

2-23-2022

Modeling and Optimization for Manufacturing Cell Scheduling Based on Improved Wolf Pack Algorithm and Simulation

Zi'an Zhao

School of Economics and Management, Beijing University of Aeronautics and Astronautics, Beijing 100191, China; ZY2008222@buaa.edu.cn

Hong Zhou

School of Economics and Management, Beijing University of Aeronautics and Astronautics, Beijing 100191, China; h_zhou@buaa.edu.cn

Yingjian Lei

School of Economics and Management, Beijing University of Aeronautics and Astronautics, Beijing 100191, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Invited Papers & Special Columns is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Modeling and Optimization for Manufacturing Cell Scheduling Based on Improved Wolf Pack Algorithm and Simulation

Abstract

Abstract: Aiming at the domestic aircraft stall spin simulation training need, a stall spin simulation training system is developed. *The training system consists of the multi-channel dome visual system, the semi-physical simulation cockpit and the maneuvering force control loading system, etc. Distributed simulation technology is used to develop a realistic man-in-the-loop simulation training environment. For the stall spin simulation, the multi-source aerodynamic data is processed comprehensively, and an unsteady aerodynamic model at high angle of attack (AOA) is constructed, and the heavy-load digital electric control loading technology is used to realize the simulation of stall spin alternating force and jitter characteristics of steering, which realizes the stall spin simulation training of man-in-the-loop.*

Keywords

stall spin, simulation and training, maneuvering force control loading, unsteady flow aerodynamics, man-in-the-loop

Recommended Citation

Zi'an Zhao, Hong Zhou, Yingjian Lei. Modeling and Optimization for Manufacturing Cell Scheduling Based on Improved Wolf Pack Algorithm and Simulation[J]. Journal of System Simulation, 2022, 34(2): 201-211.

基于改进狼群算法与仿真的单元调度优化

赵梓安, 周泓, 雷颖健

(北京航空航天大学 经济管理学院, 北京 100191)

摘要: 单元制造是现代生产系统的重要组织形式。对制造单元的调度过程中, 机器故障与中断是很常见的问题, 机器空转造成的能源浪费也不可忽视, 相关研究具有重要现实意义。考虑单元调度中的机器中断和能耗, 建立了一个最小化所有工件完工时间及机器闲置能源消耗和中断成本的整数规划模型, 并提出了一种基于改进狼群算法和离散事件仿真的混合优化方法加以求解, 有效提高了算法的寻优能力。数值实验表明, 提出的算法具有良好的收敛性, 可以在合理的迭代次数内找到问题的满意解。

关键词: 单元制造系统; 单元调度; 改进狼群算法; 离散事件仿真; 机器中断; 机器耗能

中图分类号: TP391.9

文献标志码: A

文章编号: 1004-731X(2022)02-0201-11

DOI: 10.16182/j.issn1004731x.joss.21-0372

Modeling and Optimization for Manufacturing Cell Scheduling Based on Improved Wolf Pack Algorithm and Simulation

Zhao Zi'an, Zhou Hong, Lei Yingjian

(School of Economics and Management, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: Cell manufacturing is an important organizational form of modern production systems. In scheduling of cell manufacturing systems, machine failures or interruptions are very common in practice, meanwhile the waste due to energy consumption during machine idle time cannot be ignored. Hence the relevant research is with strong significance. *This paper considers the problems of machine interruption and energy consumption in cell scheduling, and developed an integer programming model to minimize the makespan as well as the cost of energy consumption during machine idling and the interruption cost. A mixed optimization method is proposed based on improved wolf pack algorithm and discrete event simulation to solve the problem, which can effectively improve the optimization performance of the algorithm. Numerical experiments demonstrate that the proposed hybrid algorithm shows a good convergence, and a satisfactory solution to the problem can be reached within a reasonable number of iterations.*

Keywords: cellular manufacturing systems; cell scheduling; improved wolf pack algorithm; discrete event simulation; machine interruption; energy consumption

引言

单元制造系统(CMS)是现代生产系统中一种重要的组织形式, 是准时制制造、精益制造和成

组技术的结合应用。为了应对激烈的市场竞争和个性化需求, 许多企业采用了这一模式, 尤其是拥有多用途生产设施及产品多样性程度较高的大型制造中心^[1]。

收稿日期: 2021-04-27

修回日期: 2021-08-11

基金项目: 国家重点研发计划资助(2018YFB1701600)

第一作者: 赵梓安(1998-), 男, 硕士生, 研究方向为生产系统建模优化与仿真。E-mail: ZY2008222@buaa.edu.cn

通讯作者: 周泓(1965-), 男, 博士, 教授, 研究方向为生产与物流系统建模优化与仿真。E-mail: h_zhou@buaa.edu.cn

单元调度是单元制造系统中最主要的管理问题之一,得到了广泛关注和研究。例如, Feng 等提出了一种 3 层染色体遗传算法来解决所建立的混合整数规划调度模型^[2],而 Sinaki 等提出了一种加权多目标混合整数非线性规划模型来解决单元调度问题^[3]。长期以来,对于单元制造系统的研究通常集中在与效率和产出相关的性能指标方面。根据国际能源署(IEA)的估计,到 2040 年,工业部门将占能源消耗的最大份额(近 50%)^[4],制造业的能源消耗已成为越来越重要的问题。因此,有必要充分考虑能源消耗问题以实现可持续生产。Iqbal 等通过设计两阶段方法来实现单元制造系统中的节能,其模型是最小化空闲运行的时间来实现最佳作业顺序的安排^[5]。Liu 等研究了通过关闭闲置较长时间的机器来实现节能,还考虑了关闭和重新打开机器将消耗更多的能量,并使用多目标遗传算法对所构建的模型进行求解^[6]。基于已有研究, Gahm 等对节能调度领域的相关研究进行了较为全面的综述^[7]。在制造过程中,加工可能会因为机器故障而中断和延迟,机器故障的处理是生产调度中的一个难题, Yang 等在模型中同时考虑了完工时间和鲁棒性 2 个目标,并且也考虑了机器故障的概率^[8]。Tadayonirad 等研究了具有随机机器故障的调度问题,并集成元启发式算法和仿真方法^[9]。Li 等研究了同时考虑动态任务到达和机器故障的生产优化方法,其中机器故障是考虑机床空转或者待机^[10]。Sarker 等提出了一种混合进化算法来求解考虑机器故障维护的作业调度问题^[11]。本文所研究的单元调度优化问题通过最小化机器的闲置时间来优化总能耗,同时考虑了机器故障对加工完成时间的影响,以及维修成本等问题。

1 问题建模

1.1 问题描述

本文所研究的是单元制造系统中的作业调度

问题。该问题通过确定工件以何种顺序在不同单元内的机器上加工,能够使得 CMS 的性能指标达到最优。本文首先选择最小化所有工件在 CMS 上的加工完成时间作为优化目标,这是作业调度问题的常用性能指标。此外,假设机器上的操作主要是机械加工,因此工具经常被磨损,在加工过程中需要更换,从而会引起机器加工过程的中断,由此会带来机器中断成本;同时,考虑机器空转时存在能源浪费,也会带来相应成本。因此,本文将机器中断成本及空转时的能源消耗成本也作为优化目标,使其最小化。

假设单元配置是确定且事先已知的,工件在单元内的移动时间忽略不计;每台机器一次最多可以处理一个操作,且每个操作最多由一台机器处理,机器调整时间与操作顺序无关且假设包含在加工时间中;机器中断(或故障)取决于已加工完成的作业数量,即在处理完若干次数的操作后会发生中断;机器在空闲(空转)时间内会消耗能源,中断过程会产生成本(维修或部件更换等)。

1.2 模型

首先将本文使用的符号概括如下。

参数:

n : 工件数量;

m : 机器数量;

c : 单元数量;

j : 工件下标 $j \in J = \{1, 2, \dots, n\}$;

k : 机器下标 $k \in M = \{1, 2, \dots, m\}$;

u : 单元下标 $u \in U = \{1, 2, \dots, c\}$;

n_j : 工件 j 的操作数量;

O_{ij} : 工件 j 第 i 项操作 $i \in I_j = \{1, 2, \dots, n_j\}$;

p_{ijk} : 操作 O_{ij} 在机器 k 上的加工时间;

d_j : 工件 j 的完工期限;

$t_{uu'}$: 工件从单元 u 到 u' 的转移时间;

C_1 : 机器空转时的单位时间能耗成本;

BR : 机器单位时间中断成本;

$$y_{ku} = \begin{cases} 1, & \text{如果机器 } k \text{ 属于单元 } u \\ 0, & \text{否则} \end{cases};$$

n_{ijk} : 操作 O_{ij} 开始时机器 k 上已经加工完的零件数;

B_{ij} : 操作 O_{ij} 之前已完成的操作集合;

N_k : 机器 k 加工 N_k 次中断一次;

TM_k : 机器 k 每次中断的平均时间;

s_{ij} : 操作 O_{ij} 的开始时间;

C_{ij} : 操作 O_{ij} 的完成时间;

决策变量:

$$x_{ijk} = \begin{cases} 1, & \text{如果操作 } O_{ij} \text{ 被分配给机器 } k \\ 0, & \text{否则} \end{cases};$$

$$z_{j'k} = \begin{cases} 1, & \text{如果工件 } j' \text{ 紧跟 } j \text{ 之后被分配给机器 } k \text{ 加工} \\ 0, & \text{否则} \end{cases};$$

本文模型的目标为最小化所有工件加工完成时间(也即工件的最大完成时间)以及总成本(包括机器中断成本和空转能源消耗成本), 也即最小化以下目标函数

$$f_1 = \max_j C_{n,j} \quad (1)$$

$$f_2 = \sum_{k \in M} \left\{ \sum_{j,j' \in I_j} \left(\sum_{i \in I_j} s_{ij} x_{ijk} - \sum_{i \in I_j} C_{ij} x_{ijk} \right) z_{j'k} \right\} \cdot C_1 + BR \cdot \sum_{k \in M} \left[\frac{\max_j (n_{n,jk} \times x_{n,jk})}{N_k} \right] TM_k \quad (2)$$

为了简化计算, 本文采用加权和的形式来简化处理多目标。所建模型如下:

$$\min f = w_1 f_1 + w_2 f_2 \quad (3)$$

s.t.

$$\sum_{k \in M} x_{ijk} = 1, \forall i \in I_j, j \in J \quad (4)$$

$$s_{ij} \geq (s_{ij} + p_{ijk}) z_{j'k} x_{ijk} x_{i'jk}, \forall k \in M, \{i, i'\} \in I_j, \{j, j'\} \in J \quad (5)$$

$$C_{ij} = s_{ij} + \sum_{k \in M} p_{ijk} x_{ijk} + \sum_{k \in M} \left[\frac{n_{ijk}}{N_k} \right] x_{ijk} \cdot TM_k, \forall i \in I_j, j \in J \quad (6)$$

$$s_{(i+1)j} - C_{ij} \geq x_{(i+1)jk} x_{ijk} y_{ku} y_{k'u'} t_{uu'}, \forall i \in I_j, j \in J, \{k, k'\} \in M, k \neq k', \{u, u'\} \in U \quad (7)$$

$$C_{n,j} \leq d_j, \forall j \in J \quad (8)$$

$$n_{ijk} = \sum_{O_{rv} \in B_{ij}} x_{rvk}, \forall k \in M \quad (9)$$

$$\sum_{i \in I_j} x_{ijk} + \sum_{i \in I_j} x_{ij'k} \geq z_{j'k} \times 2, \forall k \in M, \{j, j'\} \in J \quad (10)$$

$$x_{ijk}, z_{j'k} \in \{0, 1\}, \forall i \in I_j, \{j, j'\} \in J, k \in M \quad (11)$$

目标函数(1)表示总完工时间, 目标函数(2)第 1 项表示机器空闲能耗成本, 第 2 项表示机器加工中断成本。式(4)表示每项操作只能分配给一个机器。式(5)表示在同一台机器上, 仅当前一项操作完成后, 后一项操作才能开始。式(6)规定了 O_{ij} 操作的完成时间, 不仅需考虑加工时间, 还需考虑中断时间。式(7)表示如果相邻操作在不同单元的机器中完成, 则需考虑单元间的移动时间。式(8)表示工件加工过程应在所规定的时限内完成。式(9)表示操作 O_{ij} 开始时机器 k 上已经加工完的零件数。式(10)是相容性约束, 表示如果工件 j 和 j' 在机器 k 上相邻加工, 则 j 和 j' 都应有工序被分配给机器 k 。

2 混合优化算法设计

2.1 狼群算法的基本结构

狼群算法(WPA)通过模拟自然界中狼群捕食猎物的过程来对问题进行求解^[12]。狼群中个体分为头狼、探狼和猛狼, 它们通过 3 种智能行为共同捕食猎物。在每一代围攻行为之后, 狼群将通过优胜劣汰机制进行更新, 淘汰性能最差的 R 匹狼, 同时在解空间中随机产生 R 匹新狼进行补充, 完成种群更新。

狼群的 3 种智能行为分别如下。

2.1.1 游走行为

以一定的比例 α 在狼群中选取除头狼外最佳的 S_{num} 匹狼作为探狼。然后开始游走探测, 每个探狼各向 h 个方向分别前进一步(游走步长为 $step_a$), 记录每前进一步后的性能指标值。探狼 i 向第 $p(p=1, 2, \dots, h)$ 个方向前进后所处的位置通过式(12)计算得出

$$x_i^p = x_i + \sin(2\pi \frac{p}{h}) \text{step}_a \quad (12)$$

记探狼 i 在当前位置的性能指标值为 Y_i^p , 若 Y_i^p 优于当前头狼的性能指标值 Y_{lead} , 则更新头狼及其性能指标值, 也即令 $X_{\text{lead}} = x_i^p$, $Y_{\text{lead}} = Y_i^p$, 直至所有方向探测完毕, 也即 $p = h$ 。若在探狼 i 的探测过程中发生了头狼更替, 则终止游走行为阶段, 由探狼 i 取代头狼发起召唤, 进入召唤行为阶段; 否则, 下一匹探狼开始其探测过程。若所有 S_{num} 匹探狼完成探测后未改变当前头狼, 则进入下一轮游走, 直至达到最大游走次数 T_{max} 。

2.1.2 召唤行为

在召唤行为中, 头狼周围的 M_{num} 匹猛狼响应头狼召唤, 以较大的奔袭步长 step_b 快速向头狼所在位置靠拢, 其中猛狼 i 第 $k+1$ 次前进后所处的位置由式(13)计算得出

$$x_i(k+1) = x_i(k) + \frac{(X_{\text{lead}} - x_i(k))}{|X_{\text{lead}} - x_i(k)|} \text{step}_b \quad (13)$$

在猛狼奔袭途中, 若猛狼 i 在 k 次前进后的性能指标值 $Y_i(k)$ 优于当前头狼, 则更新头狼及其性能指标值 $Y_{\text{lead}} = Y_i(k)$, 由猛狼 i 取代头狼重新发起召唤。否则, 猛狼 i 继续向头狼奔袭, 直到其与头狼之间的距离小于 d_{near} 。当所有 M_{num} 匹猛狼与头狼之间的距离均小于 d_{near} 时, 进入围攻行为。其中判定距离 d_{near} 可由式(14)计算

$$d_{\text{near}} = \frac{1}{D\varpi} \sum_{d=1}^D (ub_d - lb_d) \quad (14)$$

式中: ϖ 为距离判定因子; $[lb_d, ub_d]$ 为第 d 个决策变量的取值范围; D 为决策变量的数量。

2.1.3 围攻行为

猛狼经过奔袭过程后, 其与头狼的位置非常接近, 将头狼所在位置视为猎物所在位置, 此时猛狼与探狼一起以 step_c 的攻击步长围捕猎物, 狼 i 第 $k+1$ 次围捕前进后的位置由式(15)计算得出

$$x_i(k+1) = x_i(k) + (2 \times \lambda - 1) \text{step}_c \cdot \frac{|G_k - x_i(k)|}{|G_k - x_i(k)|} \quad (15)$$

式中: λ 为 $[0, 1]$ 区间均匀分布的随机数; G_k 为猎

物所在位置。若狼 i 在前进后的性能指标值优于当前头狼, 则将狼 i 作为新的头狼, 并更新头狼位置(也即猎物位置), 然后继续下一匹狼的围捕过程; 否则继续前进, 直至与猎物的距离达到某一阈值。

上述3种智能行为中的前进步长之间的关系一般可表示为

$$\text{step}_a = \frac{\text{step}_b}{2} = 2\text{step}_c = \frac{|ub - lb|}{s} \quad (16)$$

式中: s 为步长因子。

2.2 改进狼群算法设计

标准狼群算法由于其迭代层数较多, 故对某些问题的求解效率不高。目前, 已有一些研究针对其不足进行了改进^[13-17], 例如在其中加入自适应学习和领导者策略等。在将狼群算法应用于单元调度时, 容易出现算法陷入局部极值和收敛速度慢等不足。针对这些问题, 本文提出了一种混合狼群算法, 采用将细菌觅食算法(BFO)中用于改进细菌探索能力的趋向行为和柯西变异引入狼群算法^[19], 并采用自适应步长策略^[20], 以增强探索能力、加快收敛速度, 并避免陷入局部极值。考虑到经典狼群算法中用头狼位置表示猎物位置, 所得到的解尚有较大改进空间, 本文对于头狼位置加以高斯微小扰动, 从而确定猎物位置, 使得算法寻优能力得以提高。此外, 针对单元调度问题, 本文采用了一种三层编码形式, 可以有效表达单元、机器、及作业顺序的关系。算法通过离散事件仿真过程获取目标函数值, 并以此作为狼群个体的性能指标评估。简记以上混合算法为 WPA-BFO-SIM。

2.3 编码和解码

本文采用3层编码方式^[2]对每个工件的每个操作所对应的机器和单元进行编码, 第1层中的值表示操作 O_{ij} 所对应的工件号 j , 第2层表示操作 O_{ij} 所在的机器, 第3层代表操作 O_{ij} 所在的单元。例如, 图1所示编码表示对工件4的加工包含2项操

作, 第1项操作由位于单元2的机器5完成, 第2项操作由位于单元1的机器2完成; 在机器5上, 工件的加工顺序为4-1-5。

操作 O_{ij}	4	3	4	6	1	5
机器 k	5	3	2	1	5	5
单元 u	2	1	1	1	2	2

图1 3层个体编码

Fig. 1 Three-layer individual coding

2.4 初始化

首先生成个体的第1层, 随机生成工件序列(工件 j 出现的次数为其完成整个加工的操作次数); 其次, 对每个操作找到相应的可用机器, 得出个体第2层; 最后, 找出机器所在的单元号作为第3层。

2.5 自适应步长

为了加快算法的收敛速度, 本文采用了文献[20]提出的自适应步长, 使每匹狼的前进步长可以依据当前头狼的位置 X_{lead} 和自身的位置 x_i 进行动态调整, 其定义为

$$step_a = rand \cdot \|x_i - X_{lead}\|_2 \quad (17)$$

式中: $rand$ 表示[0, 1]之间的随机数。

采用自适应步长后, 式(13), (14)和(15)可分别表示为

$$x_i^p = x_i + \sin(2\pi \frac{p}{h}) \cdot step_a \cdot rand \|x_i - X_{lead}\|_2 \quad (18)$$

$$x_i(k+1) = x_i(k) + \frac{(X_{lead} - x_i(k))}{\|X_{lead} - x_i(k)\|} \cdot step_b \cdot rand \|x_i - X_{lead}\|_2 \quad (19)$$

$$x_i(k+1) = x_i(k) + (2\lambda - 1) \cdot step_c \cdot |G_k - x_i(k)| \cdot rand \cdot \|x_i - X_{lead}\|_2 \quad (20)$$

2.6 趋向行为和柯西变异策略

通过采用上述自适应步长, 可以显著提高狼群

算法的收敛速度, 但在游走行为中, 当多个方向存在性能改进时, 经典狼群算法仅沿最大改进方向前进, 之后又重复上述行为, 导致搜索效率较低。针对这一问题, 本文采用将BFO中的趋向行为加入到游走过程中, 当探狼 i 向性能改进最好的方向前进进一步后, 将会沿着相同方向继续前进几步(前进的步数用 N_s 表示), 直到目标函数不再改进或达到最大前进移动步数 N_{max} 。因此, 探狼 i 继续前进每一步后所处的位置可通过式(21)计算得到

$$x_{i,next} = x_i + \sin(2\pi \frac{p}{h}) \cdot rand \cdot \|x_i - X_{lead}\|_2 \cdot \varphi(i) \quad (21)$$

式中: $rand$ 为[0, 1]之间的随机数; $\varphi(i)$ 为性能指标最佳改进的方向, 通过式(22)计算

$$\varphi(i) = \frac{(x_j - x_i)}{|x_j - x_i|} \quad (22)$$

式中: x_i 为当前最优性能个体的位置。

此外, 为了避免游走行为陷入局部极值, 本文还在游走过程中引入了柯西变异策略, 也即, 当探狼 i 沿 h 个方向的探测均未发现改进时, 将以一定的概率 P_c 变异(更新位置); 否则, 探狼 i 的位置保持不变。本文采用(23)式的柯西变异公式对当前最优个体进行位置更新。

$$x_{next}^* (t) = x_i^* (t) \cdot (1 + cauchy(0, 1)) \quad (23)$$

式中: $x_{next}^* (t)$ 为经柯西扰动后得到的位置; $cauchy(0, 1)$ 为柯西算子。

图2给出了改进后的游走行为伪代码(最小化目标函数)。

2.7 高斯扰动

传统狼群算法在对在围捕阶段将头狼位置作为猎物位置, 使得算法探索最优值的能力受到局限。本文汲取鲸群算法^[21]思想, 在围捕阶段以一定概率 P_{ga} 对头狼进行高斯微小扰动, 以确定猎物位置, 使得算法探索最优值的能力得以提高。本文中, 通过式(24), (25)对猎物位置进行扰动。

```

for t=1: Tmax do
  for i=1: Snum do
    for j=1: h do
      根据公式(18)生成个体;
      计算该个体的目标函数值;
    end
    确定当前最优个体X1;
    Y1=f(X1);
    if Y1<Ylead then
      Xlead=X1;
      Ylead=Y1;
      while NS<NSMAX do
        根据公式(21)生成个体X2;
        Y2=f(X2);
        if Y2<Ylead then
          Xlead=X2;
          Ylead=Y2;
        else
          NS=NSMAX;
        end
      end
    elseif rand<PED then
      cauchy=tan((rand-1/2)*pi);
      根据公式(23)生成新个体
    end
    找出最优个体X3;
    Y3=f(X3);
    if Y3<Ylead then
      Xlead=X2;
      Ylead=Y3;
      flag=1;
      break;
    end
  end
end
if flag = 1
  break;
end
end
end

```

图2 改进的游走行为伪代码
Fig. 2 Pseudo code for improved scouting behavior

$$NG_{\text{best}}^t = G^t (1 + \text{Gaussian}(\mu, \sigma)) \quad (24)$$

$$G^{t+1} = \begin{cases} NG_{\text{best}}^t, & f(NG_{\text{best}}^t) < f(G^t) \\ G^t, & \text{其他} \end{cases} \quad (25)$$

式中: NG_{best}^t 为当前迭代次数下通过高斯微小扰动得到的猎物位置; G^t 为当前迭代次数下的头狼位置。

2.8 个体修复

由于不同的行为与变异策略可能导致式(5)不满足, 在这种情况下, 必须实施修复过程。如果一个零件由于工序顺序改变从而不可行, 必须重

新生成零件工序的新排列, 且需要重新分配相应的机器和单元, 具体过程如下。

首先, 随机生成不满足约束部分工序的排列, 使其满足约束; 其次, 遍历第1层中的每个工序, 选择处理时间最短的机器; 然后, 为每个工序分配单元。该工序取决于同一零件的前一个工序的加工单元, 如果该单元中存在可用的机器, 则将当前工序分配给该机器; 否则, 在有可用机器的地方随机选择一个单元。

2.9 WPA-BFO-SIM 算法结构

综上, 本文的改进狼群算法 WPA-BFO-SIM 流程如图3所示。

3 仿真实验与结果分析

3.1 实验样本与参数

仿真实验环境为2 GHz Inter® Core™i7 CPU 和 8 GB RAM, 采用 MATLAB 编程。10个样本实例数据如表1所示。加工时间在区间[1,20]之间随机生成; 机器依完成的操作数发生中断, 假设完成2个操作后发生中断, 平均中断(维修)时间为3分钟; 机器空转时的单位能耗成本为1, 单位维修成本为1。

改进狼群算法的参数设置如表2所示。

为了确定目标权重 W_1 和 W_2 的取值, 本文使用算例10, 测试了目标权重 W_1 和 W_2 分别取从0.1~0.9时2个目标函数的取值, 得出了非劣解前沿, 如图4所示。

图4显示, 当目标权重 W_1 和 W_2 的取值为0.7和0.3时, 2个目标值能够达到很好的权衡效果。故本文在后续试验中, 选取 W_1 和 W_2 分别为0.7和0.3。

3.2 实验设计

本文问题主要考虑能源消耗与机器中断的单元调度问题, 因此设计了5组仿真实验, 如表3所示。

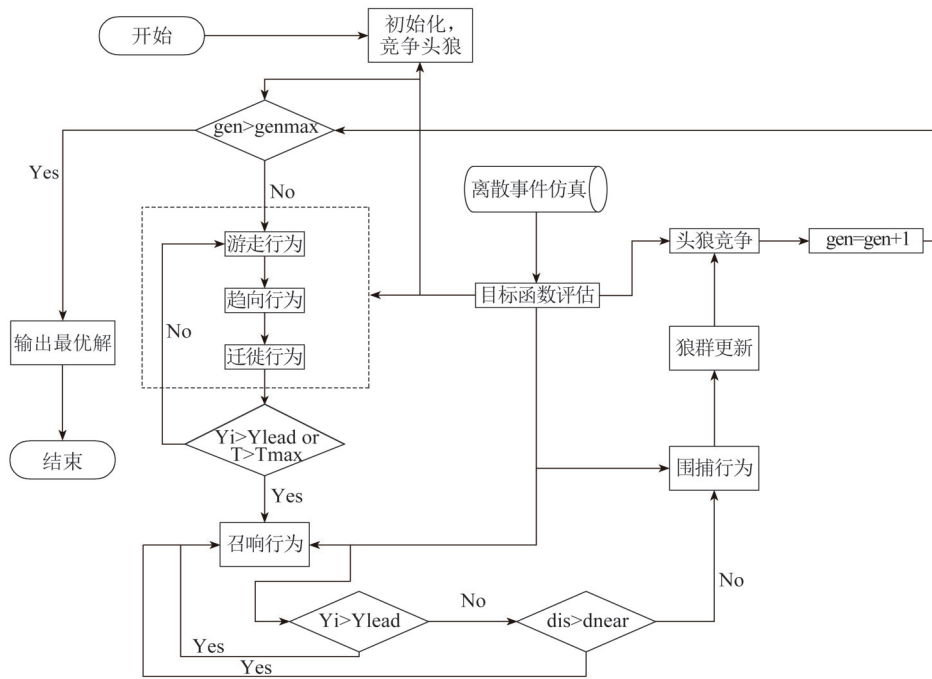


图 3 WPA-BFO-SIM 流程图
Fig. 3 Flow chart of WPA-BFO-SIM

表 1 样本问题数据表

Table 1 Specifications of sample problems

样本编号	单元数	工件数	机器数	工序数
1	2	4	4	4
2	2	4	5	5
3	2	4	6	6
4	3	5	8	6
5	3	6	10	8
6	3	10	15	10
7	4	15	20	15
8	4	20	25	20
9	4	25	30	25
10	5	31	35	30

表 2 WPA-BFO-SIM 参数设置

Table 2 Parameter values of WPA-BFO-SIM

WPA-BFO-SIM 参数	值
狼群数量	100
最大迭代次数	80
最大游走次数	30
探狼比例因子	4
距离判定因子	800
步长因子	1000
淘汰因子	10
搜寻方式上界	6
搜寻方式下界	3
柯西变异概率 P_c	0.5
高斯扰动概率 P_{ga}	0.5

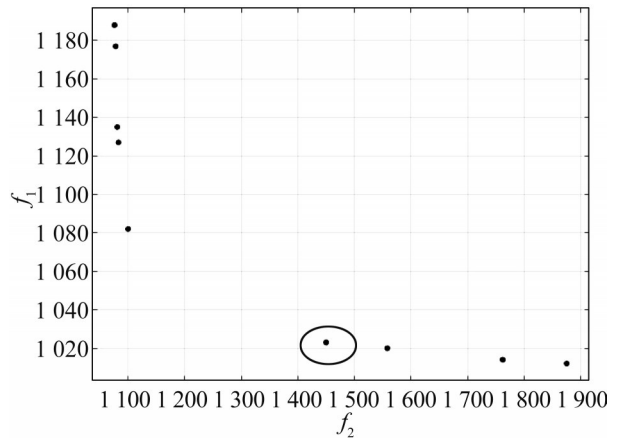


图 4 2 个目标函数值的非劣解前沿
Fig. 4 Non-inferior frontier of two objective function

表 3 5 组仿真实验表

Table 3 5 groups of simulation experiment

方案	优化算法
1	基因遗传算法
2	蚁群算法
3	标准狼群算法
4	两段式狼群算法(TWPA)
5	改进的狼群算法(WPA-BFO-SIM)

为提高狼群算法的搜索能力,在WPA框架下改进局部搜索策略从而构建混合算法是一种常见思路。但不同的混合策略性能差异较大。本文主要采用了细菌觅食(BFO)过程实现局部搜索,为检验改进效果,本节将文中提出的WPA-BFO-SIM方法与标准狼群算法(WPA)以及文献[22]提出求解柔性作业车间调度问题的两段式狼群算法(TWPA)进行了比较。此外,为了进一步验证本文方法的性能,还与常用的遗传算法(GA)、蚁群算法(ACO)也进行了比较。由于本文特别关注单元调度中的能源消耗与机器中断问题,因而在对算法进行分析比较时,除了比较加权综合的目标函数值,还采用了多个不同的具体评价指标,具体如下。

(1) 机器平均利用率改进量

该指标为优化前后机器平均利用率的绝对改进量。机器平均利用率的计算如式(26)所示。

绝对改进量通过式(27)计算。

$$\bar{R} = \frac{\sum_{k \in M} S_k}{N} \times 100\% \quad (26)$$

式中: S_k 为第 k 台机器的加工时间; H_k 为机器 k 的完工时间; N 为最大仿真次数。

$$\Delta \bar{R} = \bar{R}_L - \bar{R}_B \quad (27)$$

(2) 最大完工时间降低百分比

优化前后最大完工时间降低幅度百分比通过式(28)计算得到。

$$\Delta W = \frac{W_B - W_L}{W_B} \times 100\% \quad (28)$$

式中: W_B 为优化前最大完工时间; W_L 为优化后最大完工时间。

(3) 空闲时间能耗降低百分比

优化前后空闲时间能耗降低幅度百分比通过式(29)计算得到。

$$\Delta E = \frac{E_B - E_L}{E_B} \times 100\% \quad (29)$$

式中: E_B 为优化前空闲时间能耗; E_L 为优化后空闲时间能耗。

3.3 仿真实验结果分析

为了分析本文提出的WPA-BFO-SIM算法的优化效果,以算例10为例,对比随机调度结果和采用本文提出的WPA-BFO-SIM算法优化调度结果。调度结果分别如图5,6所示。

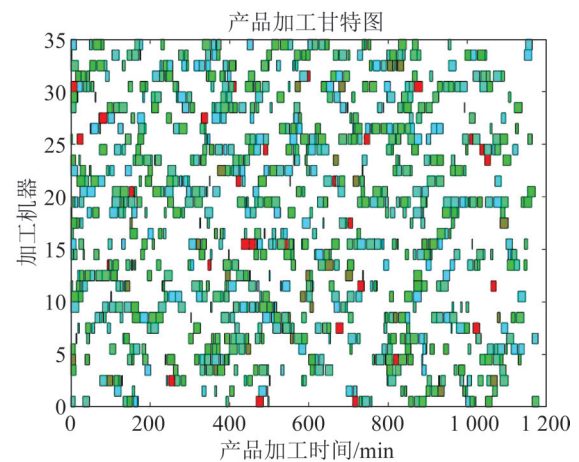


图5 随机调度得到的调度方案

Fig.5 Scheduling plan of the problem obtained by random scheduling

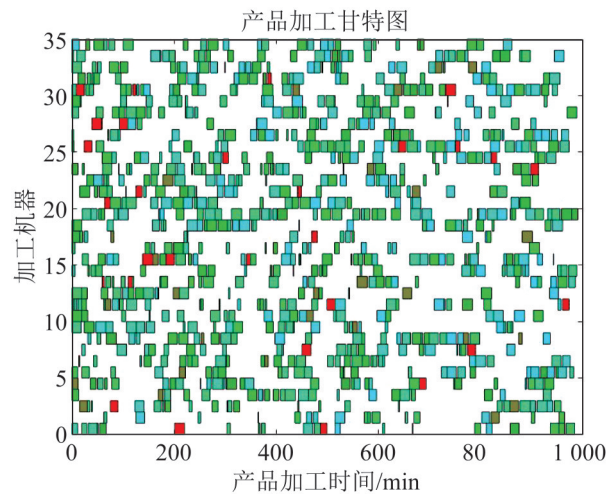


图6 WPA-BFO-SIM求解得到的调度方案

Fig.6 Scheduling plan of the problem using WPA-BFO-SIM

图5显示,未经优化的调度安排稀疏,最大完工时间为1181min,并且空闲的能源消耗高达8773,浪费了大量的能源。而图6表明,在经过本文WPA-BFO-SIM算法优化后得到了满意的结果,其最大完工时间降低为990min,降低幅度为

16.2%，空闲能源消耗下降为5 119，下降幅度为41.7%，可以极大的减少能源消耗。由此可见，本模型可以大幅降低机器空转而导致的能源浪费，极大的节约了能源的使用。

为了验证本文提出的混合算法WPA-BFO-SIM在求解单元调度问题时的寻优能力，将5种优化算法的求解结果进行了比较。对比结果如表4所示。

表4 5种优化算法的比较
Table 4 Comparison of 5 optimization algorithms

算例	GA		ACO		WPA		TWPA		WPA-BFO-SIM	
	时间/s	目标函数值	时间/s	目标函数值	时间/s	目标函数值	时间/s	目标函数值	时间/s	目标函数值
1	2.746	51.5	2.348	51.5	1.447	51.5	1.041	51.5	0.665	51.5
2	4.272	95.1	3.363	95.1	3.623	95.1	1.847	95.1	1.164	95.1
3	5.167	129.2	4.396	129.2	4.818	129.2	3.071	127.4	2.598	127.4
4	5.824	173.3	4.527	173.3	5.157	170.3	8.626	171.5	4.735	170.3
5	15.154	287.2	14.259	289.5	11.654	287	13.966	277	9.829	276.7
6	19.496	436.2	20.692	437.5	21.556	469.2	30.182	418.4	19.410	401.2
7	35.777	989.7	66.013	962.8	28.063	924.4	39.110	824.8	26.906	776.4
8	77.822	1 502	75.199	1 529.1	78.671	1 579.2	73.521	1 474.8	76.098	1 378.9
9	113.453	2 335	130.412	2 440	96.107	2 390.4	85.419	2 031.7	77.202	2 020.6
10	215.747	2 977.8	250.231	3 891.1	269.663	2 951.1	353.439	2 460.1	204.960	2 301.7

表4显示，WPA-BFO-SIM无论在求解性能还是效率方面均优于GA，ACO，WPA，TWPA。通过对比WPA-BFO-SIM与TWPA可以发现，本文引入BFO的迁徙行为在计算效率与求解性能两个维度上都要优于在WPA中引入局部搜索的两段式狼群算法情况，尤其是问题规模较大时。

为了进一步分析WPA-BFO-SIM算法对于考虑机器中断与能源消耗的单元调度问题的优化效果，以下使用所构建的3个评价指标对上述10个

算例的求解结果进行不同维度的分析，如表5所示。对于小规模问题(算例1, 2, 3)来说，5个优化算法的结果相近；对于中等规模问题(算例6)来说，WPA-BFO-SIM对于机器平均利用率提高了4.9%，最大完工时间降低了14.8%，空闲时间能耗降低了24.8%，均优于GA，ACO，WPA，TWPA；对于大规模问题(算例10)，WPA-BFO-SIM的优势更加明显，其3个指标都远超前于GA，ACO，WPA，TWPA。

表5 5种优化算法在不同数据规模下的优化比较
Table 5 Optimization comparison of 5 optimization algorithms under different data scales %

算例	GA			ACO			WPA			HWPA			WPA-BFO-SIM		
	$\Delta\bar{R}$	ΔW	ΔE	$\Delta\bar{R}$	ΔW	ΔE	$\Delta\bar{R}$	ΔW	ΔE	$\Delta\bar{R}$	ΔW	ΔE	$\Delta\bar{R}$	ΔW	ΔE
1	4.8	4.4	41.2	4.8	4.4	41.2	4.8	4.4	41.2	4.8	4.4	41.2	4.8	4.4	41.2
2	2.8	17.1	25	2.8	17.1	25	2.8	17.1	25	2.8	17.1	25	2.8	17.1	25
3	6.2	10.4	23.1	6.2	10.4	23.1	5.2	10.4	20.9	8	10.4	26.7	8	10.4	26.7
4	3.3	7.1	13.5	3.3	7.1	13.5	5.3	7.1	17.4	5.8	9.8	20.6	7	9.8	23
5	0.5	6.4	5.5	0.7	7.6	6.2	1.5	9.6	12.6	2	11.5	15.4	2.8	11.5	17
6	0.3	2	2.8	0.6	5.9	4.7	1	3.9	6.3	1.3	13	21.6	4.9	14.8	24.8
7	1.7	9.1	13.6	3.8	15.2	24.8	3.1	9.8	18.1	3.5	23.3	31.2	3.1	24.8	32.1
8	0.8	9.1	11.6	5.1	10.2	25.5	6.7	24.6	3	5.3	16	30	5.6	24.6	33.2
9	0.6	7	4.7	4.3	9.2	24	1.4	6.7	9.7	5	10.7	29	5.1	13.5	31.5
10	3	4.2	17.3	4.5	6.3	19.9	1.2	8.3	13.9	6	10.2	34.6	6.5	16.2	41.7

以上实验结果表明, WPA-BFO-SIM算法对于求解考虑机器中断与能源消耗的单元调度问题具有更好的优化效果。

4 结论

本文针对考虑机器中断和能耗的单元调度问题, 建立了一个整数规划模型, 以最小化最大完成时间和最小化中断成本与机器空转耗能成本为目标。进而提出了一种混合求解算法, 以狼群算法作为主要优化引擎, 采用在其中引入细菌觅食搜索算子以增强其寻优能力, 同时使用柯西变异进行改进, 并采用了自适应步长以加快收敛速度, 最后借助鲸群算法思想对头狼进行高斯微小扰动, 以确定猎物位置, 使得算法探索最优值的能力方面得以显著提高。考虑到解码过程的复杂性以及今后在模型随机性方面的拓展, 算法采用了离散事件仿真过程来求取目标函数值。仿真实验表明, 本文方法对于单元制造问题能够提供一种有效调度方案, 通过确定各工件的每道工序在不同单元和机器上的分配, 最大限度地减少了系统中机器的空转耗能、中断成本, 同时优化了所有工件的完工时间。仿真实验表明, 本文的混合狼群算法在所有算例上均取得了性能最好的结果。与其他4种算法相比, 当问题规模较小时, 各方法求解性能相近, 但WPA-BFO-SIM具有更高的计算效率; 而当问题规模较大时, WPA-BFO-SIM在寻优能力和运行效率两方面均有显著优势。

未来的研究可以在本文研究基础上考虑在单元内部的布局与人员在单元间的分配问题等。

参考文献

- [1] Wu X, Chu C H, Wang Y, et al. A Genetic Algorithm for Cellular Manufacturing Design and Layout[J]. *European Journal of Operational Research*, 2007, 181:156-167.
- [2] Feng Y, Li G, Sethi S P S. A Three-layer Chromosome Genetic Algorithm for Multi-cell Scheduling with Flexible Routes and Machine Sharing[J]. *International Journal of Production Economics*, 2018, 196:269-283.
- [3] Sinaki R Y, Sadeghi A, Suer G, et al. A Weighted Multi-Objective Mathematical Model for Cell Scheduling and Environmentally Sustainable Supply Chain Network[J]. *Procedia Manufacturing*, 2019, 39:1559-1566.
- [4] International Energy Agency (IEA), International Energy Outlook 2017[R]. [2017-09-14] (2021-03-21) https://www.eia.gov/pressroom/presentations/mead_91417.pdf.
- [5] Iqbal N, Aziz M H, Jahanzaib M, et al. Integration of Cell Formation and Job Sequencing to Minimize Energy Consumption with Minimum Make-span[J]. *Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture(S0954-4054)*, 2016, 231(14): 2636-2651.
- [6] Liu Y, Dong H, Lohse N, et al. A multi-objective Genetic Algorithm for Optimization of Energy Consumption and Shop Floor Production Performance[J]. *International Journal of Production Economics(S0925-5273)*, 2016, 179: 259-272.
- [7] Gahm C, Denz F, Dirr M, et al. Energy-efficient Scheduling in Manufacturing Companies: A Review and Research Framework[J]. *European Journal of Operational Research(S0377-2217)*, 2016, 248(3): 744-757.
- [8] Yang Y, Huang M, Wang Z Y, et al. Robust Scheduling based on Extreme Learning Machine for Bi-objective Flexible Job-shop Problems with Machine Breakdowns [J]. *Expert Systems with Applications(S0957-4174)*, 2020, 158: 113545.
- [9] Tadayonirad S, Seidgar H, Fazlollahtabar H, et al. Robust Scheduling in Two-stage Assembly Flow Shop Problem with Random Machine Breakdowns: Integrated Meta-heuristic Algorithms and Simulation Approach[J]. *Assembly Automation(S0144-5154)*, 2019, 39(5): 944-962.
- [10] Li Y, He Y, Wang Y, et al. An Optimization Method for Energy-conscious Production in Flexible Machining Job Shops with Dynamic Job Arrivals and Machine Breakdowns[J]. *Journal of Cleaner Production(S0959-6526)*, 2020, 254:120009.
- [11] Sarker R, Omar M, Hasan K S M, et al. Hybrid Evolutionary Algorithm for Job Scheduling under Machine Maintenance[J]. *Applied Soft Computing (S1568-4946)*, 2013, 13(3):1440-1447.
- [12] 吴虎胜, 张凤鸣, 吴庐山. 一种新的群体智能算法—狼群算法[J]. *系统工程与电子技术*, 2013, 35(11): 2430-2438.
Wu Husheng, Zhang Fengming, Wu Lushan. New Swarm Intelligence Algorithm—wolf Pack Algorithm [J]. *Systems Engineering and Electronics*, 2013, 35(11): 2430-2438.

- [13] Wu H S, Xue J J, Xiao R B, et al. Uncertain Bilevel Knapsack Problem based on an Improved Binary Wolf Pack Algorithm[J]. *Frontiers of Information Technology & Electronic Engineering*(S2095-9184), 2020, 21(9): 1356-1368.
- [14] Xiu-Wu Y U, Hao Y U, Yong L, et al. A Clustering Routing Algorithm based on Wolf Pack Algorithm for Heterogeneous Wireless Sensor Networks[J]. *Computer networks*(S1389-1286), 2020, 167(Feb. 11): 106994.1-106994.10.
- [15] Hu J, Wu H, Zhan R, et al. Hybrid Integer-coded Wolf Pack Algorithm for Multiple-Type Flatcars Loading Problem[J]. *Journal of Rail Transport Planning & Management*(S2210-9706), 2020: 100201.
- [16] Chen Y, Jia Z, Ai X, et al. A Modified Two-part Wolf Pack Search Algorithm for the Multiple Traveling Salesmen Problem[J]. *Applied Soft Computing*(S1568-4946), 2017, 61:714-725.
- [17] Chen X Y, Tang C J, Wang J, et al. Improved Wolf Pack Algorithm Based on Differential Evolution Elite Set[J]. *IEICE Transactions on Information and Systems*(S1745-1361), 2018, E101D (7): 1946-1949.
- [18] Passino K M. Biomimicry of Bacterial Foraging for Distributed Optimization and Control[J]. *IEEE Control Systems Magazine*(S1066-033X), 2002, 22(3): 52-67.
- [19] 李堂金. 基于狼群算法的模糊时间序列预测模型研究[D]. 重庆: 重庆邮电大学, 2020.
Li Tangjin. Research on the Fuzzy Time Series Forecasting Model Based on Wolf Pack Algorithm[D]. Chongqing: Chongqing University of Posts and Telecommunications, 2020.
- [20] 郭立婷. 基于自适应和变游走方向的改进狼群算法[J]. *浙江大学学报(理学版)*, 2018, 45(3): 284-293.
Guo Liting. Improved Wolf Pack Algorithm based on Adaptive Step Length and Adjustable Scouting Direction [J]. *Journal of Zhejiang University (Science Edition)*, 2018, 45(3): 284-293.
- [21] Mirjalili S, Lewis A 2016. The Whale Optimization Algorithm[J]. *Advances in Engineering Software*(S0965-9978), 95(5), 51-67.
- [22] 谢锐强, 张惠珍. 求解柔性作业车间调度问题的两段式狼群算法[J]. *计算机工程与应用*, 2021, 57(7): 251-256.
Xie Ruiqiang, Zhang Huizhen. Two-Vector Wolf Pack Algorithm for Flexible Job Shop Scheduling Problem[J]. *Computer Engineering and Applications*, 2021, 57(7): 251-256.