文章编号:1674-2974(2019)04-0010-07 DOI:10.16339/j.cnki.hdxbzkb.2019.04.002

基于轨迹映射模型的天车多目标调度方法

周炳海†,廖秀梅
(同济大学 机械与能源工程学院,上海 201804)

摘 要:为有效解决生产过程中的多天车调度问题,在统筹考虑天车任务初始态和时空约束等特征的基础上,提出一种全新的天车轨迹映射模型.结合传统差分进化方法,将库位分配规则和天车分配算法融合到调度算法的每一次迭代过程中以指导算法寻优.以最小化入库订单延迟成本和最小化出库订单等待成本作为评价指标,设计仿真试验并与经典多目标优化算法进行对比,验证了算法是有效可行的,进一步的数值试验表明了合理的调度规则可以有效提高天车调度性能.

关键词:天车轨迹映射;时空约束;多目标;启发式算法
中图分类号:TP391 **文献标志码**:A

Multi-objective Scheduling Method for Workshop Cranes Based on Projection Model of Trajectories

ZHOU Binghai†,LIAO Xiumei
(School of Mechanical and Energy Engineering,Tongji University,Shanghai 201804,China)

Abstract:To efficiently solve the multi-crane scheduling problem during the production process,a novel projection model of trajectories the proposed with the consideration of initial states of tasks and space-time constraints. Based on the differential evolution algorithm,stock allocation rules and a crane allocation algorithm were combined to guide the optimization process in each iteration. Taking the minimization of delay cost and waiting cost as the evaluation indices,the simulation experiment was designed and compared with the classical multi-objective optimization algorithms. The results show that the algorithm is effective and feasible. Further numerical experiments indicate that reasonable scheduling rules can effectively improve the crane scheduling performance.

Key words:projection of crane trajectories;space-time constraints;multi-objectives;heuristic algorithms

随着制造业物流系统的日趋完善,车间天车调度问题(crane scheduling in workshop,CSW)引起了广泛的关注.CSW属于典型的NP难问题,具有多机多任务特点,如何科学有效地提升天车调度水平已成为生产过程中亟待解决的问题[1-2].

目前,已有诸多学者对CSW问题进行了研究.

* 收稿日期:2018-04-11
基金项目:国家自然科学基金资助项目(71471135),National Natural Science Foundation of China(71471135)
作者简介:周炳海(1965—),男,浙江浦江人,同济大学教授
† 通讯联系人,E-mail:bhzhou@tongji.edu.cn

针对天车调度过程中的路径冲突问题,文献[3]考虑将车间分为多个平行区域,每台天车只负责其所在区域内的搬运任务.文献[4]分析了调度过程中天车的不可行路径,通过建立混合整数规划模型解决路径冲突.文献[5]创新性地提出了天车轨迹映射模型以减少决策变量,并通过移除决策树中次优状态的方法完成调度.文献[6]从排队论角度分析了车间天车调度问题,为提高天车利用率、减少浪费提供了新思路.

在天车调度过程中,考虑入库和出库订单的联合调度可以有效提升天车利用率.文献[7]针对卷钢存储的特殊性,对卷钢出库过程的天车调度展开研究.文献[8]在卷钢出库问题的基础上,考虑了卷钢入库的联合调度,分别设计了时空网络流方法和近似动态规划算法求解中小规模问题和大规模问题,但针对的是单天车调度. 鲜有文献考虑入库和出库订单联合调度背景下的 CSW 决策问题.

本文在考虑天车任务初始态和时空约束的基础上,对联合优化入库、出库订单的车间多天车调度问题进行研究,提出天车多目标调度方法,并以入库订单延迟成本和出库订单等待成本为指标,评价调度性能.

1　问题描述

1.1　天车调度问题

天车作为连接上、下游工序之间的桥梁,其车间内布局如图 1 所示. 当上游工序 M 加工完成后,天车将半成品从卸料区送入库位进行存储,当下游工序 N 准备加工时,天车将半成品从库位搬运至上料区等待加工.

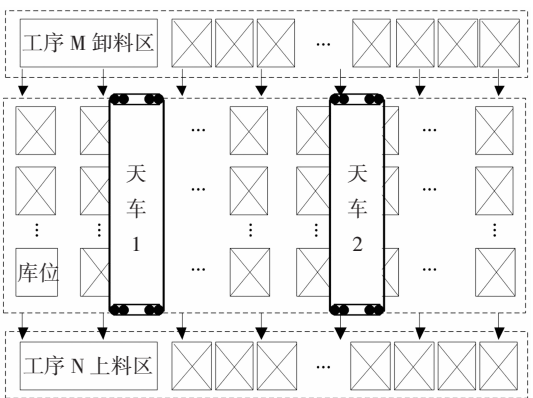

图 1　车间布局图

Fig.1　The workshop layout

为有效描述天车调度问题, 做如下基本假设:1)订单 i 包含两个任务,天车将 i 从起始库位(x_{i_1}, y_{i_1})起吊的操作为任务 i_1,将其运送到目标库位(x_{i_2}, y_{i_2})的操作为任务 i_2;2)天车集合为 C,其编号从左至右依次为:1,2,…,K;3)沿作业跨方向,相邻两天车的安全距离为 Δd;4)天车满载时沿 x 轴和 y 轴的速度分别为 $v_{x,f}$ 和 $v_{y,f}$,空载时为 $v_{x,e}$ 和 $v_{y,e}$;5)天车移动满足切比雪夫距离,若大车沿作业跨方向以速度 v_x 前后运动,小车以速度 v_y 左右运动,则天车运输 i 的时间为 $t=\max\{|x_{i_2}-x_{i_1}|/v_x, |y_{i_1}-y_{i_2}|/v_y\}$;6)吊运任务必须起吊至安全高度后才能运输,运输必须停止才能将吊运任务放下,起吊和放下所需时间分别为 δ_{i_1} 和 δ_{i_2};7) 入库订单 Φ 和出库订单 Ω 同时生成;8)产品按照库位进行存储,所有库位均可被任一天车检索,库位最多可堆垛 l 层,存放于出库订单之上的产品为倒垛订单;9)每台天车每次只能执行一个任务;10)不考虑天车的失效和设备维护.

为方便描述, 定义符号如下:T 为所有订单集合;α_i 为订单 i 的单位等待成本, 其中 $i\in\Omega$;β_i 为订单 i 的单位延迟成本,其中 $i\in\Phi$;$z_{(X,Y)}(t)$ 为库位(X, Y)在时刻 t 的层高.决策变量:s_i 为任务 i 的开始时间;a_i 代表执行任务 i 的天车编号;$x_k(t)$ 和 $y_k(t)$ 分别代表天车 k 在时刻 t 时 x 方向和 y 方向上的位置.

由于订单库位的存储特性, 任意订单 i 和 j 之间均存在两种不同的存储状态,分别是:

$i<<j$:i 和 j 在不同库位;

$i\rightarrow j$:i 和 j 在相同库位且 i 的存储位在 j 之上.

根据库位存储特性,下层存储位订单必须在上层存储位订单起吊完成之后才能开始,即需满足:

$$\left.\begin{aligned}&s_{i_1}+\delta_{i_1}\leqslant s_{j_1}\\&(x_{i_1},y_{i_1})=(x_{j_1},y_{j_1})\end{aligned}\right\}\forall i\rightarrow j$$
$$(x_{i_1},y_{i_1})\neq(x_{j_1},y_{j_1}),\forall i<<j \tag{1}$$

根据假设 2)~假设 6), 天车需满足轨迹约束,包括:任意两台天车之间的安全距离约束,天车 x 和 y 方向上的速度约束,天车执行任务时的位置约束,以及变量 $x_k(t)$,$y_k(t)$ 的取值范围约束,即需满足:

$$\begin{aligned}&x_{k'}(t)-x_k(t)\geqslant(k'-k)\cdot\Delta d,\forall k',k\in C,k'>k\\&x_k(t)-v_x\Delta t\leqslant x_k(t+\Delta t)\leqslant x_k(t)+v_x\Delta t,\\&\qquad\forall t,\Delta t\geqslant 0,v_x\in\{v_{x,f},v_{x,e}\}\\&y_k(t)-v_y\Delta t\leqslant y_k(t+\Delta t)\leqslant y_k(t)+v_y\Delta t,\\&\qquad\forall t,\Delta t\geqslant 0,v_y\in\{v_{y,f},v_{y,e}\}\\&(x_{a_i}(t),y_{a_i}(t))=(x_i,y_i),\forall t\in[s_i,s_i+\delta_i],i\in T,a_i\in C\\&x_k(t),y_k(t)\in R\end{aligned} \tag{2}$$

根据假设 7)、假设 8),任意时刻的库位层高均不能超过其规定层高,即需满足:

$$0\leqslant z_{(x_{i_1},y_{i_1})}(t)\leqslant l,\forall k\geqslant 0,i\in T \tag{3}$$

调度目标由两部分组成,$g(s_{i_1})$ 为最小化入库订

单延迟成本,$f(s_{i_2})$为最小化出库订单等待成本.

$$g(s_{i_1}) = \min \sum_{i\in\Phi} \beta_{i_1} \cdot s_{i_1} \tag{4}$$

$$f(s_{i_2}) = \min \sum_{i\in\Omega} \alpha_{i_2} (s_{i_1} + \delta_{i_2}) \tag{5}$$

1.2 天车轨迹映射模型

天车的轨迹映射模型由 Peterson 等人[5]于 2014 年首次提出,其基本思想是变量之间的等效转换.通过研究约束条件与 s_i 和 a_i 之间的映射关系,实现变量 $x_k(t)$和 $y_k(t)$的等效消除,从而减少决策变量.

在本文中,由于天车满载和空载时速度不同,导致天车在执行不同任务时的初始状态不一,同时由于订单上下堆垛的存储特性,使得调度问题兼具时空复杂性,所以天车轨迹映射模型不仅需要考虑前后两个任务之间的关系,还需考虑任务初始态和时空约束.

给定任意两个有时空关系的订单 i 和 j,共生成 4 个任务:i_1,i_2,和 j_1,j_2.天车执行任务时将产生 4 种任务关系,分别为:

1)i_1 和 i_2

$$s_{i_1}+\delta_{i_1}+\max\left\{\frac{|x_{i_1}-x_{i_2}|}{v_{x,f}},\frac{|y_{i_1}-y_{i_2}|}{v_{y,f}}\right\}\leqslant s_{i_2},\quad \forall i_1,i_2\geqslant T, a_{i_1}=a_{i_2} \tag{6}$$

2)i_2 和 j_1

$$s_{i_2}+\delta_{i_2}+\max\left\{\frac{|x_{i_2}-x_{j_1}|}{v_{x,e}},\frac{|y_{i_2}-y_{j_2}|}{v_{y,e}}\right\}\leqslant s_{j_1},\quad \forall i_1,i_2\geqslant T, a_{i_1}=a_{i_2} \tag{7}$$

$$\begin{cases} s_{i_2}+\delta_{i_2}+\dfrac{x_{i_2}+(a_{j_1}-a_{i_2})\cdot\Delta d-x_{j_1}}{v_{x,e}}\leqslant s_{j_1} \\ \text{or} \\ x_{i_2}+(a_{j_1}-a_{i_2})\cdot\Delta d-x_{j_1} \quad \forall i_2,j_1\in T, a_{i_2}<a_{j_1} \\ \text{or} \\ s_{j_1}+\delta_{j_1}+\dfrac{x_{i_2}+(a_{j_1}-a_{i_2})\cdot\Delta d-x_{j_1}}{v_{x,f}}\leqslant s_{j_2} \end{cases} \tag{8}$$

3)i_1 和 j_1

$$s_{i_2}+\delta_{i_2}+\max\left\{\frac{|x_{i_2}-x_{j_1}|}{v_{x,e}},\frac{|y_{i_2}-y_{j_1}|}{v_{y,e}}\right\}\leqslant s_{j_1},\quad \forall i_1,j_1\geqslant T, a_{i_1}=a_{j_1} \tag{9}$$

$$\begin{cases} s_{i_1}+\delta_{i_1}+\dfrac{x_{i_1}+(a_{j_1}-a_{i_1})\cdot\Delta d-x_{j_1}}{v_{x,f}}\leqslant s_{j_1} \\ \text{or} \\ x_{i_1}+(a_{j_1}-a_{i_1})\cdot\Delta d-x_{j_1} \quad \forall i_1,j_1\in T, a_{i_1}<a_{j_1} \\ \text{or} \\ s_{j_1}+\delta_{j_1}+\dfrac{x_{i_1}+(a_{j_1}-a_{i_1})\cdot\Delta d-x_{j_1}}{v_{x,f}}\leqslant s_{j_1} \end{cases} \tag{10}$$

$$\begin{cases} s_{i_1}+\delta_{i_1}+\dfrac{(a_{j_1}-a_{i_1})\cdot\Delta d}{v_{x,f}}\leqslant s_{j_1} \\ \text{or} \quad \forall i_1,j_1\in T, x_{i_1}=x_{j_1}, a_{i_1}<a_{j_1} \\ s_{j_1}+\delta_{j_1}+\dfrac{(a_{j_1}-a_{i_1})\cdot\Delta d}{v_{x,f}}\leqslant s_{i_1} \end{cases} \tag{11}$$

4)i_2 和 j_2

$$s_{i_2}+\delta_{i_2}+\max\left\{\frac{|x_{i_2}-x_{j_1}|}{v_{x,e}},\frac{|y_{i_2}-y_{j_1}|}{v_{y,e}}\right\}\leqslant s_{i_2},\quad \forall i_2,j_2\geqslant T, a_{i_2}=a_{j_2} \tag{12}$$

$$\begin{cases} s_{i_2}+\delta_{i_2}+\dfrac{x_{i_2}+(a_{j_2}-a_{i_2})\cdot\Delta d-x_{j_2}}{v_{x,f}}\leqslant s_{j_2} \\ \text{or} \\ x_{i_2}+(a_{j_2}-a_{i_2})\cdot\Delta d-x_{j_2} \quad \forall i_2,j_2\in T, a_{i_2}<a_{j_2} \\ \text{or} \\ s_{j_2}+\delta_{j_2}+\dfrac{x_{i_2}+(a_{j_2}-a_{i_2})\cdot\Delta d-x_{j_2}}{v_{x,f}}\leqslant s_{i_2} \end{cases} \tag{13}$$

$$\begin{cases} s_{i_2}+\delta_{i_2}+\dfrac{(a_{j_2}-a_{i_2})\cdot\Delta d}{v_{x,f}}\leqslant s_{j_2} \\ \text{or} \quad \forall i_2,j_2\in T, x_{i_2}=x_{j_2}, a_{i_2}<a_{j_2} \\ s_{j_2}+\delta_{j_2}+\dfrac{(a_{j_2}-a_{i_2})\cdot\Delta d}{v_{x,f}}\leqslant s_{i_2} \end{cases} \tag{14}$$

这里,i_1 和 i_2 为同一台天车先后执行的两个任务.其中,式(6)、式(7)、式(9)、式(12)确保当两个任务由同一台天车执行时,天车执行完第一个任务后有足够的时间执行第二个任务;式(8)、式(10)、式(13)表示当两个任务不能同时完成时,一台天车必须等待另一台天车任务完成后才可以执行任务;式(11)、式(14)表示库位相同时前后两任务的时间约束.

根据天车和库位状态以及任务的开始时间,对式(6)~式(14)进行整理,得

$$s_{i_1}\geqslant \max\{B^1_{i_1}(j_1), B^2_{i_1}(j_2)\},\ \forall j<i \tag{15}$$

式中:

$$B^1_{i_1}(j_1)=\begin{cases} s_{j_1}+\delta_{j_1}+\dfrac{x_{i_1}+(a_{j_1}-a_{i_1})\cdot\Delta d-x_{j_1}}{v_{x,f}}, & \forall a_{i_1}<a_{j_1}, (i_1|j_1)_x>0 \\ s_{j_1}+\delta_{j_1}+\dfrac{x_{j_1}+(a_{i_1}-a_{j_1})\cdot\Delta d-x_{i_1}}{v_{x,f}}, & \forall a_{i_1}<a_{j_1}, (j_1|i_1)_x>0 \\ s_{j_1}+\delta_{j_1}+\dfrac{|(a_{j_1}-a_{i_1})\cdot\Delta d|}{v_{x,f}}, & \forall a_{i_1}\neq a_{j_1}, x_{i_1}=x_{j_1} \\ s_{j_1}, & \text{其他} \end{cases}$$

$$B^2_{i_1}(j_2)=\begin{cases} s_{j_2}+\delta_{j_2}+\max\left\{\dfrac{|x_{i_2}-x_{i_1}|}{v_{x,e}},\dfrac{|y_{j_2}-y_{i_1}|}{v_{y,e}}\right\}, & \forall a_{i_1}=a_{j_2} \\ s_{j_2}+\delta_{j_2}+\dfrac{x_{i_1}+(a_{j_2}-a_{i_1})\cdot\Delta d-x_{j_2}}{v_{x,e}}, & \forall a_{i_1}<a_{j_2}, (i_1|j_2)_x>0 \\ s_{j_2}+\delta_{j_2}+\dfrac{x_{j_2}+(a_{i_1}-a_{j_2})\cdot\Delta d-x_{i_1}}{v_{x,e}}, & \forall a_{i_1}>a_{j_2}, (j_2|i_1)_x>0 \\ s_{j_2}, & \text{其他} \end{cases}$$

$$s_{i_2}\geqslant\max\{B^1_{i_2}(i_1),B^2_{i_2}(j_1),B^3_{i_2}(j_2)\},\ \forall j<i \tag{16}$$

式中:

$$B^2_{i_2}(i_1)=s_{i_1}+\delta_{i_1}+\max\left\{\frac{|x_{i_1}-x_{i_2}|}{v_{x,f}},\frac{|y_{i_1}-y_{i_2}|}{v_{y,f}}\right\},\ \forall a_{i_2}=a_{j_1}$$

$$B^2_{i_2}(j_1)=\begin{cases}s_{j_1}+\delta_{j_1}+\dfrac{x_{i_2}+(a_{j_1}-a_{i_2})\cdot\Delta d-x_{j_1}}{v_{x,f}},\ \forall a_{i_2}<a_{j_1},(i_2|j_1)_x>0\\ s_{j_1}+\delta_{j_1}+\dfrac{x_{j_1}+(a_{i_2}-a_{j_1})\cdot\Delta d-x_{i_2}}{v_{x,f}},\ \forall a_{i_2}<a_{j_1},(j_1|i_2)_x>0\\ s_{j_1},\qquad\qquad\text{其他}\end{cases}$$

$$B^2_{i_2}(j_2)=\begin{cases}s_{j_2}+\delta_{j_2}+\dfrac{x_{i_2}+(a_{j_2}-a_{i_2})\cdot\Delta d-x_{j_2}}{v_{x,f}},\ \forall a_{i_2}<a_{j_2},(i_2|j_2)_x>0\\ s_{j_2}+\delta_{j_2}+\dfrac{x_{j_2}+(a_{i_2}-a_{j_2})\cdot\Delta d-x_{i_2}}{v_{x,f}},\ \forall a_{i_2}<a_{j_2},(j_2|i_2)_x>0\\ s_{j_2}+\delta_{j_2}+\dfrac{|(a_{i_2}-a_{j_2})\cdot\Delta d|}{v_{x,f}},\ \forall a_{i_2}\neq a_{j_2},x_{i_2}=x_{j_2}\\ s_{j_2},\qquad\qquad\text{其他}\end{cases}$$

其中,$B^N_i(j),N=1,2,3$ 表示任务 i 在任务 j 之后开始,$(i|j)_x>0$ 表示 $x_i+(a_j-a_i)\cdot\Delta d-x_j>0$.

由此,通过建立天车轨迹映射模型,调度问题的约束条件可转化为式(3)、式(15)、式(16),决策变量减少为:任务 i 的开始时间 s_i 和执行任务 i 的天车 a_i.

2　算法构建

本文研究的车间天车调度需解决 3 个子问题:1)各订单的执行序列;2)各订单的天车分配;3)入库订单的库位分配. 该问题属于典型的组合优化难题,用传统解析数学方法无法对其有效求解.差分进化(Differential Evolution,DE)是一种过程简单、受控参数少和鲁棒性强的全局优化算法,与其他进化算法相比,DE 在解决复杂全局优化问题时的性能更加优秀,目前已广泛应用于各类调度问题[9-10].本文在构建天车轨迹映射模型的基础上,以 DE 为框架,提出了基于泛化反向学习的差分进化算法,并设计天车分配算法和相应的库位分配规则以指导算法寻优.

2.1　库位分配规则

库位分配规则在式(15)、式(16)的基础上,以天车路径不冲突和负重移动距离最短为原则,可在调度过程中解决子问题 3).为方便描述,作如下定义:

定义 1　中心搜寻规则(center search rule,CS).记订单当前所在库位为(X,Y),CS 规则为先固定 X 列,令 $i=1$,搜寻库位$\{(X,Y-i),(X,Y+i)\}$,选择可用空间大的库位作为订单分配的位置.若无可用空间,则令 $i=i+1$,继续搜寻直至 X 列搜寻完毕.若当前 X 列均无可用空间,则按同样的方法搜寻 $X+1$ 列和 $X-1$ 列,直至找到可用空间为止.

定义 2　中心向右搜寻规则(center to right search rule,CTR).基本步骤与 CS 相同,当 X 列搜寻完毕后,CTR 只搜寻 $X+1$ 列,直至找到可用空间.

定义 3　中心向左搜寻规则(center to left search rule,CTL).基本步骤与 CS 相同,当 X 列搜寻完毕后,CTL 只搜寻 $X-1$ 列,直至找到可用空间.

假设种群规模为 NP,基因数为 nVar,当前库位为 $S_{(X,Y)}$,其基本步骤为:

```
1)  for p=1:NP
2)   for k=1:2*nVar
3)    if k∈{i2}, i2∈T\Ω
4)      记第 j 个任务分配的天车为 aj;
5)      记 aj 的前置任务为 i;
6)      if  ai > aj ∧ |xj − xi| < (ai − aj)·d
7)      S(X,Y) = (xi − (ai − aj)·d, yj);
8)       根据 CTL 为 j 分配库位;
9)      else if  ai < aj ∧ |xj − xi| < (aj − ai)·d
10)        S(X,Y) = (xi + (aj − ai)·d, yj);
11)        根据 CTR 为 j 分配库位;
12)     else
13)     S(X,Y) = (xj, yj);
14)       根据 CS 为 j 分配库位;
15)     end if
16)    end if
17)  end for
18) end for
```

2.2　天车分配算法

天车分配算法可在较短的时间内解决子问题 2),并得到较为优质的天车分配方案.基本步骤为:

步骤 1　为平衡各天车间的任务负载,按照天车编号,将 K 台天车按照 1,2,…,K 进行循环升序排列,直至排列总数达到 N.

步骤 2　取序列中的第一台天车成对进行排列.

步骤 3　令 $i=2$,将序列中第 i 台天车成对插入到前面 $i\times(i+1)/2$ 个可能的位置,结合库位分配规则为入库和倒垛订单分配库位,利用快速非支配排序取目标值较优的排列作为新的排列.

步骤 4　令 $i=i+1$,重复步骤 3,直至所有天车分配完毕.

2.3　基于泛化反向学习的差分进化算法

泛化反向学习(Generalized Opposition Based Learning,GOBL)指的是在种群的初始化阶段和代跳跃阶段使用 GOBL 机制生成变换种群,对于种群 P_G,其变换种群 $\mathrm{GOP}_{r,j,G}$ 为:

$$\mathrm{GOP}_{r,j,G}=\begin{cases}k(a_j+b_j)-P_{r,j,G}, \text{if } \mathrm{GOP}_{r,j,G}\in[a_j,b_j]\\ \mathrm{rand}(a_j,b_j),\ \text{otherwise}\end{cases}$$

式中:$k=\mathrm{rand}(0,1)$;$P_{r,j,G}\in[a_j,b_j]$.基于 GOBL 的差分进化在每一次迭代中融合库位分配规则和天车分配算法以指导寻优进程.

步骤 1 编码.采用双层编码机制,如图 2 所示.第一层编码中,订单总数为 N,每个订单的第一个位置为订单编号,第二个位置为随机实数.所有随机实数从小到大排列,决定了订单的执行序列.第二层编码为天车编号,编码长度为 $2N$,代表 $2N$ 个任务(对应 N 个订单)分配的天车.

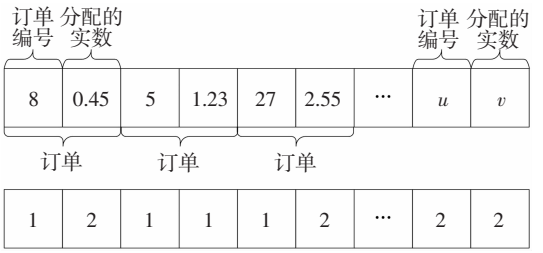

图 2 编码方式

Fig.2 Encoding presentation

上下层编码的对应关系为:从下层编码开始,第一个天车编号对应第一个订单的第一个任务,第二个天车编号若与第一个相同,则执行上层编码第一个订单的第二个任务,否则执行第二个订单的第一个任务,依此类推.

步骤 2 初始化.1)针对上层种群,采用随机生成的方法,为每个订单分配一个随机实数,并将其从小到大排列,产生一个初始目标向量.针对下层种群,①令 i=1,生成[1,K]之间的随机整数,填入位置 i;②令 $i=i+1$,重复步骤①,直至 $i=2N$.上下层目标向量共同构成种群 P_0.2)采用 GOBL 机制生成 P_0 的变换种群 GOP_0;3) 结合库位分配规则, 为 P_0 和 GOP_0 分配库位;4)同时评价 P_0 和 GOP_0,利用快速非支配排序选择最优的 NP 个个体组成新的种群 P_0',将 P_0'作为初始种群.

步骤 3 变异.传统 DE 算法的变异率 F 一般为固定实数, 以对差分量进行缩小和放大控制.F 过大,算法近似随机搜索,最优解精度降低;F 过小,种群多样性降低,算法易早熟收敛.故本文提出自适应变异算子 F_1 和学习算子 F_2 进行改善. 在算法初期,F_1 具有较大值,有利于保持个体多样性,随着算法的推进,F_1 逐步减小, 可以增加搜索到全局最优解的概率.F_2 则根据历代的状态调节数值大小,学习订单在前几代中的排序.根据文献[11]的设计规则,得:

$$F_1=F_0\cdot 2^{\lambda},\lambda=\mathrm{e}^{1-\frac{G_m}{G_m+1-G}}$$

$$F_2=\sin(\Delta k_j/|\Delta k_j^{\max}|\cdot\pi)$$

式中:F_0 为缩放因子;G_m 为外层最大进化代数;G 为当前外层代数;Δk_j 为订单 j 在第 G 代和第 G–1 代的位置差;$\Delta k_j^{\max}$ 为订单 j 在历代进化中的最大位置差.

变异机制为:

$$v_{r,j,G+1}=x_{r,j,G}+F_1(x_{\mathrm{best},j,G}-x_{r,j,G})+F_2(x_{r_1,j,G}-x_{r_2,j,G})$$

式中:$x_{r,j,G}$,$x_{\mathrm{best},j,G}$,$x_{r_1,j,G}$,$x_{r_2,j,G}$ 分别表示第 G 代中个体 r,best,r_1,r_2 第 j 个订单的基因,$r\neq r_1\neq r_2$.

步骤 4 交叉.对父代种群 P_G 中的个体 $x_{r,G}$ 及变异的中间体 $v_{r,G+1}$ 进行个体间交叉操作,得到子代种群 P_{G+1},用公式表示为:

$$u_{r,j,G+1}=\begin{cases}v_{r,j,G+1}, \text{if } \mathrm{rand}(j)\leqslant \mathrm{CR} \text{ or } j=\mathrm{rd}(j)\\ x_{r,j,G},\ \text{otherwise}\end{cases}$$

式中:CR 为交叉因子;rand(j)为评价订单 j 时产生的随机数;rd(j)为随机选择的整数.

步骤 5 结合天车分配算法和库位分配规则为交叉得到的个体分配天车和库位.同时,令 $P=P_G+P_{G+1}$,从而得到待评价种群 P.

步骤 6 种群代跳跃.在种群代跳跃阶段,每次均需生成一个 0 到 1 之间的随机数 rand (0,1).若 $\mathrm{rand}(0,1)<J$,则根据 GOBL 机制生成 P_{G+1} 的变换种群 GOP_{G+1},然后执行步骤 7.若 $\mathrm{rand}(0,1)\geqslant J$,则执行步骤 8.这里,$J$ 为代跳跃概率.

步骤 7 结合天车分配算法和库位分配规则,为变换种群 GOP_{G+1} 中的每个个体分配天车和库位,并更新待评价种群 $P=P\cup\mathrm{GOP}_{G+1}$.

步骤 8 选择.根据 Pareto 非支配排序和拥挤距离排序选择种群 P 中最优的 NP 个个体进入下一代.

算法流程图如图 3 所示.

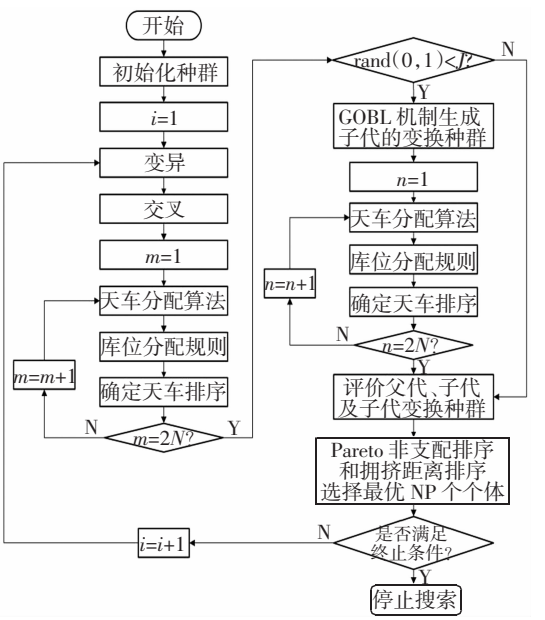

图 3 算法流程图

Fig.3 Framework of the algorithm

3　仿真试验分析

3.1　参数分析

本试验在基于 Windows 10 操作系统的 Core i5/2.5GHz 内存 4 GB 的计算机上进行.

由于此问题的真实帕累托前沿很难得到,本文在进行试验时采用以下方法获得近似帕累托前沿:算法独立运行多次后记录每次的帕累托解集,从所有帕累托前沿中获得新的帕累托解作为近似前沿.试验参数按照文献[11-13]中的设计规则,选取不同的参数组合进行多次试验. 当设置种群规模 nPop = 50,迭代次数 G = 3 000,种群代跳跃概率 J = 0.3,缩放因子 F_0 = 0.5,交叉概率 CR = 0.6,最大与最小允许变异的值 $mu_{\max}$、$mu_{\min}$ 分别为 4 和 0 时, 算法以较高质量求解.

为测试天车分配算法以及库位分配规则在求解问题中的表现,设计未使用天车分配算法、未使用库位分配规则的差分进化(GOBL-DE)进行对比,得到不同订单组合下的目标函数和求解时间,试验结果如表 1 所示.

表 1　不同问题规模和订单组合下的对比结果

Tab.1　Comparison of experimental results between various tasks and order combination

任务数	订单组合	GOBL-DECS			GOBL-DE		
		$\bar{g}(s_{i_1})$	$\bar{f}(s_{i_2})$	CPU(s)	$\bar{g}(s_{i_1})$	$\bar{f}(s_{i_2})$	CPU(s)
30	C(18,6,6)	65	222	618	246	489	547
	C(12,6,12)	91	134	599	318	276	541
50	C(30,10,10)	239	614	660	649	1 789	600
	C(20,10,20)	475	329	666	1 236	984	581
70	C(30,20,20)	1 551	944	736	3 168	2 421	624
	C(40,10,20)	583	2 354	757	1 925	3 735	633
90	C(30,30,30)	5 683	1 030	777	7 877	3 468	651
	C(30,20,40)	2 058	2 236	811	4 879	4 322	661

由表 1 可知,结合了天车分配算法和库位分配规则的 GOBL-DECS(GOBL-DE-Crane allocation algorithm-Stock allocation rule)在对作业进行调度时,平均延迟成本和平均等待成本均明显低于 GOBL-DE 算法.这是由于 GOBL-DE 算法在求解时,无法合理分配天车以及入库订单和倒垛订单的位置,导致天车避让和空行程的增加.

对任务数为 50,订单组合为 C(30,10,10)的情况进行分析,得到迭代结束时两种算法的帕累托前沿,如图 4 所示.从图中可以看出,GOBL-DECS 算法的帕累托前沿能向更优解逼近. 尽管 GOBL-DECS 求解时增加了一定的时间成本,但算法表现却得到了极大的提高. 可见本文提出的天车分配算法和库位分配规则在解决车间多天车多任务调度问题时具有良好的性能.

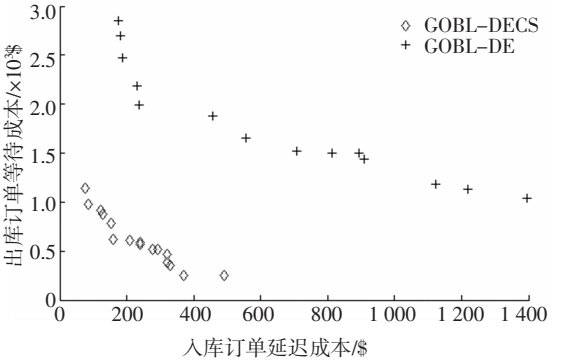

图 4　订单组合为 C(30,10,10)的帕累托前沿

Fig.4　Pareto frontier of order combination C(30,10,10)

3.2　数值分析

为测试本文提出的算法在求解车间天车多目标调度问题时的整体性能,设计具有代表性的多目标优化 NSGA-Ⅱ和 SPEA-Ⅱ算法进行对比计算试验. 以帕累托解的个数 (NF), 解的平均拥挤距离(crowding distance,CD,CD 越大代表解的分散性越好),单次迭代运行时间($\overline{\text{CPU}}$)作为评价算法性能的指标,取任务数分别为 10、30、50、70、90 5 种情况下的调度问题进行仿真试验,结果如表 2 所示.

表 2　3 种算法的数值对比结果

Tab.2　Numerical calculation results of 3 algorithms

任务数	GOBL-DECS			NSGA-Ⅱ			SPEA-Ⅱ		
	NF	$\overline{\text{CD}}$	$\overline{\text{CPU}}$/s	NF	$\overline{\text{CD}}$	$\overline{\text{CPU}}$/s	NF	$\overline{\text{CD}}$	$\overline{\text{CPU}}$/s
10	8	0.653	0.020	6	0.602	0.018	8	0.651	0.011
30	17	0.224	0.164	14	0.111	0.109	27	0.126	0.084
50	20	0.589	0.259	15	0.210	0.204	9	0.529	0.161
70	19	0.343	0.300	19	0.263	0.222	23	0.318	0.166
90	14	0.387	0.352	12	0.296	0.308	8	0.324	0.217

由表 2 可知,在不同任务数组合下,本文提出的 GOBL-DECS 算法得到的帕累托解的个数和解的平均拥挤距离均优于 NSGA-Ⅱ.与 SPEA-Ⅱ对比,GOBL-DECS 的 $\overline{\text{CD}}$ 值也更优.尽管 GOBL-DECS 算法在时间表现上稍显劣势,但结合考虑解的优度之后,本文认为 GOBL-DECS 在时间上的差距对计算成本影响甚微.

图 5 和图 6 分别以任务数为 30 和 90 的一组数据为例,给出了分别运行 GOBL-DECS、NSGA-Ⅱ和 SPEA-Ⅱ3 种算法的帕累托解的情况. 从图中可

以看出,NSGA-Ⅱ和 SPEA-Ⅱ在算法表现上较为相近. 当问题规模较小时,GOBL-DECS 算法和其他两种算法得到的部分解集相互支配, 但总体而言 GOBL-DECS 得到的解集更优. 随着问题规模的增大,GOBL-DECS 算法的优势更加明显,具有更好的跳出局部最优解的能力,可以引导种群向更优的帕累托前沿逼近.

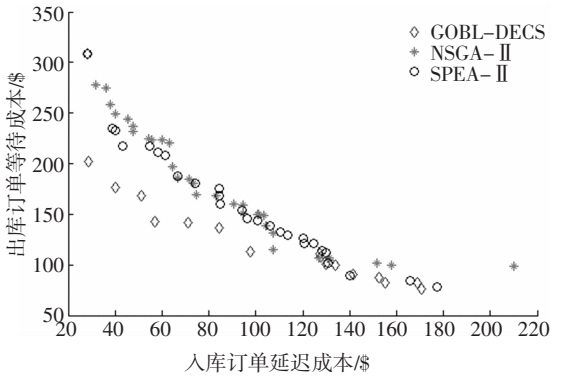

图 5 任务数为 30 的 3 种对比算法的帕累托前沿

Fig.5 Pareto frontier of three algorithms with 30 tasks

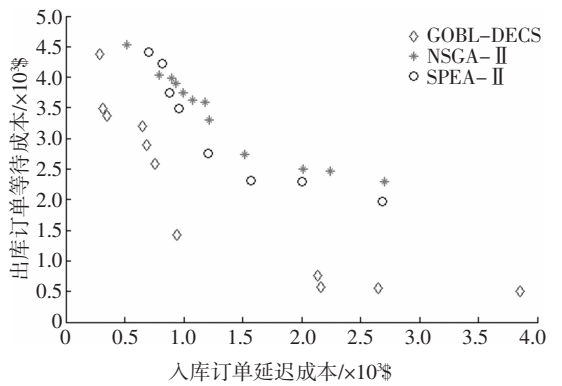

图 6 任务数为 90 的 3 种对比算法的帕累托前沿

Fig.6 Pareto frontier of three algorithms with 90 tasks

4 结 论

1)建立了考虑天车任务初始态和时空约束的轨迹映射模型,采用双层编码机制,将订单排序和天车分配问题映射统一,为解决车间天车调度问题提供了新的研究思路.

2)仿真试验的对比验证表明了算法具有良好的求解性能,证明了算法的有效性和可行性.

3)研究补充了车间天车调度中联合调度入库、出库作业的短缺,丰富了此类调度问题的理论方法.

4)本文仅对单一运行速度下的天车调度进行了探讨,今后将结合能源节约的目标,对搬运过程中速度可变的车间多天车调度问题进行深入研究.

参考文献

[1] KITAMURA S,MURAO H. Simulation-based optimization model and metaheuristic solution of multiple crane scheduling problems [C]//Proceedings of 2004 IEEE International Conference on Systems,Man and Cybernetics. Washington D C:IEEE,2004:1469—1474.

[2] FENG J,CHU C,CHE A. Cyclic job shop hoist scheduling with multi-capacity reentrant tanks and time-window constraints [J]. Computers & Industrial Engineering,2018,120:382—391.

[3] ZHOU Z,LI L. A solution for cyclic scheduling of multi-hoists without overlapping [J]. Annals of Operations Research,2009,168(1):5—21.

[4] MAO Y N,TANG Q H,LI Z X,*et al*. Mixed-integer linear programming method for multi-degree and multi-hoist cyclic scheduling with time windows [J]. Engineering Optimization,2018,50(11):1—18.

[5] PETERSON B,HARJUNKOSKI I,HODA S,*et al*. Scheduling multiple factory cranes on a common track [J]. Computers & Operations Research,2014,48:102—112.

[6] YANG X,LI S. Research of crane scheduling based on birth and death chain in the production shop field [C]//Proceedings of 2015 International Conference on Computer Science and Mechanical Automation (CSMA). Washington D C:IEEE,2015:246—250.

[7] XIE X,ZHENG Y,LI Y. Multi-crane scheduling in steel coil warehouse [J]. Expert Systems with Applications,2014,41(6):2874—2885.

[8] YUAN Y,TANG L. Novel time-space network flow formulation and approximate dynamic programming approach for the crane scheduling in a coil warehouse [J]. European Journal of Operational Research,2017,262(2):424—437.

[9] LIN Q,ZHU Q,HUANG P,*et al*. A novel hybrid multi-objective immune algorithm with adaptive differential evolution [J]. Computers & Operations Research,2015,62:95—111.

[10] ZHOU B H,SHEN C Y. Multi-objective optimization of material delivery for mixed model assembly lines with energy consideration [J]. Journal of Cleaner Production,2018,192:293—305.

[11] 周炳海,王科.带多重加工前约束的单机 MOPJ 调度方法[J].哈尔滨工业大学学报,2017,49(7):158—164.
ZHOU B H,WANG K. Scheduling method of multi-order-per-job for a single machine with multiple preprocess constraints [J]. Journal of Harbin Institute of Technology,2017,49(7):158—164. (In Chinese)

[12] 吴亮红,王耀南,袁小芳,等.多目标优化问题的差分进化算法研究[J]. 湖南大学学报(自然科学版),2009,36(2):53—57.
WU L H,WANG Y N,YUAN X F,*et al*. Research on differential evolution algorithm for MOPs [J]. Journal of Hunan University (Natural Sciences),2009,36(2):53—57.(In Chinese)

[13] MEMARI A,RAHIM A R A,HASSAN A,*et al*. A tuned NSGA-II to optimize the total cost and service level for a just-in-time distribution network [J]. Neural Computing and Applications,2017,28(11):3413—3427.