J]. **IIE Transactions**, 2014, 46(3): 283-300.
- [6] MATUSIAK M, DE KOSTER R, KROON L, *et al.* A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse[J]. **European Journal of Operational Research**, 2014, 236(3): 968-977.
- [7] CHEN T L, CHENG C Y, CHEN Y Y, *et al.* An efficient hybrid algorithm for integrated order batching, sequencing and routing problem[J]. **International Journal of Production Economics**, 2015, 159: 158-167.
- [8] CHEN C Y, CHEN Y Y, CHEN T L, *et al.* Using a hybrid approach based on the particle swarm optimization and ant colony optimization to solve a joint

- order batching and picker routing problem[J]. **International Journal of Production Economics**, 2015, 170: 805-814.
- [9] HENN S. Order batching and sequencing for the minimization of the total tardiness in picker-to-part warehouses[J]. **Flexible Services and Manufacturing Journal**, 2015, 27(1): 86-114.
- [10] ÖNCAN T. MILP formulations and an iterated local search algorithm with tabu thresholding for the order batching problem[J]. **European Journal of Operational Research**, 2015, 243(1): 142-155.
- [11] PAN J C H, SHIH P H, WU M H. Order batching in a pick-and-pass warehousing system with group genetic algorithm[J]. **Omega**, 2015, 57: 238-248.
- [12] CHEN F Y, WANG H W, XIE Y, *et al.* An ACO-based online routing method for multiple order pickers with congestion consideration in warehouse[J]. **Journal of Intelligent Manufacturing**, 2016, 27(2): 389-408.
- [13] CORTÉS P, GÓMEZ-MONTOYA R A, MUÑUZURI J, *et al.* A tabu search approach to solving the picking routing problem for large- and medium-size distribution centres considering the availability of inventory and K heterogeneous material handling equipment[J]. **Applied Soft Computing**, 2017, 53: 61-73.
- [14] HENN S, SCHMID V. Metaheuristics for order batching and sequencing in manual order picking systems[J]. **Computers & Industrial Engineering**, 2013, 66(2): 338-351.
- [15] MENÉNDEZ B, PARDO E G, ALONSO-AYUSO A, *et al.* Variable neighborhood search strategies for the order batching problem[J]. **Computers & Operations Research**, 2017, 78: 500-512.
- [16] ŽULJ I, KRAMER S, SCHNEIDER M. A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem[J]. **European Journal of Operational Research**, 2018, 264(2): 653-664.
- [17] 陆汉东, 何卫平, 周旭, 等. 基于禁忌搜索的柔性作业车间分批调度[J]. **上海交通大学学报**, 2012, 46(12): 2003-2008.
- LU Handong, HE Weiping, ZHOU Xu, *et al.* An integrated tabu search algorithm for the lot streaming problem in flexible job shops[J]. **Journal of Shanghai Jiao Tong University**, 2012, 46(12): 2003-2008.
- [18] SCHOLZ A, SCHUBERT D, WÄSCHER G. Order picking with multiple pickers and due dates-simultaneous solution of order batching, batch assignment and sequencing, and picker routing problems[J]. **European Journal of Operational Research**, 2017, 263(2): 461-478.
- [19] STORN R, PRICE K. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces [J]. **Journal of Global Optimization**, 1997, 11(4): 341-359.
- [20] 黄裘俊, 张凯, 宋锦春, 等. 基于改进粒子群优化算法的取件机械手轨迹综合优化设计[J]. **东北大学学报(自然科学版)**, 2018, 39(11): 1636-1641.
- HUANG QiuJun, ZHANG Kai, SONG Jinchun, *et al.* Path synthesis optimal design of pick-up manipulator based on modified particle swarm optimization [J]. **Journal of Northeastern University (Natural Science)**, 2018, 39(11): 1636-1641.

(本文编辑:陈晓燕)