

文章编号: 2095-2163(2022)10-0219-06

中图分类号: F406

文献标志码: A

基于混合策略改进的贪心算法 PSS 订单调度研究

任春慧

(上海工程技术大学 管理学院, 上海 201620)

摘要: 对于产品服务系统调度问题,本文提出了基于改进迭代贪心算法的 PSS 订单调度混合策略求解模型,大幅度缩短了所有订单的交付时间。运用 NEH 初始化方法对最早授权服务时间进行排序,引进随机邻域搜索算法,同时在贪心算法破坏和重建阶段进行优化,最后基于轮盘赌的比例筛选策略提出新的判断准则。通过仿真实验表明,本文所设计的混合改进策略对于求解订单调度问题的效果显著,求解速度更快,并表现出更好的鲁棒性。

关键词: PSS 订单调度; 迭代贪心算法; 随机邻域搜索; 扰动算子; 轮盘赌

Research on PSS order scheduling hybrid strategy model based on improved greedy algorithm

REN Chunhui

(School of Management, Shanghai University of Engineering Science, Shanghai 201620, China)

【Abstract】 For the product service system scheduling problem, this paper proposes a PSS order scheduling hybrid strategy solution model based on improved iterative greedy algorithm, which greatly reduces the delivery time of all orders. The NEH initialization method is used to sort the earliest authorized service time, the random neighborhood search algorithm is introduced, and the greedy algorithm is optimized at the stage of destruction and reconstruction. Therefore, a new judgment criterion is proposed based on the proportion selection strategy of roulette. The simulation results show that the hybrid improvement strategy designed in this paper has a significant effect on solving the order scheduling problem, with faster solving speed and better robustness.

【Key words】 PSS order scheduling; iterative greedy algorithm; random neighborhood search; perturbation operator; roulette

0 引言

随着现代化经济的发展,市场同类型产品的竞争不断加剧,如何在更好地满足客户的个性化需求的条件下,降低生产成本,提高资源的利用率和减少环境污染,将制造业向绿色智能、服务信息化转型升级,是许多企业现在及未来的发展方向。

在当今社会服务化生产的模式下,制造生产型企业把产品和服务相结合提出产品服务系统^[1](Product Service System, PSS)。对于 PSS 的研究, Bu 等人^[2]创建 VR 平台将收集并处理用户生成的数据和 VR 系统生成的数据,实现 VR 辅助的用户体验和实时数据反馈的增值服务。Mariusz 等人^[3]改变 PSS 原有的基于销售和产品所有权从制造商转移到用户的商业模式,允许销售产品提供的可访问性和功能,为印刷行业提供更大发展空间。Kang 等人^[4]在物流行业采用 PSS 的基础上定义和阐述了 LPSS,提出了多单位维克拍卖和单面 Vickrey-Clarke-Groves (O-VCG) 组合拍卖并研究了对应的

相关属性,包括激励相容性、分配效率、预算平衡和个人理性。Zhang 等人^[5]提出了一个基于设计中心复杂性(DCC)理论的可持续 PSS 开发框架,采用 DCC 理论确定系统的复杂性类型,结合 TRIZ 子域模型对问题进行转换和求解并建立泛函周期性,降低系统复杂度,保持系统的长期运行稳定性。Haber 等人^[6]通过 Kano 模型增强了 PSS 质量功能部署方法以筛选客户需求并将有吸引力的需求转化为接收器状态参数,将模糊层次分析法集成到该过程中评估这些参数及其固有的不确定性。Li 等人^[7]针对存在个性化需求的客户,设计了块结构的马尔可夫链优化模型,较好地解决 PSS 问题。上述文献均对生产服务调度的运营及其管理模式进行了探讨并具有可参考价值,但并未提及到同时间大量 PSS 订单的调度问题,究其根源性问题是协调 PSS 订单的制造生产和满足客户的服务需求,达到服务效率和效果最优化。

为此,本文将研究产品服务系统订单调度问题,以一家有多条生产线和多支安装团队的服务型制造

基金项目: 上海市自然科学基金(19ZR1421600); 上海工程技术大学研究生创新项目(20KY0322)。

作者简介: 任春慧(1997-),女,硕士研究生,主要研究方向:群智能计算、大数据处理与分析。

收稿日期: 2021-12-13

企业为研究对象,在满足客户个性化需求的同时,寻求订单交付时间的最小化,构建数学模型,基于贪心算法求解问题,最后仿真实验验证。

1 问题描述

假设面向市场服务型制造企业 $L = \{1, 2, \dots, l\}$, 拥有 m 条生产线和 l 支安装团队, 需要满足客户的 PSS 订单需求, 如图 1 所示。从规划期开始即在零点, 企业接到一组订单, 其中包括来自 n 个 PSS 订单的订单集 $N = \{1, 2, \dots, n\}$ 。在这种情况下, 每个订单包含一个产品单元和相应的安装服务, 并指定客户要求的第一个授权服务时间, 在此之前, 客户由于各种原因(如缺乏安装条件)无法获得服务。

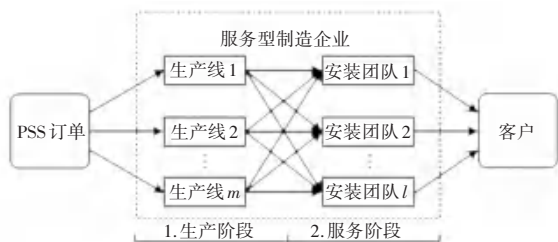


图1 PSS订单交付流程

Fig. 1 PSS order delivery process

PSS 订单调度问题的研究重点是协调生产与服务之间的平衡, 寻求最优的调度方案。对于服务型制造产业, 因为产品和服务之间存在时间上的差异, 服务需要以产品为媒介得到体现, 且两者之间存在相互约束, 所以如何协同规划生产调度已然成为亟需解决的研究课题。

2 生产服务模型

2.1 模型假设

- (1) 订单的生产和服务时间确定且已知。
- (2) 不同生产线上生产同一订单的时间相等, 且不同安装队伍的服务时间也相同。
- (3) 每个生产线和安装队伍不能同时生产和服务不同任务。
- (4) 生产和服务任务都能完成。

2.2 数学模型

首先, 定义以下符号和变量:

i, j 为订单编号; k 为生产线(安装团队)编号; h_1 为生产阶段; h_2 为服务阶段; pt_i 表示订单 i 的生产时间; st_i 表示订单 i 的服务时间; e_i 为订单 i 的最早授权服务时间; Q 表示足够大的正数; STP_i 为订单 i 的生产开始时间, CTP_i 为订单 i 的生产结束时

间; STS_i 为订单 i 的服务开始时间; CTS_i 为订单 i 的服务结束时间; y_{ikh} 表示订单 i 的任务分配给生产线(安装团队) k , 则 $y_{ikh} = 1$, 否则, $y_{ikh} = 0$; x_{ikh} 表示订单 i 的任务分配给生产线(安装团队) k , 且订单 i 早于订单 j , 则 $x_{ikh} = 1$; 否则, $x_{ikh} = 0$;

PSS 订单调度问题的数学模型如下:

(1) 为得到订单最小交付时间的总和, 目标函数为:

$$\min Z = \sum_{i \in N} CTS_i \quad (1)$$

(2) 在生产产品时, 过程不能被打断、即不间歇生产:

$$CTP_i = STP_i + pt_i \quad \forall i \in N \quad (2)$$

(3) 在生产过程中, 同一条生产线有且只有一项生产任务在执行:

$$STP_j \geq CTP_i + Q(x_{ijk1} - 1) \quad \forall i, j \in N, i \neq j, k \in M \quad (3)$$

(4) 生产的开始时间不能早于结束时间:

$$STS_i \geq CTP_i \quad \forall i \in N \quad (4)$$

(5) 同一订单的服务开始时间不能早于其最早授权服务时间:

$$STS_i \geq e_i \quad \forall i \in N \quad (5)$$

(6) 服务在提供过程中必须是连续不间断:

$$CTS_i = STS_i + st_i \quad \forall i \in N \quad (6)$$

(7) 同一时刻每支安装队伍不能同时服务不同任务:

$$STS_j \geq CTS_i + Q(x_{ijk2} - 1) \quad \forall i, j \in N, i \neq j, k \in L \quad (7)$$

(8) 每条生产线的生产任务执行次序, 以及每支安装团队的服务任务执行次序:

$$x_{ijkh} + x_{jikh} \leq y_{ikh} \quad \forall i, j \in N, i \neq j, k \in \begin{cases} M & h = 1 \\ L & h = 2 \end{cases}, h \in H \quad (8)$$

$$x_{ijkh} + x_{jikh} \geq y_{ikh} + y_{jikh} - 1 \quad \forall i, j \in N, i < j, k \in \begin{cases} M & h = 1 \\ L & h = 2 \end{cases}, h \in H \quad (9)$$

(9) 每个订单的生产任务必须被分配给一条生产线, 服务任务必须被分配给一支安装团队:

$$\sum_k y_{ikh} = 1$$

$$\forall i \in N, k \in \begin{cases} M & h = 1 \\ L & h = 2 \end{cases}, h \in H \quad (10)$$

(10) 各决策变量的取值范围:

$$STP_i \geq 0, CTP_i \geq 0, STS_i \geq 0, CTS_i \geq 0 \quad \forall i \in N \quad (11)$$

$$x_{ijkh}, y_{ikh} \in \{0, 1\}$$

$$\forall i, j \in N, i \neq j, k \in \begin{cases} M & h = 1 \\ L & h = 2 \end{cases}, h \in H \quad (12)$$

3 算法改进

在属于较大规模 NP-hard 问题^[8]的情况下,需要借助有效的智能算法求解模型。迭代贪心算法(Iterated Greedy, IG)是 Ruiz 等人^[9]提出的一种新型智能优化算法,该算法主要由邻域搜索、扰动算子和接受准则三个基本部分组成。IG 算法被提出后,因其便于实现、效率高而受到国内外学者的关注和研究,并已广泛应用于约束车间流水调度^[10]、传感器网络覆盖增强问题研究^[11]等领域。

本文提出了一种混合策略改进的 IG 算法,可以用来求解 PSS 订单调度问题。首先,对订单进行排序,然后初始化订单序列,运用改进的邻域搜索,迭代寻找最优解并不断替换,再对扰动算子重新进行破坏和重建,防止过早收敛。最后,根据轮盘赌设计出一套新的接受准则更新最优解,直到算法运行结束。

3.1 初始化

本文利用 Nawaz-Enscore-Ham(NEH)算法^[12]对 IG 初始化加以改进,通过 NEH 算法对所研究的订单最早授权服务时间重新进行排序,获得一个数列。首先选择数列中时间之和较短的前 2 个订单调度为基本序列,然后其他订单分别调入基本序列,选择最短时间和的序列为新的基本序列,接着重复以上步骤并不断调整基本序列,最后会获得一个新的数列。

3.2 邻域搜索

本文为增强 IG 算法的寻优能力,根据随机搜索和交换邻域的思想,提出一种随机邻域搜索(Random Neighborhood Search, RNS)算法,其思路主要如下:每次在当前解之中随机选取一个个体,而后再随机插入其任何可能最优预测的区域,或者将个体与最优位置的个体进行位置互换,一旦新解优于原解,则保留新解并重复上述过程,直至不再生成较高解。这个方式能够大幅增加搜寻的范围并提高寻优效果。

3.3 破坏与重建

为了避免 IG 算法在求解过程中陷入局部最优,需要对其运行过程进行破坏与重建、跳出原有机理,又称为扰动算子。原始 IG 算法的扰动算子在发生作用之后,可能会发生丢失局部最优解的情况,为此本文提出一种结合破坏(Destruction)、优化(Optimization)和重建(Construction)过程(DOC)作为改进 IG 算法的扰动算子,其操作过程如下:在实施破

坏过程后,将得到的解重新进行多次邻域搜索得到优化解,此后再对其进行重建,提升了扰动解的优良性。

3.4 阈值接受准则

为实现 IG 算法更快找出最好解这一目的,本文提出一种基于轮盘赌选择(Roulette Wheel Selection, RWS)策略的接受准则。根据求解问题的特点,设计一个优选表,在优选表中依次添加局部搜索产生的较优解,当达到表的长度时删除表中原有最差解,最后利用 RWS 选择表中解。设置优选表长 τ ($\tau \leq \omega$),对 RWS 选择策略步骤可给出阐释分述如下。

步骤 1 对优选表中的解进行归一化处理:

$$Z'_i = \frac{Z_i - Z_{\min}}{Z_{\max} - Z_{\min}} \quad (13)$$

步骤 2 优选表中第 i 个解的选择概率为 $p_i = f_i / \sum_{i=1}^r f_i$, $f_i = 1/\exp(Z_i)$ 为第 i 个解的适应度函数。计算第 i 个解的累积概率 $q_i = \sum_{j=1}^i p_j$ 。

步骤 3 在 $[0, 1]$ 中选择随机数 $Rand$ 。如果 $Rand < q_1$, 则选择优选表中第一个解;否则,选择表中第 i 个解,使得 $q_{i-1} \leq Rand < q_i$ 成立。

3.5 终止条件

考虑到问题规模,保证算法充分收敛,本文设置改进 IG 算法的终止条件为 $10 * n^2$ ms。

3.6 算法流程

研发后得到的整个算法的求解流程具体如下。

Set λ, ω ;

$\pi_0 := NEH$;

$\pi := RNS(\pi_0)$;

$\pi^* := \pi$;

$EliteList := \{\pi\}$

While CPU time $\leq 100 n$ do

$\pi' := DOC(\pi, \lambda)$;

$\pi'' := RNS(\pi')$;

$\{\pi, \pi^*, EliteList\} : RWS(\pi'', \pi^*, EliteList, \omega)$;

end while

return π^*

end procedure

4 仿真实验

4.1 算例构造与性能评价指标

本文从测试集合中随机选择测试算例。设置 pt_i 和 st_i 由区间 $[1, 100]$ 的均匀分布随机产生, e_i 由区间 $[pt_i, (1 + \theta)pt_i]$ 均匀分布随机产生。实验

中关键参数的取值从 108 种参数集合中选取, $n = \{50, 100, 150, 200\}$ 、 $m = \{2, 5, 10\}$ 、 $l = \{2, 3, 5\}$ 和 $\theta = \{1, 2, 3\}$ 。每个参数序列随机产生 2 个算例,最后可形成 216 个基准测试集合。

使用相对偏差指数 (relative deviation index, RDI) 评价 IG 的能力:

$$RDI = \frac{Z^{alg} - Z^{best}}{Z^{worst} - Z^{best}} \times 100 \quad (14)$$

4.2 实验参数设置

设置 50 个算例,每个算例通过选择 $n' = \{20, 50, 80, 100, 120, 150, 180, 200\}$ 以及 m 、 l 和 θ 集合中随机值。MIG 算法主要测试 λ 和 ω , 设置: $\lambda = \{1, 2, 3, 4\}$ 和 $\omega = \{5, 6, 7, 8\}$ 。测试结果分别取运行 5 次后的平均值。

运用双因素方差分析所得结果,见表 1。当 p - Value 值比 0.05 小的时候,代表着对应因子的影响显著。由表 1 可见, λ 对 MIG 影响显著,而 ω 对 MIG 则表现出较为明显的影响性。并且在 λ 和 ω 同时产生作用的情况下,对 MIG 的性能影响也不明显。

表 1 MIG 算法参数设置实验的 ANOVA 结果

Tab. 1 ANOVA results of MIG algorithm parameters setting experiment

Source	Sum of squares	Df	Mean square	F - Ratio	p - Value
Main effects					
A: λ	69 119.1	3	23 039.7	93.59	0
B: ω	288.5	3	96.2	0.39	0.759 8
Interactions					
AB	827.5	9	91.9	0.37	0.947 6
Error	114 225.6	464	246.2		
Total	184 460.7	479			

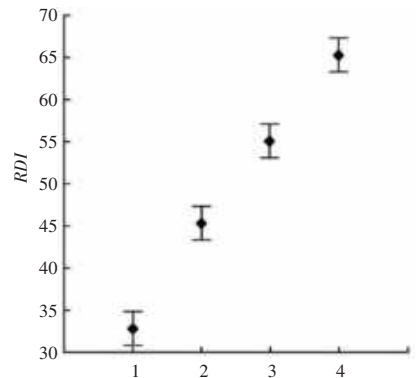
为进一步研究 λ 和 ω 对 MIG 算法的影响,对参数选取不同值得到 RDI 均值和 95%LSD 置信区间,如图 2 所示。由图 2 可以看出,MIG 的性能随着 λ 的增大而下降,而 ω 对 MIG 的影响并不显著。取 $\lambda = 1$, $\omega = 6$ 。

4.3 有效性与鲁棒性分析

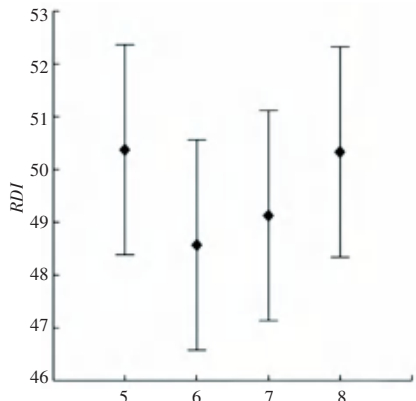
将混合策略改进的贪心算法分别与经典 IG 算法、随机邻域改进的贪心算法 (IGRNS)、扰动算子改进的贪心算法 (IGDOC)、轮盘赌改进接受策略的贪心算法 (IGRWS) 进行对比实验,分别测试不同改进策略效果。

IG 算法改进的 RDI 均值和 95% LSD 置信区间图如图 3 所示。由图 3 可以看出本文提出的改进策略对于提升贪心算法的性能均是有效的,MIG 效果

最为明显, RDI 值优于 IG 约为 84%。同时,IGRNS 的 RDI 值优于 IG 算法约 35%,IGDOC 的 RDI 值优于 IG 算法约 58%,IGRW 算法的 RDI 值优于 IG 算法约 5%,明显地,破坏重建过程中的扰动算子对算法性能的影响最为显著。



(a) λ 的 RDI 均值和置信区间



(b) ω 的 RDI 均值和置信区间

图 2 λ 和 ω 分别取不同值时改进 IG 算法置信区间

Fig. 2 Improved confidence interval of IG algorithm when λ and ω take different values respectively

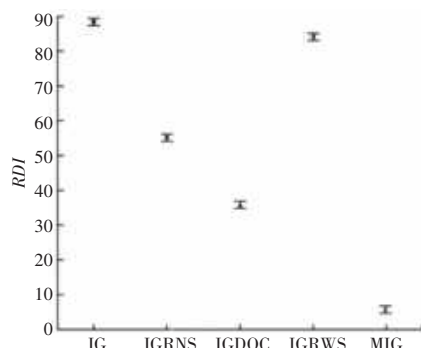


图 3 IG 算法改进的 RDI 值

Fig. 3 Improved RDI value of IG algorithm

MIG 算法对求解算例中的具有关键性作用的参数敏感性变化趋势如图 4~图 7 所示。

由图 4 可以看出,随着 n 的增加,MIG 的 RDI 均值呈现明显的减小,而 IG 的 RDI 均值持续上涨,这反映了 MIG 算法对于求解更加庞大、及复杂调度类型问题的效果会更为准确。由图 5 和图 6 可见,MIG 的 RDI 值几乎不随着 l 和 m 的增加而波动,这反映了生产产品的流水线数量和安装团队数量的增加几乎不会降低 MIG 算法性能,反映了 MIG 对 l 和 m 的鲁棒性能较优。在图 7 中,随着 θ 的增大,各订单的最早授权服务时间就越分散,并同样反映出 MIG 较优的鲁棒性。

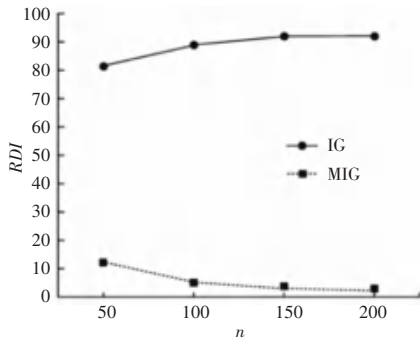


图 4 MIG 关于 n 的变化趋势

Fig. 4 Variation trend of MIG on n

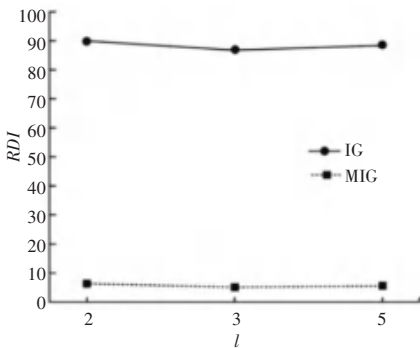


图 5 MIG 关于 l 的变化趋势

Fig. 5 Variation trend of MIG on l

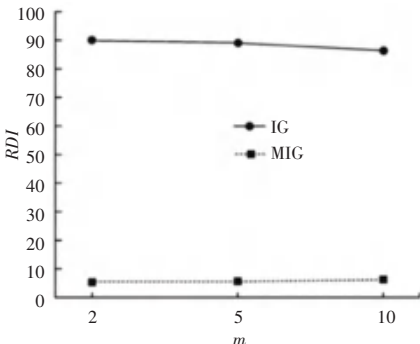


图 6 MIG 关于 m 的变化趋势

Fig. 6 Variation trend of MIG on m

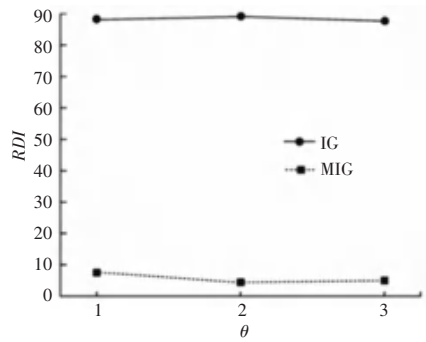


图 7 MIG 关于 θ 的变化趋势

Fig. 7 Variation trend of MIG on θ

5 结束语

本文面向具有数条产品生产流水线和安装队伍的生产服务型企 业,可以为 客户推荐较为合适的订单交货前的最早授权服务日期,减少不必要的等待时间,寻求最短交货时间,建立了标准数学模型,并针对 PSS 订单调度问题特点设计了求解问题的混合策略改进 IG 算法。通过仿真测试,证明了 IG 算法改进策略的有效性和鲁棒性。本文提出的 PSS 订单调度模型及 IG 算法混合改进策略求解方法可以为企 业决策提供有效的决策建议,为消费者提高服务水平降低成本。在未来的研究中可以将本文方法推广到汽车制造,交通调度等行业中。

参考文献

[1] BAINES T, LIGHTFOOT H, EVANS S, et al. State of the art in product-service systems [J]. Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture, 2007, 221(10): 1543-1552.

[2] BU Lingguo, CHEN C H, NG K K H, et al. A user-centric design approach for smart product-service systems using virtual reality: A case study [J]. Journal of Cleaner Production, 2021, 280 (P2): 124413.

[3] MARIUSZ S, KRASLAWSKI A, LIPIAK J, et al. Product-service system business model for printing houses [J]. Journal of Cleaner Production, 2020, 274: 122939.

[4] KANG Kai, ZHONG R Y, XU Suxiu, et al. Auction-based cloud service allocation and sharing for logistics product service system [J]. Journal of Cleaner Production, 2020, 278: 123881.

[5] ZHANG Peng, JING Shaohua, NIE Zifeng, et al. Design and development of sustainable product service systems based on design-centric complexity [J]. Sustainability, 2021, 13(2): 1-27.

[6] HABER N, FARGNOLI M, SAKAO T. Integrating QFD for product-service systems with the Kano model and fuzzy AHP [J]. Total Quality Management & Business Excellence, 2020, 31: 9-10.

[7] LI Na, JIANG Zhibin. Modeling and optimization of a product-service system with additional service capacity and impatient customers [J]. Computers & Operations Research, 2013, 40(8): 1923-1937.