

6-2-2020

Improved Harmony Search Algorithm in Application of Vulcanization Workshop Scheduling

Longcheng Wu

Engineering Research Center of IoT Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Huang Song

Engineering Research Center of IoT Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Wang Yan

Engineering Research Center of IoT Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;

Zhicheng Ji

Engineering Research Center of IoT Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, 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 Paper 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.

Improved Harmony Search Algorithm in Application of Vulcanization Workshop Scheduling

Abstract

Abstract: For solving the problem of low productivity and low machine utilization caused by unreasonable assignment in tire vulcanizing process, an improved Discrete Harmony Search (Improved Harmony Search, IHS) Algorithm was proposed to optimize the maximum completion time of vulcanizing workshop scheduling. *This algorithm has three processes including individual memory consideration process, disturbance search process and competitive selection mechanism. In the competitive selection mechanism, the corresponding individual in harmony memory or worst solution was replaced with new solution and harmony memory was updated. That strategy made each individual in memory more quickly fly to the optimal solution.* The algorithm was simulated on MATLAB. The results show that IHS is superior to HS and PSO algorithm. This algorithm can efficiently solve the vulcanizing shop scheduling problem, and the analysis demonstrates the validity and feasibility of the algorithm.

Keywords

Discrete optimization algorithm, Harmony search algorithm, Harmony memory, Vulcanization workshop scheduling

Recommended Citation

Wu Longcheng, Huang Song, Wang Yan, Ji Zhicheng. Improved Harmony Search Algorithm in Application of Vulcanization Workshop Scheduling[J]. Journal of System Simulation, 2017, 29(3): 630-638.

改进和声搜索算法在硫化车间调度中的应用

吴龙成, 黄松, 王艳, 纪志成

(江南大学物联网技术应用教育部工程研究中心, 无锡 214122)

摘要: 针对硫化车间生产调度环节分配不合理导致的机台利用率不高和生产效率低的问题, 提出了一个改进的离散和声搜索(Improved Harmony Search, IHS)算法来优化硫化车间调度的最大完工时间。该算法包含个体记忆考虑过程、扰动搜索过程和竞争淘汰机制。在竞争淘汰机制中, 用新解替换和声记忆库中对应的个体或最差解, 更新和声记忆库, 加速优胜劣汰的进程, 使记忆库中每个个体更快速地向最优解方向靠近。通过 MATLAB 实验仿真, 结果表明, IHS 算法获得的结果要优于 HS 和 PSO 算法, 在解决硫化车间调度问题有良好的性能, 验证了该算法的有效性和可行性。

关键词: 离散优化算法; 和声搜索算法; 和声记忆库; 硫化车间调度

中图分类号: TP391.9 文献标识码: A 文章编号: 1004-731X (2017) 03-0630-09

DOI: 10.16182/j.issn1004731x.joss.201703022

Improved Harmony Search Algorithm in Application of Vulcanization Workshop Scheduling

Wu Longcheng, Huang Song, Wang Yan, Ji Zhicheng

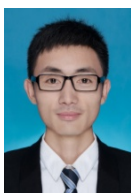
(Engineering Research Center of IoT Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China)

Abstract: For solving the problem of low productivity and low machine utilization caused by unreasonable assignment in tire vulcanizing process, an improved Discrete Harmony Search (Improved Harmony Search, IHS) Algorithm was proposed to optimize the maximum completion time of vulcanizing workshop scheduling. This algorithm has three processes including individual memory consideration process, disturbance search process and competitive selection mechanism. In the competitive selection mechanism, the corresponding individual in harmony memory or worst solution was replaced with new solution and harmony memory was updated. That strategy made each individual in memory more quickly fly to the optimal solution. The algorithm was simulated on MATLAB. The results show that IHS is superior to HS and PSO algorithm. This algorithm can efficiently solve the vulcanizing shop scheduling problem, and the analysis demonstrates the validity and feasibility of the algorithm.

Keywords: Discrete optimization algorithm; Harmony search algorithm; Harmony memory; Vulcanization workshop scheduling

引言

在轮胎生产过程中, 硫化车间的轮胎硫化工序



收稿日期: 2016-08-01 修回日期: 2016-09-09;
基金项目: 国家自然科学基金(61572238), 江苏省杰出青年基金(BK20160001);
作者简介: 吴龙成(1990-), 男, 湖北随州, 硕士, 研究方向为控制工程及应用; 黄松(1984-), 男, 湖北随州, 博士, 研究方向为智能优化算法、智能控制。

是轮胎加工的最后道工序, 也是整个轮胎制造过程中的瓶颈所在。目前, 我国轮胎企业的硫化车间普遍采用的是人工调度的方法, 管理人员根据自己的经验对生产订单进行调度安排, 这种方法做出的生产任务不够精确, 车间机台利用率不高。在竞争越来越激烈的今天, 对硫化车间进行优化调度, 提高机台的利用率和生产效率, 进而提高企业的竞争

力具有很重要的现实意义。生产调度优化的性能指标分为基于完成时间的指标、基于成本的指标、基于交货期的指标和基于设备负荷的指标。完工时间的长短是衡量一个优化调度策略好坏的最根本指标之一, 直接体现了车间的生产效率, 所以本文中主要是针对基于完成时间的指标来进行调度优化。

和声搜索 (Harmony Search, HS) 算法^[1-3]是 Geem 等人模仿乐师音乐创作, 达到一个美妙和声的过程而提出的一个启发式全局优化搜索算法。该算法近些年受到了广泛关注和研究, 并被运用到很多领域中。文献[4-5]中使用改进的 HS 算法解决 0-1 背包问题。文献[4]提出一种简化的二进制和声搜索算法并动态调整记忆库考虑概率。文献[5]根据适应度值自适应调节和声记忆考虑概率并动态调节基音调整概率。文献[6]采用模糊聚类的思想, 使和声记忆库考虑概率和微调概率随着记忆库中粒子变化而自适应变化, 并用来求解配电系统网络重构问题。文献[7]提出一种自适应和声搜索算法来求解结构工程设计问题, 算法加入了变异率并且使它在算法运行初期保持较小值, 到了后期增加为较大值来提高算法的收敛性和空间开发能力。文献[8]中提出混合的和声搜索算法并采用离散化编码来求解柔性作业车间调度问题。

虽然 HS 算法凭借自身优异性能在求解其他最优化问题时表现出色, 但是未使用在硫化车间调度问题中。本文对搭建的硫化车间调度模型采用离散编码和解码方式, 提出了离散和声搜索算法。基本的 HS 算法属于单个体更新操作, 在解决硫化车间调度问题时收敛效果不佳; 为了解决这个问题, 本文提出了 IHS 算法, 包括个体记忆考虑过程、扰动搜索过程和竞争淘汰机制 3 个过程。新解有 HMCR 的概率直接取 HM 内对应和声的值; 有 1-HMCR 的概率在 HM 外, 变量可能取值范围内随机取值; 若新解取自 HM 内, 再以微调概率 PAR 对其进行局部扰动。如果新解优于它对应的 HM 内和声, 替换这个和声, 否则替换掉差于新解的 HM 内最差和声。将硫化车间模型结合 IHS 算法来求

解以最大完工时间为优化目标的调度问题, 提高车间机台利用率和产品生产效率。

1 硫化车间调度模型

1.1 模型描述

硫化车间的生产调度问题是一种单工序、多产品、并行机器的批量调度问题, 对该类模型描述如下:

硫化车间配备有 m 台硫化机, 待加工轮胎的规格有 n 种, 每种规格型号的轮胎需要硫化的轮胎数量为 $N_j(j=1, 2, \dots, n)$, 硫化车间内 m 台硫化机每台机器都可以硫化这 n 种规格的轮胎。每台硫化机同一时刻只能硫化一条轮胎, 而且该轮胎只能在一台硫化机上完成硫化工作。每台硫化机机器的硫化能力不尽相同, 所以每台硫化机对各种规格的轮胎硫化时间也不完全相同。每台硫化机上加工不同规格的轮胎时需要更换模具, 更换模具的时间取决于进行切换的相邻两种轮胎的规格以及硫化机本身。订单中每种规格的轮胎需要加工的数量已知, 同时订单的优先级别仅取决于下达订单的时间先后。

已知条件为各种规格轮胎需要硫化的数量、各硫化机上更换轮胎规格所需要的换模时间和各种规格轮胎的硫化时间。因为硫化车间生产调度的安排是根据交货期限来制定而来, 所以调度的优化目标是在订单下达后产品生产周期最短, 即最大完工时间 $T(\text{makespan})$ 最小化。由上面的数学描述可知, 每台机器上的完工时间 T 包含两部分: 硫化时间 TP 和换模时间 TH 。假设第 j 台硫化机上的完工时间为 TE_j , 则优化目标可用公式(1)表示。

$$\min(\max(TE_j)) \quad (1)$$

约束条件:

$$TE_j = \sum TH_{j,p1,p2} + T_{j,p}, \quad p \in P \quad (2)$$

$$\sum_{j=1}^{B_j} B_{j,p} = A_p \quad (3)$$

$$TS_{j,p} = TH_{j,p1,p2} + TC_{j,p1} \quad (4)$$

$$TC_{j,p} = TS_{j,p} + T_{j,p} \quad (5)$$

$$\begin{aligned}
 p \in P\{1, 2, \dots, B_j\}, B_{j,p} \geq 0, A_p > 0, \\
 TS_{j,p} \geq TC_{j,p1}, TC_{j,p} > 0
 \end{aligned}
 \tag{6}$$

其中, j 为硫化机机台号; p 为待硫化的轮胎型号; B_j 为机器 j 上安排硫化的规格种数; A_p 为 p 型号轮胎的订单需求量; $T_{j,p}$ 为硫化机 j 硫化一条 p 型号轮胎所需要的时间; $TH_{j,p1,p2}$ 为硫化机 j 上硫化的轮胎型号由 p_1 切换到 p_2 的换模时间; $TS_{j,p}$ 为 p 型号轮胎在硫化机 j 上硫化的开始时间; $TC_{j,p}$ 为 p 型号轮胎在硫化机 j 上的完工时间; $B_{j,p}$ 为 p 型号轮胎在硫化机 j 上的硫化数量。

1.2 编码方式

对于求解硫化车间 m 台机器硫化 n 种规格轮胎的组合优化问题, 本文采用基于任务的编码方式, 即将每个轮胎看成一个任务。假设车间共有 4 台硫化机, 要硫化 1, 2, 3, 4 共 4 种规格的轮胎, 每种规格的轮胎需求量分别为 {2, 2, 2, 3}, 则可以确定编码长度为 $9(2+2+2+3)$, 具体编码方式见表 1。

表 1 编码策略表

Tab. 1 Encoding strategy table

X	1	2	3	4	5	6	7	8	9
Y	1	1	2	2	3	3	4	4	4
Z	4	2	1	4	3	2	2	3	1

在表 1 中, X 表示任务序号, Y 表示轮胎型号, 表示将每个轮胎看作一个任务, 各轮胎规格数量与订单中各规格轮胎数量一致。任务序号和轮胎型号保持不变, Z 表示机台号, 数值为随机生成 1~4 的整数, 表示对应型号的轮胎在该台机器上进行硫化。比如表 1 中任务序号为 6, 轮胎型号为 3, 机台号为 2 就表示第 6 个任务是由机器 2 硫化型号 3 的轮胎。

1.3 解码方式

当轮胎型号和任务序号的值固定后, 机台号 Y 随机生成, 对硫化任务进行分配, 进而可以明确每台硫化机上的硫化任务。解码时首先统计每台机器上硫化的轮胎规格数量, 以及每种规格轮胎所硫化的数量, 然后再根据已知的各机器上对应的轮胎硫

化时间和换模时间计算机器上的完工时间。如表 1 中, 可以看出在机器 2 上进行硫化的轮胎为 1, 3 和 4 三种型号, 硫化这三种型号轮胎的数量分别为 {1, 2, 1}。由于每台硫化机上硫化各种型号轮胎的时间以及更换各种型号轮胎之间的换模时间可知, 所以就可以得出在机器 2 上的完工时间为:

$$T = t(2,1) + t(2,3) + t(2,4) + h(2,1,3) + h(2,3,4)。$$

其中: $t(i, j)$ 表示在机器 i 上硫化 j 种型号轮胎所需时间; $h(i, j, k)$ 表示在机器 i 上硫化的轮胎型号由 j 换为 k 的换模时间。机器上的完工时间=各类型轮胎硫化数量×该型号轮胎硫化时间+换模时间。用同样的方法可以完成对其他机器的解码, 这样就可以知道对应 Y 这个调度结果硫化完订单中所有轮胎的完工时间了。

2 基本和声搜索算法

和声搜索(HS)算法是 Geem 等人在 2001 年提出的一种元启发式算法。和声搜索算法将优化问题的解向量类比于乐器声调的和声, 各对应的目标函数的值即是对音乐创作的评价。下面对连续和声搜索算法进行介绍。

和声记忆库初始化是通过随机生成 HMS 个和声 x^1, x^2, \dots, x^{HMS} 放在和声记忆库 HM 中, 和声记忆库的形式如下所示^[9]:

$$HM = \begin{bmatrix} x^1 & f(x^1) \\ x^2 & f(x^2) \\ \vdots & \vdots \\ x^{HMS} & f(x^{HMS}) \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_N^1 & f(x^1) \\ x_1^2 & x_2^2 & \dots & x_N^2 & f(x^2) \\ \vdots & \vdots & \dots & \vdots & \vdots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_N^{HMS} & f(x^{HMS}) \end{bmatrix}。$$

算法采用随机初始化记忆库的方式, 如式(7)所示。其中, rand 为 [0, 1] 之间的随机数, x_i^L 和 x_i^U 分别为第 $i(i=1, 2, \dots, N; N$ 为问题维数)维决策变量的下限和上限。

$$x_i = x_i^L + \text{rand}(x_i^U - x_i^L) \tag{7}$$

HS 在每次迭代中产生一个新解 $x^{new} = (x_1^{new}, x_2^{new}, \dots, x_N^{new})$ 。新和声中的每维变量 x_i^{new} 通过和声记忆库考虑概率(Harmony Memory Considering Rate, HMCR)判断新解中每一维变量是否在 HM 内取值, 如式(8)所示。其中, $u(1, HMS)$ 表示 [1, HMS] 上均匀分布的随机整数。

$$x_i^{new} = \begin{cases} x_i^{u(1, HMS)}, & \text{rand1} < \text{HMCR} \\ x_i^L + u(x_i^U - x_i^L), & \text{rand1} \geq \text{HMCR} \end{cases} \quad (8)$$

如果新解来自 HM 内, 通过基音调调整概率(Pitch Adjusting Rate, PAR)判断是否对新解进行微调, 通过独立的变量调节来产生新解, 如式(9)所示。如果新解优于 HM 内最优解, 则将最差解替换为新解^[10]。达到最大迭代次数算法停止运行。

$$x_i^{new} = \begin{cases} x_i^{new} + bw \times \text{rand2}, \text{rand2} < \text{PAR} \\ x_i^{new}, & \text{otherwise} \end{cases} \quad (9)$$

由上述可知 HS 算法有如下 3 点优势: (1) HS 算法产生的新解考虑了 HM 内现存所有的解; (2) HS 算法通过独立的变量调节对新解的每一维进行微调; (3) HS 算法的即兴创作通过随机数的变化进行核实记忆库考虑, 音调微调, 随机变异操作, 是一种全局随机搜索模式^[11]。

3 离散和声搜索算法和改进

3.1 离散和声搜索算法

硫化车间调度是离散问题, 需要使用离散的和声搜索算法来进行求解。离散 HS 算法首先对和声记忆库进行初始化, 每个决策变量的值域为, $X_i = (x_i(1), x_i(2), \dots, x_i(K))$ K 为离散型决策变量可能取值的个数。在离散 HS 算法每次迭代过程中根据记忆库考虑概率、随机选择和音调微调三种机理产生一个新解 $x^{new} = (x_1^{new}, x_2^{new}, \dots, x_N^{new})$ 。新解的产生方式如式(10)所示。

$$x_i^{new} = \begin{cases} x_i^{new} \in (x_i^1, x_i^2, \dots, x_i^{HMS}), \text{if } \text{rand} < \text{HMCR} \\ x_i^{new} \in X_i, & \text{otherwise} \end{cases} \quad (10)$$

对来自 HM 内的新和声 x_i^{new} 要进行微调, 如式(11)所示。

$$x_i^{new} = \begin{cases} x_i^{new}(k+m), m \in \{-1, 1\}, \text{if } \text{rand1} < \text{PAR} \\ x_i^{new}, & \text{otherwise} \end{cases} \quad (11)$$

如果产生的新和声优于 HM 内最差和声, 则将最差和声替换为新和声。当达到最大迭代次数时, 算法结束运行。

离散 HS 算法计算步骤如下:

Step 1: 定义离散 HS 算法参数: 和声记忆库的大小 HMS、音调微调概率 PAR、和声记忆库取值概率 HMCR 和最大迭代次数 M。

Step 2: 在决策变量可能取值的整数范围内随机生成 HMS 个和声, 并计算对应和声适应度值, 初始化和声记忆库。

Step 3: 按照式(10)产生一个新和声。根据式(11)对取自 HM 内的新和声进行音调微调。若新和声优于 HM 内最差和声, 替换最差和声, 更新 HM。

Step 4: 检查算法结束条件, 当前迭代次数大于最大迭代次数时结束算法运行, 否则返回到 Step2。

3.2 离散 HS 算法的改进

由离散和声搜索算法的计算步骤可知, 该算法运行过程中每次迭代只利用和声记忆库产生一个新和声, 新和声再跟 HM 中最差的和声作对比, 如果比最差和声好就替换最差和声, 以达到更新 HM 的目的。这种更新方式属于单个体更新操作, 不能充分利用 HM 内所积累的信息, 同时因为算法随机搜索的特性, 也使得产生的新解随着迭代次数的增加越来越难进入和声记忆库内, 和声搜索算法的局部搜索能力大打折扣。

(1) 个体记忆考虑过程

个体记忆考虑过程能充分利用 HM 内所积累的优势信息。由于 HS 属于维记忆考虑过程, 只针对和声的部分维进行记忆。在本文中, 个体记忆考虑过程则针对每个个体进行整体记忆考虑。部分维记忆的方式可能会破坏和声信息的完整性, 而个体记忆考虑过程不会破坏和声信息的完整性。

算法在每次迭代过程中针对现有和声记忆库

中的每个和声 $x^i(i=1, 2, \dots, \text{HMS})$ 产生一个对应的新和声 $x^{\text{new}} = (x_1^{\text{new}}, x_2^{\text{new}}, \dots, x_N^{\text{new}})$, 使用优胜劣汰的方法对原和声 x^i 以及 HM 内最差和声 x^{worst} 进行更新。为了使算法能够充分利用 HM 内所积累的信息, 也能适当从 HM 外随机取值, 避免算法陷入局部最优, 产生新解如式(12)所示。取 $[0, 1]$ 上均匀分布的随机数 u_1 , 如果 u_1 大于和声记忆库取值概率 HMCR, 新和声的每一维在决策变量值域 X_j 内随机取值, $j=1, 2, \dots, N$ 。如果 $u_1 < \text{HMCR}$, 直接将 HM 内原和声 x^i 的值赋给新和声 x^{new} 。假如针对 HM 中第 2 个和声产生一个新解, 新解有 HMCR 的概率直接取值 HM 中第二个和声, 即 $x^{\text{new}} = x^2$ 。同时为了提高算法全局搜索能力, 保持和声记忆库内种群的多样性, 新解有 $1 - \text{HMCR}$ 的概率来自值域 X_j 内的任意值, 即 $x_j^{\text{new}} \in X_j$ 。其中, $j=1, 2, \dots, N$ 。

$$x_j^{\text{new}} = \begin{cases} x_j^i, & \text{if } u_1 < \text{HMCR} \\ x_j^{\text{new}} \in X_j, & \text{if } u_1 \geq \text{HMCR} \end{cases} \quad (12)$$

(2) 扰动搜索过程

扰动搜索过程是一种局部搜索过程, 具有很强的局部搜索能力。它通过对音调进行微调, 在局部找到更好的和声, 从而跳出局部最优。

如果新解取自 HM 内, 根据音调微调概率 PAR 对 x^{new} 的每一维变量进行微调扰动搜索, 如式(13)和(14)所示。当 $u_3 < 0.5$ 时, k 取 -1, 否则取 1^[12], u_2 和 u_3 为 $[0, 1]$ 之间的随机数。

$$x_j^{\text{new}} = \begin{cases} x_j^{\text{new}} + k, & \text{if } u_2 < \text{PAR} \\ x_j^{\text{new}}, & \text{if } u_2 \geq \text{PAR} \end{cases} \quad (13)$$

$$k = \begin{cases} -1, & \text{if } u_3 < 0.5 \\ 1, & \text{if } u_3 \geq 0.5 \end{cases} \quad (14)$$

(3) 竞争淘汰机制

由于 HS 每次更新由新解和最差解之间的两两竞争进行更新, HM 内信息更新缓慢。在本文中, IHS 通过竞争淘汰机制对和声记忆库进行更新。在竞争淘汰机制中, 每个新解都和 HM 内对应的个体以及最差解进行竞争。竞争淘汰机制能加快 HM

内优胜劣汰的进程, 更快提高单个个体以及和声库的整体质量。

每次产生新和声要对该新解进行优势评估来更新和声记忆库。更新方式如式(15)所示。通过这种竞争淘汰机制加快 HM 整体向最优解靠近的速度。每次更新后重新寻找 HM 内最差和声 x^{worst} 。

$$\begin{cases} x^i = x^{\text{new}}, & \text{if } x^{\text{new}} < x^i \\ x^{\text{worst}} = x^{\text{new}}, & \text{if } x^i < x^{\text{new}} < x^{\text{worst}} \end{cases} \quad (15)$$

上述 IHS 算法流程如图 1 所示。

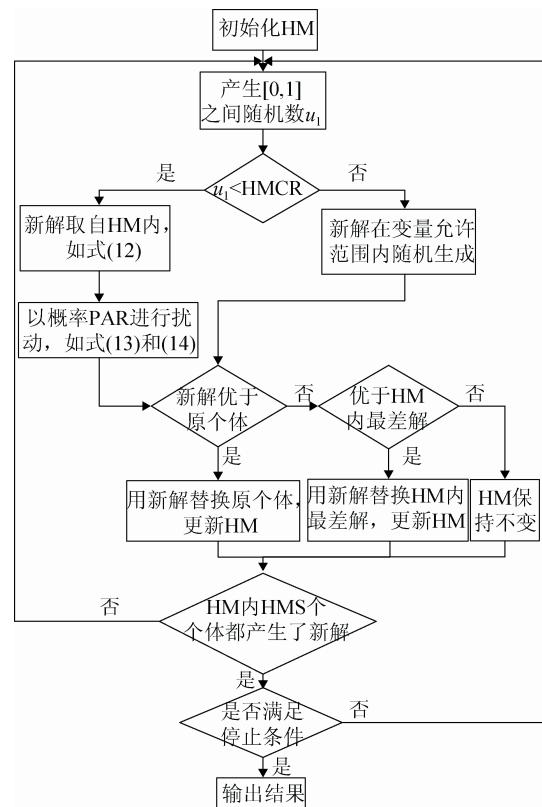


图 1 IHS 算法流程图
Fig. 1 Flow chart of IHS algorithm

4 仿真实验和结果分析

为了验证 IHS 算法在硫化车间调度中应用的有效性, 本文将 IHS 算法和 HS 算法、粒子群优化算法(PSO)算法进行对比。实验以某大型轮胎厂的 2 个硫化车间生产小组为对象, 对 3 台硫化机硫化 4 种型号轮胎、4 台硫化机硫化 3 种型号轮胎^[13]以及 7 台硫化机硫化 10 种型号轮胎 3 种情况进行仿真实验, 以最短加工周期为优化目标。

(1) 3×4 VWS(Vulcanization Workshop Scheduling) (3 台硫化机, 4 种型号的轮胎), 订单中各型号轮胎需要硫化的数量为[10, 15, 15, 10]。3 台硫化机分别用 M1, M2 和 M3 表示, 轮胎型号 225/40R17, 195/60R14, 185/70R13 和 165/70R13 分别用 J1, J2, J3 和 J4 表示。各型号轮胎在机器上硫化时间和各硫化机上硫化不同型号轮胎的换模时间具体参考文献[13]。

(2) 4×3 VWS(4 台硫化机, 3 种轮胎型号), 订单中各型号轮胎需求量为[1 000, 950, 900]。4 台硫化机用 M1, M2, M3 和 M4 表示, 3 种轮胎型号 185/70R13, 195/60R14 和 225/40R17 分别用 J1, J2 和 J3 表示。各型号轮胎在不同硫化机上的硫化时间和 3 种型号轮胎在 4 台机器上的轮胎换模时间具体参考文献[13]。

(3) 7×10 VWS(7 台硫化机, 10 种型号轮胎), 订单中对这 10 种型号轮胎需求量为[10, 15, 20, 20, 10, 15, 15, 25, 10, 10]。7 台硫化机分别用 M1, M2, M3, M4, M5, M6 和 M7 表示。10 种轮胎型号 235/65R17, 225/40R17, 205/45R17, 215/65R16, 205/65R16, 195/60R14, 175/70R14, 165/60R14, 185/70R13 和 165/70R13 分别用 J1, J2, J3, J4, J5, J6, J7, J8, J9 和 J10 表示。各型号轮胎在不同硫化机上的加工时间如表 2 所示。

表 2 7×10VWS 硫化时间表

Tab. 2 7×10 VWS vulcanization timetable

型号\机器	M1	M2	M3	M4	M5	M6	M7
J1	41	43	43	46	45	42	41
J2	35	37	42	43	39	35	40
J3	31	35	39	34	37	34	34
J4	30	31	33	34	34	33	27
J5	26	31	29	31	30	28	26
J6	23	30	25	28	30	24	23
J7	21	29	23	26	29	22	22
J8	19	28	20	25	25	22	21
J9	18	27	18	21	20	21	18
J10	17	18	16	17	18	18	16

第 i 台硫化机上第 j 种型号轮胎与第 j' 种型号轮胎之间的换模时间为 $Tr_{i+(i-1)*3,j'}$ 。其中, $Tr = [0, 1,$

$3, 2, 4, 2, 1, 2, 1, 3; 1, 0, 3, 2, 1, 4, 1, 4, 1, 2; 1, 3, 0, 2, 1, 3, 1, 4, 4, 3; 4, 1, 2, 0, 4, 2, 1, 4, 1, 3; 2, 1, 3, 1, 0, 3, 2, 2, 1, 4; 2, 1, 1, 2, 3, 0, 2, 4, 1, 1; 1, 4, 2, 4, 4, 2, 0, 3, 2, 2; 3, 4, 4, 2, 1, 2, 4, 0, 3, 1; 4, 1, 2, 3, 4, 2, 2, 1, 0, 3; 3, 2, 3, 1, 1, 4, 1, 2, 1, 0; 0, 3, 2, 1, 2, 1, 1, 2, 4, 3; 1, 0, 1, 2, 1, 3, 2, 4, 2, 3; 1, 2, 0, 3, 2, 2, 1, 1, 2, 1; 1, 1, 2, 0, 3, 1, 4, 3, 2, 1; 3, 1, 4, 2, 0, 3, 2, 2, 1, 1; 1, 4, 2, 4, 2, 0, 3, 1, 4, 2; 4, 3, 2, 2, 1, 1, 0, 3, 4, 3; 2, 3, 1, 4, 2, 2, 4, 0, 1, 3; 4, 1, 3, 2, 3, 4, 2, 2, 0, 1; 3, 3, 1, 2, 2, 1, 1, 4, 2, 0; 0, 2, 3, 4, 3, 2, 1, 1, 3, 3; 2, 0, 1, 1, 2, 3, 4, 2, 1, 3; 4, 2, 0, 2, 4, 3, 2, 1, 1, 3; 1, 3, 3, 0, 1, 3, 4, 2, 2, 3; 2, 1, 3, 3, 0, 3, 2, 4, 3, 1; 2, 3, 1, 1, 1, 0, 3, 1, 2, 2; 2, 4, 1, 3, 1, 4, 0, 1, 4, 1; 2, 1, 2, 1, 1, 3, 2, 0, 4, 1; 3, 1, 1, 3, 2, 1, 1, 3, 0, 2; 4, 1, 2, 4, 1, 1, 2, 1, 3, 0; 0, 2, 4, 4, 3, 2, 1, 1, 3, 2; 3, 0, 4, 1, 2, 3, 2, 1, 2, 3; 2, 1, 0, 1, 3, 4, 3, 4, 3, 2; 2, 4, 3, 0, 3, 2, 2, 3, 4, 1; 4, 2, 2, 4, 0, 1, 2, 3, 2, 1, 1, 3, 2; 3, 0, 4, 1, 2, 3, 2, 1, 1, 3, 2; 3, 0, 4, 1, 2, 3, 2, 1, 2, 3; 2, 1, 0, 1, 3, 4, 3, 4, 3, 2; 2, 4, 3, 0, 3, 2, 2, 3, 4, 1; 4, 2, 2, 1, 0, 2, 3, 1, 4, 1; 2, 4, 2, 3, 4, 0, 1, 2, 4, 2; 3, 1, 3, 3, 4, 3, 0, 2, 3, 1; 4, 2, 3, 4, 3, 1, 2, 0, 1, 2; 1, 4, 1, 3, 2, 4, 3, 3, 0, 1; 4, 2, 3, 2, 1, 4, 2, 3, 1, 0; 0, 2, 4, 1, 4, 1, 3, 2, 4, 1; 2, 0, 1, 2, 4, 3, 2, 1, 3, 1; 4, 1, 0, 1, 2, 1, 1, 2, 4, 3; 1, 2, 4, 0, 1, 3, 1, 1, 2, 4; 3, 3, 2, 3, 0, 1, 3, 3, 4, 1; 3, 2, 4, 1, 3, 0, 2, 1, 2, 1; 2, 1, 1, 3, 3, 2, 0, 2, 4, 1; 1, 3, 3, 2, 4, 3, 2, 0, 1, 1; 2, 2, 1, 4, 4, 3, 2, 2, 0, 1; 3, 3, 1, 4, 3, 4, 2, 3, 2, 0; 0, 1, 1, 1, 3, 3, 1, 4, 3, 2; 4, 0, 4, 2, 2, 1, 1, 2, 1, 3; 3, 1, 0, 3, 2, 4, 4, 3, 2, 1; 4, 3, 2, 0, 2, 3, 2, 1, 2, 1; 4, 2, 3, 4, 0, 2, 2, 2, 3, 2; 2, 1, 2, 2, 1, 0, 3, 1, 4, 3; 1, 3, 1, 1, 2, 2, 0, 2, 4, 2; 4, 3, 1, 4, 1, 2, 3, 0, 1, 3; 4, 4, 3, 2, 1, 1, 2, 2, 0, 3; 3, 4, 1, 2, 3, 2, 3, 3, 1, 0]。$

本文实验对 HS 算法、IHS 算法以及 PSO 算法 3 种优化算法进行仿真实验, 实验环境为: Interl Core i5-4590 CPU, 3.30 GHz 处理器, 4GB 内存, 采用 MATLAB R2010a 编程实现。HS 算法的和声记忆库的大小为 100, PSO 算法种群大小为 100,

每次实验 3 种算法的迭代次数都设为 500。

为了准确的对比分析这 3 种算法的收敛精度，本文在表 3~5 中列出了这 3 种算法在这 3 种硫化模式下进行的 20 次实验中优化结果的最大值，最小值，平均值和均方差。表 3~5 三种算法的种群大小分别为 30，100 和 150。其中的加粗值为这 3 种算法的最优结果。

表 3 三种算法优化 3×4VWS 对比结果(HMS=30)
Tab. 3 Comparison of three algorithms on 3×4 VWS (HMS=30)

例子	算法	最大值	最小值	平均值	均方差
3×4 VWS	PSO	466	453	459.95	3.0079
	HS	468	454	461.9	3.8846
	IHS	458	450	455.6	1.8815
4×3 VWS	PSO	369.9	367.6	368.385	0.58419
	HS	370.2	368.3	369.265	0.53691
	IHS	368.5	367.3	368.015	0.31824
7×10 VWS	PSO	689	661	677.1	8.0368
	HS	704	662	685.5	10.4857
	IHS	667	648	661	5.02

表 4 三种算法优化 4×3VWS 对比结果(HMS=100)
Tab. 4 Comparison of three algorithms on 4×3 VWS (HMS=100)

例子	算法	最大值	最小值	平均值	均方差
3×4 VWS	PSO	461	453	456.9	2.1656
	HS	468	454	461.2	3.2802
	IHS	456	448	452.7	1.9519
4×3 VWS	PSO	368.9	367.6	368.28	0.35157
	HS	369.7	367.7	368.96	0.59532
	IHS	368.6	366.7	367.795	0.29065
7×10 VWS	PSO	679	662	670.55	4.9947
	HS	695	662	679.75	8.1048
	IHS	667	642	658.15	4.6613

表 5 三种算法优化 7×10VWS 对比结果(HMS=150)
Tab. 5 Comparison of three algorithms on 7×10 VWS (HMS=150)

例子	算法	最大值	最小值	平均值	均方差
3×4 VWS	PSO	460	454	456.45	1.4992
	HS	468	454	461.55	3.8532
	IHS	455	447	452.65	1.621
4×3 VWS	PSO	369.3	367.4	368.365	0.48608
	HS	371.2	368	369.295	0.84704
	IHS	367.8	366.1	367.565	0.14239
7×10 VWS	PSO	676	658	667.5	5.1137
	HS	699	656	678.4	11.5732
	IHS	660	640	655.15	3.8115

从表 3~5 中可知，在 3 种硫化模式下 3 种算法进行的 20 次实验中，除了种群大小为 150 时 3×4VWS 优化结果中 PSO 算法的均方差小于 IHS 外，IHS 算法优化结果的最大值、最小值、平均值和均方差都要优于其他两种算法，可见 IHS 算法在收敛能力上有所提高。另外，从表 3~5 中可知，3 种算法在不同种群大小的情况下对硫化车间调度问题进行优化，随着种群大小的增大，算法的优化结果也随之变的更优，并且在不同情况下 IHS 算法达到的最优值以及 20 次实验优化结果的平均值都要小于其他 2 种算法。因此，IHS 算法具有较好的优化性能和收敛精度。

为了便于观察 3 种算法的实验结果，实验将种群大小为 100 时 3 种对比算法在 3×4，4×3 和 7×10 3 种硫化车间调度优化问题的平均值的收敛曲线如图 2~4 所示。

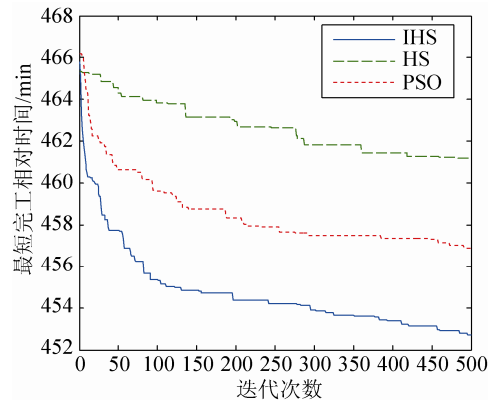


图 2 3 种算法在 3×4VWS 优化曲线
Fig. 2 Evolution curve of three algorithms on 3×4 VWS

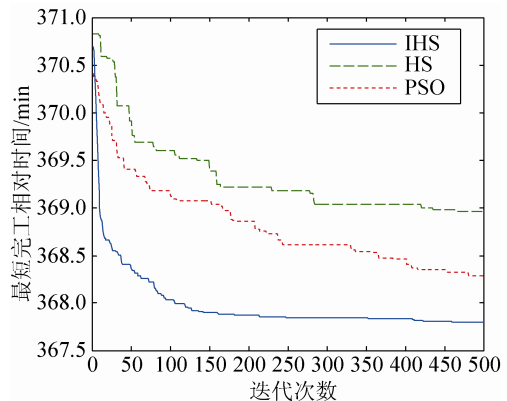


图 3 3 种算法在 4×3VWS 优化曲线
Fig. 3 Evolution curve of three algorithms on 4×3 VWS

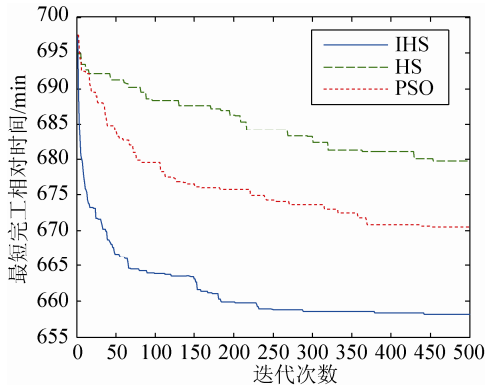


图 4 3 种算法在 7×10VWS 优化曲线
Fig. 4 Evolution curve of three algorithms on 7 × 10 VWS

从图 2~4 中 3 种算法的收敛曲线可以看出, 经

过 500 次迭代, IHS 算法的收敛幅度以及最终收敛值都要优于其余二种算法。在图 3~4 中可看出 IHS 相较于 HS 和 PSO 算法收敛速度更快, 在收敛精度上也有所提高, 这主要是因为算法迭代过程中针对 HM 内每个个体都产生新解并进行更新, 促进了 HM 整体向最优解靠近的速度。

在 20 次实验中 IHS 得到的最优调度结果甘特图如图 5~7 所示, 与表 3~5 中的结果相对应。图 7 方框中斜杠“/”前面的数字表示轮胎型号, 后面的数字表示该型号轮胎的数量。例如“2/247”表示 247 个类型 2 的轮胎。

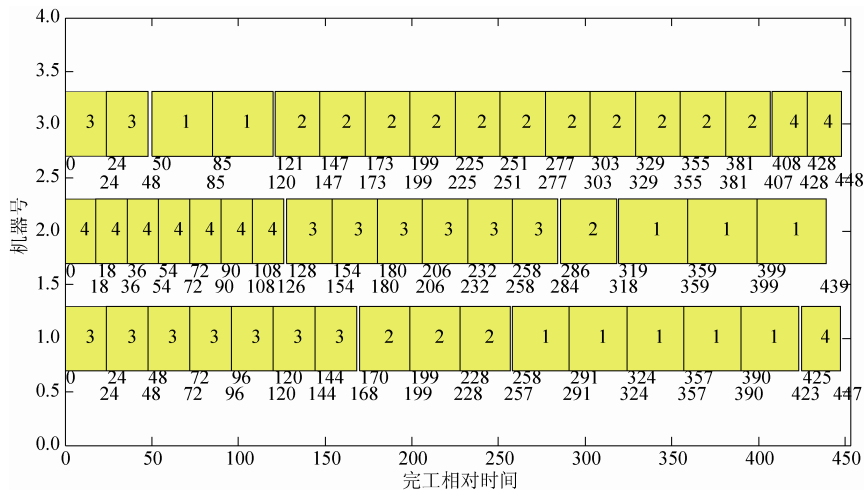


图 5 IHS 算法在 3×4VWS 中最优调度结果甘特图
Fig. 5 Gantt chart of the optimal results for IHS algorithm on 3×4 VWS

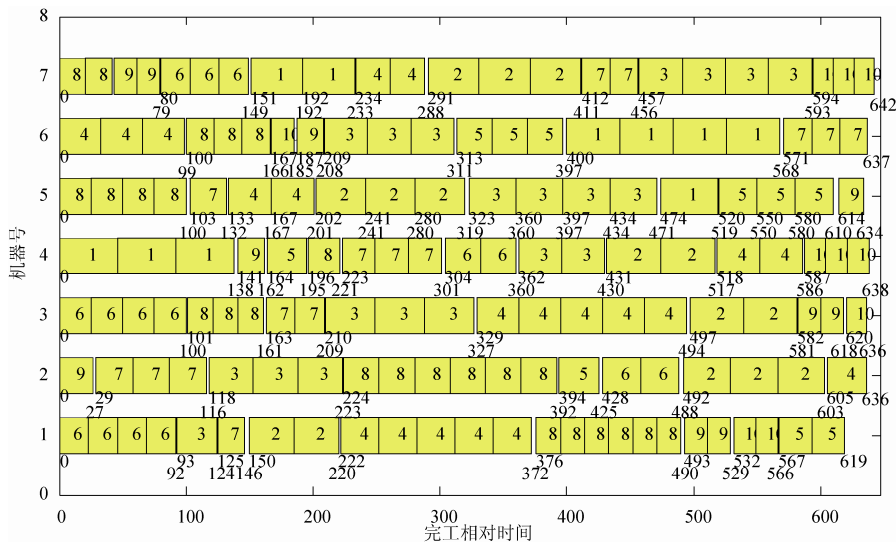


图 6 IHS 算法在 7×10VWS 中最优调度结果甘特图
Fig. 6 Gantt chart of the optimal results for IHS algorithm on 7×10 VWS

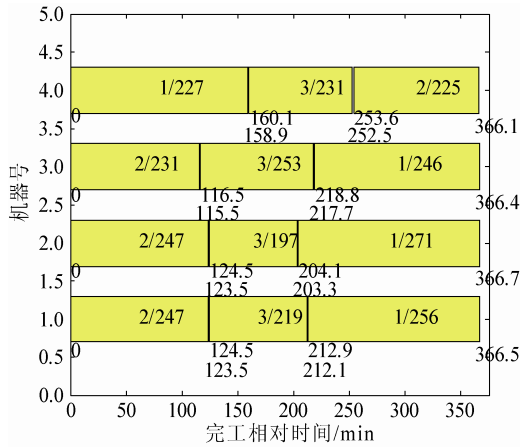


图7 IHS算法在4×3VWS中最优调度结果甘特图

Fig. 7 Gantt chart of the optimal results for IHS algorithm on 4×3 VWS

5 结论

针对硫化车间如何合理分配硫化任务,利用车间现有资源进而提高产品生产效率的问题,本文提出了一种改进的离散和声搜索算法 IHS 来对硫化车间调度进行求解。算法在每次迭代进化中,依次对记忆库中的每个个体生成一个对应的新和声。新解有 HMCR 的概率直接取 HM 内原个体的值,再以 PAR 的进行微调;有 1-HMCR 的概率在变量值域内随机取值。相应的新和声和原个体以及 HM 内最差和声作对比,如果新和声优于原个体,替换原个体。如果新和声差于原个体但优于最差和声,则替换掉最差和声,重新在 HM 内寻找最差和声。通过每次迭代提高 HM 内和声的整体质量。仿真结果分析验证了该 IHS 算法的可行性和有效性。

参考文献:

- [1] Zong W G, Kim J H, Loganathan G V. A New Heuristic Optimization Algorithm: Harmony Search [J]. Simulation Transactions of the Society for Modeling & Simulation International (S0037-5497), 2001, 76(2): 60-68.
- [2] Khalili M, Kharrat R, Salahshoor K, et al. Global Dynamic Harmony Search algorithm: GDHS [J]. Applied Mathematics & Computation (S0096-3003), 2014, 228(9): 195-219.
- [3] 翟军昌, 高立群, 欧阳海滨, 等. 一种自适应全局和声搜索算法 [J]. 控制与决策, 2015, 30(11): 1953-1959. (Zhai Junchang, Gao Liqun, Ouyang Haibin, et al. An Adaptive Global Harmony Search Algorithm [J]. Control and Decision, 2015, 30(11): 1953-1959.)
- [4] Kong X, Gao L, Ouyang H, et al. A simplified binary harmony search algorithm for large scale 0-1 knapsack problems [J]. Expert Systems with Applications (S0957-4174), 2015, 42(12): 5337-5355.
- [5] 李若平, 欧阳海滨, 高立群, 等. 学习型和声搜索算法及其在 0-1 背包问题中的应用 [J]. 控制与决策, 2013, 28(2): 205-210. (Li Ruoping, Ouyang Haibin, Gao Liqun, et al. Learned Harmony Search Algorithm and Its Application to 0-1 Knapsack Problems [J]. Control and Decision, 2013, 28(2): 205-210.)
- [6] 陈春, 汪枫, 刘蓓, 等. 基于基本环矩阵与改进和声搜索算法的配电网重构 [J]. 电力系统自动化, 2014, 38(6): 55-60. (Chen Chun, Wang Feng, Liu Bei, et al. Network Reconfiguration Based on Basic Ring Matrix and Improved Harmony Search Algorithm [J]. Automation of Electric Power Systems, 2014, 38(6): 55-60.)
- [7] 吴沛锋, 高立群, 周翔, 等. 自适应和声搜索算法在结构工程设计中的应用 [J]. 仪器仪表学报, 2012, 33(7): 1676-1680. (Wu Peifeng, Gao Liqun, Zhou Xiang, et al. Application of Adaptive Harmony Search Algorithm in Structural Engineering Design [J]. Chinese Journal of Scientific Instrument, 2012, 33(7): 1676-1680.)
- [8] Yuan Y, Xu H, Yang J, et al. A hybrid harmony search algorithm for the flexible job shop scheduling problem [J]. Applied Soft Computing (S1568-4946), 2013, 13(7): 3259-3272.
- [9] 周雅兰, 黄韬. 和声搜索算法改进与应用 [J]. 计算机科学, 2014, 41(增 1): 52-56. (Zhou Yalan, Huang Tao. Modifications and Applications of Harmony Search Algorithm [J]. Computer Science, 2014, 41(S1): 52-56.)
- [10] 黄帅, 马良. 一种改进的和声搜索算法 [J]. 小型微型计算机系统, 2012, 33(11): 2418-2420. (Huang Shuai, Ma Liang. An Improved Harmony Search Algorithm [J]. Journal of Chinese Computer Systems, 2012, 33(11): 2418-2420.)
- [11] 欧阳海滨, 高立群, 邹德旋, 等. 和声搜索算法探索能力研究及其修正 [J]. 控制理论与应用, 2014, 31(1): 57-65. (Ouyang Haibin, Gao Liqun, Zou Dexuan, et al. Exploration Ability Study of Harmony Search Algorithm and Its Modification [J]. Control Theory & Applications, 2014, 31(1): 57-65.)
- [12] 夏红刚, 欧阳海滨, 高立群. 多子群混合和声搜索算法 [J]. 东北大学学报(自然科学版), 2015, 36(2): 171-175. (Xia Honggang, Ouyang Haibin, Gao Liqun. Multiple-sub-groups Hybrid Harmony Search Algorithm [J]. Journal of Northeastern University (Natural Science), 2015, 36(2): 171-175.)
- [13] 张亚妮. 多种群协同进化的微粒群优化算法及其在橡胶硫化车间生产调度中的应用 [D]. 青岛: 青岛科技大学, 2009. (Zhang Yani. Coevolution Discrete Particle Swarm Optimization and the Application in the Production Scheduling of Rubber Vulcanization Workshop [D]. Qingdao, China: Qingdao University of Science & Technology, 2009.)