

DOI:10.19344/j. cnki. issn1671-5276.2024.03.027

基于改进型哈里斯鹰算法的云制造服务组合优化方法研究

张舒淇,唐敦兵,张毅,周世辉,王威奇

(南京航空航天大学 机电学院,江苏 南京 210016)

摘 要:云制造环境下,云平台以制造服务组合的形式为个性化需求提供按需服务,可显著提升订单响应速度与提高资源利用率。采用哈里斯鹰算法构建制造服务组合,针对制造服务间物流转运问题,建立独特的编码与解码机制。为解决该算法存在早熟收敛问题,引入 Logistic 一维混沌系统,设计非线性逃逸能量更新机制和 3 种邻域搜索策略。通过实验验证了改进后的哈里斯鹰算法在解决云制造服务组合优化问题时具有显著优越性。

关键词:云制造;服务组合;哈里斯鹰算法;任务调度

中图分类号:TH164 **文献标志码:**A **文章编号:**1671-5276(2024)03-0127-05

Research on Cloud Manufacturing Service Composition Method Based on Improved Harris Hawks Optimization Algorithm

ZHANG Shuqi, TANG Dunbing, ZHANG Yi, ZHOU Shihui, WANG Weiqi

(College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract:In cloud manufacturing environment, the cloud platform provides on-demand services for personalized needs in the form of manufacturing service composition, which can significantly improve order response speed and resource utilization. This paper uses Harris Hawks optimization algorithm to construct manufacturing service composition, and establishes a unique coding and decoding mechanism for the logistics transfer between manufacturing services. In order to solve the premature convergence problem of the algorithm, the Logistic one-dimensional chaotic system is introduced, the nonlinear escape energy updating mechanism and three types of neighborhood search strategies are designed. Experiments verify the significant advantages of the improved Harris Hawk algorithm in solving the cloud manufacturing service portfolio optimization problem.

Keywords:cloud manufacturing; service composition; Harris Hawks optimization algorithm; task scheduling

0 引言

制造业技术的不断突破推动了智能制造领域的蓬勃发展。为解决产能过剩并提高分布式制造资源的利用率,制造服务优化配置问题受到了密切关注。

目前针对制造服务优化配置问题已经有很多研究成果。DAO 等^[1]针对多企业的协同生产问题进行深入研究,提出一种独特的遗传算法,使得求解效率大幅提高。AKBARIPOUR 等^[2]针对分布式制造资源之间的物流问题对服务质量的影响进行充分研究,提出一种混合整数规划模型。WANG 等^[3]在制造服务异常的情况下提出了一种新的编码形式,可以对制造服务的异常情况进行动态重构。ZHANG 等^[4]提出了制造社区的概念与一种基于多目标优化的制造服务选择和组合

方法。LIANG 等^[5]提出一种基于深度强化学习算法的 QoS-aware 云制造服务组合优化方法,规避了元启发式算法参数设置复杂的问题。王平等^[6]提出一种基于双层规划模型的资源服务组合方法并采用云遗传算法对模型进行求解。

目前大多数的优化配置模型对多个任务与多个子任务进行求解的较少。为此,本文提出一种基于改进哈里斯鹰算法,提高了全局的搜索能力,以最小化加工成本为目标,解决了分布式制造服务优化配置的问题。

1 制造服务优化配置问题分析

在云制造环境中,接受订单的主体企业会将一部分订单需求分配给合适的制造商,以减轻自己的生产压力。因此需要在满足制造任务要求的情况下,选取合适的制造商和合适的物流线路,使

基金项目:国家重点研发计划项目(2020YFB1710500);江苏省重点研发计划项目(BE2021091);成飞-南航“智汇蓝天”校企协同育人项目(2022QYGCSJ38)

第一作者简介:张舒淇(1998—),男,辽宁锦州人,硕士研究生,研究方向为云平台、数据中台,paleatta@nuaa.edu.cn。

得所有任务的完工总成本最小,具体的问题研究模型如图1所示。

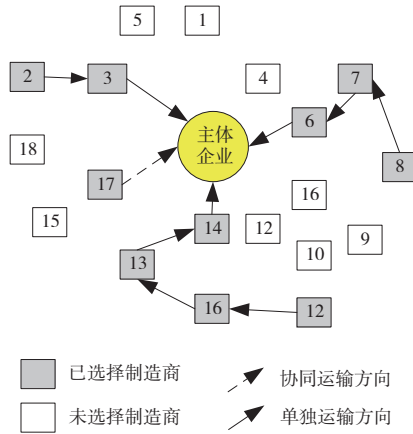


图1 制造服务优化配置

为了能够更加直观地描述这一模型,现对该模型进行如下假设:

1) 每个制造任务的不同制造子任务在不同制造商可以并行加工,不同制造任务的总装任务也可以在主体企业并行完成。

2) 不同的制造商与制造商之间,制造商与主体企业之间只有一次运输。

3) 货物运输时运载能力为无穷大,同时货物都可以在一次运输的过程中完成。

基于以上假设,本文建立以最小化成本为目标函数的模型,目标函数如下所示。

$$F = C_M + C_T \quad (1)$$

式中: C_M 表示不同制造商的制造成本; C_T 表示在整个生产过程中由于物流而产生的成本。

$$C_M = \sum_{k=1}^a \sum_{j=1}^m \sum_{i=1}^{n_j} P_{ij}^k \alpha_{ij}^k \quad (2)$$

$$C_T = C_{T1} + C_{T2} \quad (3)$$

$$C_{T1} = \sum_{k_1=1}^a \sum_{k_2=1}^a \left\{ \left[F_{k_1 k_2} + l_{k_1 k_2} \left(\sum_{j=1}^m \sum_{i=1}^{n_j} \beta_{ij}^{k_1 k_2} \right) \right] \cdot \text{sign} \left(\sum_{j=1}^m \sum_{i=1}^{n_j} \beta_{ij}^{k_1 k_2} \right) \right\} \quad (4)$$

$$C_{T2} = \sum_{k=1}^a \left\{ \left[F_{k_0} + l_{k_0} \left(\sum_{j=1}^m \sum_{i=1}^{n_j} \gamma_{ij}^{k_0} \right) \right] \text{sign} \left(\sum_{j=1}^m \sum_{i=1}^{n_j} \gamma_{ij}^{k_0} \right) \right\} \quad (5)$$

式(1)表示所有任务的制造成本包括制造商和主体企业的制造成本两部分;式(2)表示制造商的制造成本包括所有制造商的制造任务成本总和;式(3)表示物流成本包括制造商之间的物流成本(式(4))和运输到主体企业之间的物流成本(式(5))两部分。式中符号的描述如表1所示。

表1 目标函数符号描述

符号	描述
k	制造商编号
a	制造商的个数
i	制造子任务编号
m	制造任务的数目
j	制造任务编号
α_{ij}^k	如果制造商 k 有能力并完成第 j 个制造任务的第 i 个制造子任务则为 1, 否则为 0
$\beta_{ij}^{k_1 k_2}$	如果第 j 个制造任务的第 i 个制造成品从制造商 k_1 运往制造商 k_2 则为 1, 否则为 0
$\gamma_{ij}^{k_0}$	如果第 j 个制造任务的第 i 个制造成品从制造商 k 运往主体企业则为 1, 否则为 0
n_j	第 j 个制造任务中共有 n 个制造子任务
P_{ij}^k	由制造商 k 加工的制造任务 j 的第 i 制造子任务所产生的成本
$F_{k_1 k_2}$	制造商 k_1 运往 k_2 由于运输距离产生的成本
F_{k_0}	制造商 k 运往主体企业由于运输距离产生的成本
$l_{k_1 k_2}$	制造商 k_1 运往 k_2 由于运输数目产生的成本
l_{k_0}	制造商 k 运往主体企业由于运输数目产生的成本
$\text{sign}(x)$	如果 $x > 0$ 则为 1, 否则为 0

2 制造服务优化配置问题模型求解

2.1 编码与解码

本文主要考虑3个问题:多个制造任务、制造商的选择和物流方式的选择,因此种群编码也与这3部分有关,如图2所示。

项目	制造商选择										物流运输选择										
	制造任务1					制造任务2			制造任务3		运输节点序列										
子任务序列/物流节点序列	1	2	3	4	5	1	...	6	1	...	7	1	2	3	4	5	6	7	8	9	10
制造商节点/物流节点决策变量	3	5	3	10	4	4	...	1	1	...	3	3	5	2	1	1	8	9	4	1	6

图2 改进型哈里斯鹰算法编码

图2中制造任务由制造任务数目与制造任务子任务的数目有关。深色序列的上半部分表示制造子任务序列,深色序列的下半部分表示制造子任务的制造商。物流运输选择包括物流节点序列

和物流节点决策变量两个部分。物流部分的上方表示物流节点序列,其具体的节点序列与制造子任务序列有关,主要满足以下规则:

1) 运输节点序列中有意义数目与第一部分的

不重复制造商节点数有关;

2) 运输节点序列的顺序与第一部分制造商节点的序列顺序相同。

转化后得到的物流部分编码如图 3 所示,无色部分无意义。深色上方序列表示运输节点序列,深色下方序列表示对应节点的决策变量。

项目	运输节点序列									
运输节点序列	3	5	10	4	1	6	7	8	9	10
运输节点决策变量	3	5	2	1	1	8	9	4	1	6

图 3 物流编码

决策变量的解析主要满足以下规则:

- ① 具有相同的决策变量的运输节点在同一次运输过程中;
- ② 不同的运输节点在同一个运输过程中,则物流路线总是从左侧向右侧运输;
- ③ 所有运输线路的终点都是主体企业;
- ④ 如果没有相同的决策变量,则该制造商完成任务后直接运输到主体企业。

根据以上规则,可以得出 5 条物流线路,其中 4 条分别是 3 号、5 号、10 号、6 号直接运输到主体企业;第 5 条物流线路是 4 号,首先运输到 1 号然后再运输到主体企业。

2.2 种群初始化

为了使算法的初始解具有更强的全局搜索能力,引入一维 logistic 混沌系统,如式(6)所示。

$$x(i+1) = \mu \times x(i) \times (1 - x(i)) \quad (6)$$

初始化流程如下(图 4)。

1) 生成小数序列:随机生成范围为(0,1)的 28 个混沌个体与种群的 28 个维度相对应,完成第一步初始化。

2) 整数化:取得的 28 个混沌个体,扩大 10 倍,并向上取整得到整数序列。

3) 物流编码处理:在整数序列中,前 18 位是制造商选择部分,后 10 位是物流部分。将制造商选择部分在不改变顺序的情况下,除去重复的制造商,得到运输节点序列。

1	2	3	4	5	...	6	7	1	2	3	4	5	6	7	8	9	10
0.211	0.416	0.261	0.972	0.388	...	0.529	0.225	0.262	0.478	0.118	0.009	0.035	0.773	0.890	0.360	0.018	0.547

1	2	3	4	5	...	6	7	1	2	3	4	5	6	7	8	9	10
3	5	3	10	4	...	6	3	3	5	2	1	1	8	9	4	1	6

1	2	3	4	5	...	6	7	3	5	10	4	1	6	7	8	9	10
3	5	3	10	4	...	6	3	3	5	2	1	1	8	9	4	1	6

图 4 种群初始化

2.3 种群更新策略

本文提出 MHHO (Multi-HHO) 算法,在 HHO 算法的基础上通过改进逃逸能量更新策略和增加邻域搜索策略两种方式,增加算法的全局搜索能力,改进后的 HHO 算法(MHHO)流程如图 5 所示。

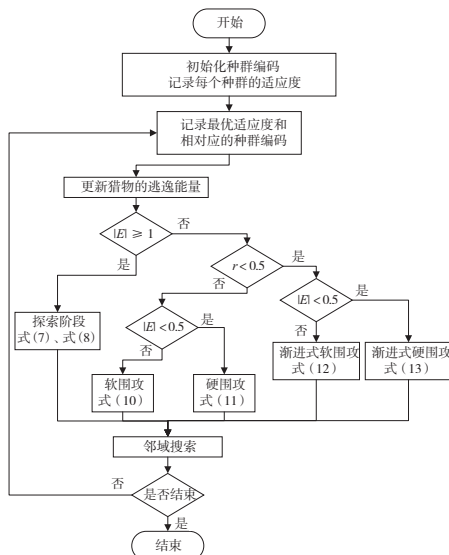


图 5 改进后的算法(MHHO)

搜索阶段如式(7)和式(8)所示,在开始阶段,HHO 算法会根据初始种群,在可行解空间内进行搜索。

$$\mathbf{x}(t+1) =$$

$$\begin{cases} x_{\text{rand}}(t) - r_1 |x_{\text{rand}}(t) - 2r_2 \mathbf{x}(t)| & q \geq 0.5 \\ (x_{\text{rabbit}}(t) - x_m(t)) - r_3 [L_B + r_4 (U_B - L_B)] & q < 0.5 \end{cases} \quad (7)$$

$$x_m(t) = \frac{1}{N} \sum_{i=1}^N \mathbf{x}(t) \quad (8)$$

式中: $\mathbf{x}(t)$ 和 $\mathbf{x}(t+1)$ 分别表示本次迭代和下一次迭代过程中 Harris 鹰的位置向量; $x_m(t)$ 表示本次迭代中 Harris 鹰种群的平均位置; $x_{\text{rand}}(t)$ 是猎物的位置; r_1 、 r_2 、 r_3 、 r_4 和 q 是区间(0,1)中的随机数; U_B 、 L_B 分别表示变量的上限和下限; x_{rand} 是当前种群中随机选择的 Harris 鹰的位置; N 表示种群数量。

由图 5 可以看出,开发阶段较为复杂,HHO 算法会根据猎物的捕捉概率 r 和猎物的逃逸能量 E 两部分确定 Harris 鹰的迭代点位。逃逸能量的更新如式(9)所示。

$$E = 2 \times (1 - t/T) \times E_0 \quad (9)$$

式中: E 表示猎物的逃逸能量; E_0 表示区间 $(-1, 1)$ 的随机数; t 表示当前的迭代次数; T 表示算法预设的最大迭代次数。当 $|E| \geq 0.5$ 时表示有较多的逃逸能量。 $r \geq 0.5$ 表示在本次捕猎中, 猎物最终被追捕。基于这两个参数, 在这个阶段包括 4 种迭代策略。

1) 软围攻: 当 $|E| \geq 0.5$ 和 $r \geq 0.5$ 时, 如式 (10) 所示。

$$\mathbf{x}(t+1) = \begin{cases} Y: x_{\text{rabbit}}(t) - E |Jx_{\text{rabbit}}(t) - \mathbf{x}(t)| & \text{if } (F(Y) < F(\mathbf{x}(t))) \\ Z: Y + S \times L_F(D) & \text{if } (F(Z) < F(\mathbf{x}(t))) \end{cases} \quad (12)$$

式中: D 表示种群的维度; S 表示一个随机个体; L_F 的 Levy 飞行函数; $<$ 表示更优。

$$\mathbf{x}(t+1) = \begin{cases} Y: x_{\text{rabbit}}(t) - E |Jx_{\text{rabbit}}(t) - x_m(t)| & \text{if } (F(Y) < F(\mathbf{x}(t))) \\ Z: Y + S \times L_F(D) & \text{if } (F(Z) < F(\mathbf{x}(t))) \end{cases} \quad (13)$$

5) 逃逸能量更新策略

文中通过使用一种非线性逃逸能量更新策略, 其逃逸能量更新如式 (14) 所示, 曲线如图 6 所示。

$$E = 2 \times (1 - (t/T)^{\frac{1}{3}})^{\frac{1}{3}} \times E_0 \quad (14)$$

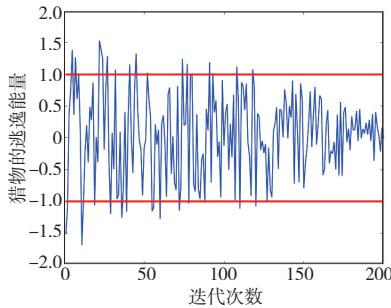


图 6 改进后的逃逸能量

由图 6 可以看出, 改进的逃逸能量曲线中在 120 次迭代左右依然有一定的概率进入搜索阶段, 能够在全局的范围内搜索更优的可行解, 大大提高了算法的全局搜索能力。

6) 邻域搜索策略

邻域搜索通过不同种群个体进行交叉、置换等操作, 寻找该个体附近的优秀编码。在本算法中主要包括 3 种邻域搜索策略:

策略 1, 随机选择 1 个个体, 随机选择某个位置, 将该位置的编码重新生成;

策略 2, 随机选择 2 个个体, 随机选择 2 个位置, 将 2 个位置之间的编码交换;

策略 3, 随机选择 2 个个体, 随机选择位置, 将指定位置的编码进行交换。

$$X(t+1) = X_{\text{rabbit}}(t) - X(t) - E |JX_{\text{rabbit}}(t) - X(t)| \quad (10)$$

式中 $J = 2(1 - r_5)$ 表示猎物尝试逃离时的能量强度, 其中 $r_5(0, 1)$ 是一个随机数。

2) 硬围攻: 当 $|E| < 0.5$ 和 $r \geq 0.5$ 时, 如式 (11) 所示。

$$X(t+1) = X_{\text{rabbit}} - E |X_{\text{rabbit}}(t) - X(t)| \quad (11)$$

3) 渐进式软围攻: $|E| \geq 0.5$ 和 $r < 0.5$, 如式 (12) 所示。

4) 渐进式硬围攻: 当 $|E| < 0.5$ 和 $r < 0.5$ 时, 如式 (13) 所示。

3 仿真实验与分析

3.1 算法设计

文中设计的制造服务优化配置仿真实验包括 3 种情况: 制造规模为 3×10 、 4×10 、 5×10 , 表示制造任务数目分别为 3、4、5 个, 制造商的数目为 10 个。其中制造任务 1—5 的生产周期分别为 27 d、26 d、25 d、26 d 和 28 d。

3.2 仿真结果分析

为了验证文中 MHHO 算法的优越性, 将相同的问题与 HHO、PSO (particle swarm optimization)、NSGA-II (non-dominated sorting genetic algorithm-II) 4 种算法在上述仿真环境中进行比较。仿真实验利用 MATLAB2021a 进行程序编写, 运行环境为 Win10 系统, 2.6 GHz, 16 G 内存。每个算法的种群规模为 100, 迭代次数选定为 200 次。收敛曲线对比如图 7—图 9 所示。

从图 7—图 9 中可以看出, MHHO 算法具有最低的最小化成本, 在不陷入局部收敛的情况下, 具有最快的收敛速度。同时, 由于引入 Logistic 一维混沌系统, MHHO 算法在初始具有最低的适应度。MHHO 算法的中期乃至后期都具有一定的搜索能力, 并没有出现局部收敛的问题。

将上述算法重复运行 30 次, 最优解、平均解、平均收敛次数和平均求解时间等结果如表 2 所示。不同制造规模下 MHHO 算法都可以获得更高质量的最优解和平均解。在平均收敛次数方面, MHHO 算法稍劣于 HHO 和 PSO, 但是有效避免了陷入局部最优的问题。相较于 NSGA-II,

MHHO 算法的平均收敛次数更低,最优解与平均解的质量更高。MHHO 算法的平均求解时间最

长。但随着制造任务数量的增多,平均求解时间与其他算法的差距越来越小。

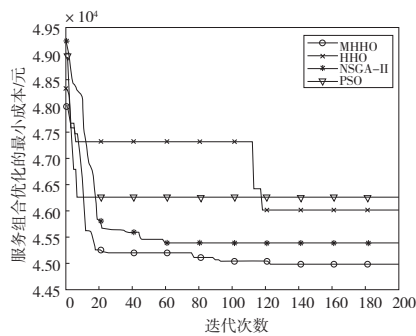


图7 制造规模为 3×10 的情况下收敛曲线比较

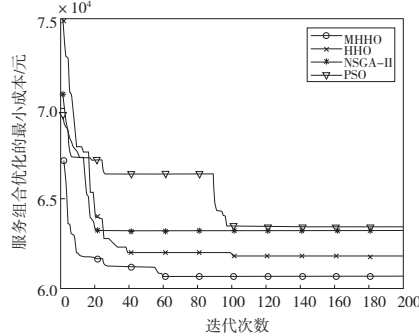


图8 制造规模为 4×10 的情况下收敛曲线比较

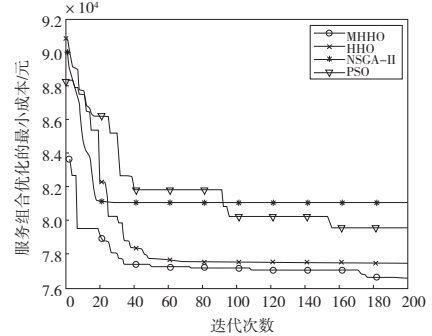


图9 制造规模为 5×10 的情况下收敛曲线比较

表2 不同制造任务规模下多种算法的求解性能比较

算法	最优解			平均解			平均收敛次数			平均求解时间/s		
	3×10	4×10	5×10	3×10	4×10	5×10	3×10	4×10	5×10	3×10	4×10	5×10
MHHO	31 415	42 954	54 340	31 530	43 158	54 623	32	44	52	4.487	4.983	5.344
HHO	32 408	44 750	57 588	33 077	45 446	58 274	21	52	83	3.918	4.379	5.049
NSGA-II	31 571	43 184	54 640	31 709	44 107	55 478	37	51	78	4.212	4.875	5.331
PSO	32 521	45 414	57 248	33 251	44 392	58 584	19	24	32	3.795	4.067	4.914

4 结语

针对分布式制造服务优化配置问题,文中从制造任务数目、制造商选择和物流方案三方面进行研究,建立优化目标模型,设计独特的编码与解码方式。针对 HHO 算法全局搜索能力差的问题,从种群初始化、逃逸能量更新和邻域搜索 3 个角度进行改进。最后,通过不同规模制造任务的案例,验证了 MHHO 算法在求解质量、收敛速度方面的有效性和优越性。同时随着制造规模的增加,MHHO 算法将具有更强的优势。

参考文献:

- [1] DAO S D, ABHARY K, MARIAN R. An innovative GA for optimisation of integrated manufacturing - transportation scheduling in VCIM systems[J]. Operational Research, 2020, 20(3): 1289-1320.
- [2] AKBARIPOUR H, HOUSHMAND M. Service composition and optimal selection in cloud manufacturing: Landscape analysis and optimization by a hybrid imperialist

competitive and local search algorithm [J]. Neural Computing and Applications, 2020, 32(15): 10873-10894.

- [3] WANG Y, WANGS, KANG L, et al. An effective dynamic service composition reconfiguration approach when service exceptions occur in real - life cloud manufacturing[J]. Robotics and Computer - Integrated Manufacturing, 2021, 71: 102143.
- [4] ZHANG Y, TANG D B, ZHU H H, et al. A flexible configuration method of distributed manufacturing resources in the context of social manufacturing [J]. Computers in Industry, 2021, 132: 103511.
- [5] LIANG H, WEN X, LIU Y, et al. Logistics - involved QoS-aware service composition in cloud manufacturing with deep reinforcement learning [J]. Robotics and Computer-Integrated Manufacturing, 2021, 67: 101991.
- [6] 王平,肖涵,潘燕华. 基于双层规划的云制造资源服务组合[J]. 计算机集成制造系统, 2022, 28(1): 51-58.

收稿日期:2022-11-11