

面向综合作业调度的簇群优化算法

隗 昊¹ 李正光¹ 陈 恒¹ 李远刚²
WEI Hao LI Zhengguang CHEN Heng LI Yuangang

摘 要

传统遗传算法不仅易出现早熟现象,而且计算实际应用中的大规模综合作业调度解效率较差。通过研究遗传编码规律和并行计算原理,提出了一种簇群并行遗传算法。利用簇群抱团思想对种群进行划分,将划分后的种群分布到不同的并行机上运行,以此提高初始种群的多样性,保证算法的执行效率和搜索空间。同时,设计了一种动态种群调整策略,在各“簇”种群趋于局部最优时自动调整种群结构,从而达到全局收敛。在FT10问题和一组综合作业调度数据上的仿真结果表明,算法在保证寻优速度的同时提高了了解的稳定性。

关键词

早熟现象; 编码规律; 并行遗传算法; 簇群; 全局收敛

doi: 10.3969/j.issn.1672-9528.2024.04.017

0 引言

遗传算法是经典的仿生优化算法,在求解作业车间调度中有广泛的应用^[1]。然而,传统遗传算法在求解综合作业调度问题时,因为算法本身的并行性限制和任务的复杂性,常常导致效率低下,所以学者根据遗传算法的内在并行性,提出了并行遗传算法^[2-4]。

并行遗传算法是对传统遗传算法的一种改进和扩充,是对遗传算法的天然并行性的显式运用,基于并行机原理并采用多种群协同进化,在一定程度上提高了了解的质量和求解速度,然而,并行遗传算法没有根本解决遗传算法易出现早熟^[5]的缺点,同时还必须考虑到通信开销问题。

综合作业调度(comprehensive job-shop scheduling problem, CJSSP)^[6-7]是同时考虑加工与装配、装配零件和组件的一类复杂调度问题。CJSSP问题在实际生产中较为常见,但对按照工艺布局、具有装配作业排序的生产调度问题,现有的研究成果很少,零散见于一些调度相关的论文中,其中大多数是基于规则的调度研究^[8-10]。

本文在研究遗传算法解决作业车间调度问题(job-shop scheduling problem, JSSP)的编码规律基础 and 经济学中的“集群”思想上,针对CJSSP问题和早熟现象,提出了一种簇群并行遗传算法(parallel genetic algorithm based on dividing

population, PGABDP)。算法通过将具有相同特征的个体集合在一起形成子群,并利用调整策略自动调整子群的组成,保证了群体的多样性,有效减少了陷入局部最优的可能性;通过对各个子群中的近优解的收集和释放,提高了寻优的速度。实验表明,本文算法在提高解的质量、降低陷入局部最优的概率、提高算法的收敛速度上有了较大的提高。

1 簇群并行遗传算法

本文算法由服务端程序和并行机程序组成,初始时根据并行机数量和簇群思想将种群划分为若干个子群。算法框架如图1所示。

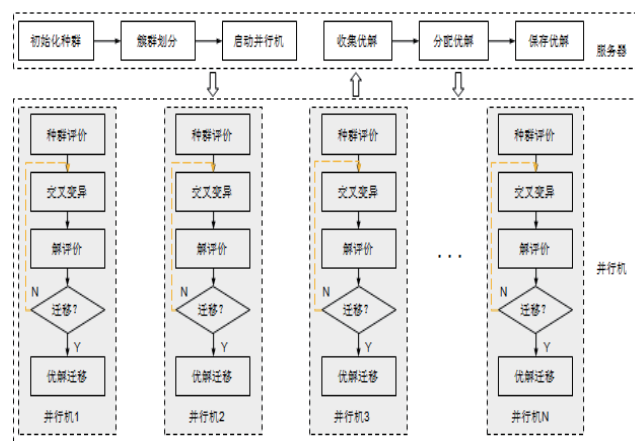


图1 算法运行框架

1. 大连外国语学院软件学院 辽宁大连 116044
2. 大连外国语学院一带一路城市与区域发展研究院
辽宁大连 116044

[基金项目] 辽宁省高等学校基本科研课题项目“基于异构图知识嵌入的生物医学实体关系抽取研究”(LJKMZ20221550)

1.1 簇群划分法

在生态学中,有一种“集群现象”,指在一定的区域或环境里,各种生物按照特定的规律结合在一起的生物系统。纵观人类的发展史,可以发现,一个部落大部分人的姓氏是

一样的,说明部落是一个“集群”组织。正是在研究集群现象的基础上,经济学中提出了产业(集)群的思想并得到成功运用,产业集群不仅包括地理聚集,还包含产业特性。

目前,应用比较广泛的聚类分析,其目的就是发现样本点之间的“抱团”性质,这种“抱团”也是集群的一种体现。集群和聚类都可以用一个统一概念簇群来描述。

定义1:簇群E是依据一定规则来划分的具有相同特征的若干个体的集合。

换言之,簇群是一个子群,这个子群中的每个个体具有相同的特征。同时,簇群根据分类规则不同,可以产生不同的簇群。

簇群数量将对算法的收敛结果和算法的收敛速度产生重要影响,如果簇群很多,那么势必会显著增加彼此间的通信开销,反之,很可能降低算法的收敛效果。在综合作业调度中,工件数量不会太多,因此本文的算法按照如下规则进行自动划分簇群:

$$X = \begin{cases} J & J > 1 \\ M & M = 1 \end{cases} \quad (1)$$

式中: X 是簇群数, J 是工件数, M 是工件下的装配体数。

为了描述簇群划分算法先作如下假设。

设 C 个簇群分别为 $c_1, c_2, \dots, c_X$, 根据公式(1)可知, $c_i(i=1, 2, \dots, X)$ 与工件 J_i 或者装配体 M_i 对应, 为了表示算法, 都用工件 J_i 表示。

设每个工件 J_i 都有 O_i 个工序, 每个簇群中的初始个体数量为 Pop_Num , 则簇群 c_i 的生成算法描述如下。

Step1: 生成 O_i 个 J_i , 即首先生成簇群 c_i 所对应工件的工序番号;

Step2: 生成 O_k 个 $J_k(k=1, 2, \dots, X, k \neq i)$;

Step3: 重复 Step2 直至所有的工件都处理完毕;

Step4: 打乱个体中除第1个基因外的基因值顺序;

Step5: 重复 Step1 ~ Step4, 直至生成 Pop_Num 个个体结束。

1.2 动态簇群调整策略

每个簇群内进化初期寻优的速度会很快,但是到了中后期,由于遗传算法的缺陷,很容易陷入局部最优解而徘徊不前。为了克服这个问题,就需要在进化过程中不断地监控簇群并适时调整簇群,以在保留簇群特征的同时提高簇群的多样性。本文使用簇群染色体的带权海明距离计算染色体的相似度,并进一步使用带权海明距离来评价簇群的多样性。

假设基于工序编码染色体的长度为 L , 共有 M 个工件, 第 k 个领域的种群规模为 N , 领域的权值为 ω_k 。

定义2(带权海明距离) 设染色体 $A_1=(a_1^1, a_1^2, \dots, a_1^L)$ 和 $A_2=(a_2^1, a_2^2, \dots, a_2^L)$, $a_i^j \in (1, M)$, $(1, M)$ 则染色体 A_1 和染色体 A_2

间的海明距离定义为 $H(A_1, A_2) = \omega_k \sum_{i=1}^L \frac{|a_i^1 - a_i^2|}{M}$ 。

定义3(染色体相似) 设簇群中有2条染色体 A_1 和 A_2 的适应度值分别为 f_1 和 f_2 , 若 $H(A_1, A_2) \leq C_1$ 且 $|f_1 - f_2| \leq C_2$, 其中 C_1 和 C_2 是2个常数, 则这2条染色体相似, 记作 $A_1 \approx A_2$, 否则记作 $A_1 \neq A_2$ 。

定义4(簇群多样) 对于簇群中的任意2条染色体 A_1 和 A_2 , 定义 $p = \begin{cases} 0 & A_1 \neq A_2 \\ 1 & A_1 \approx A_2 \end{cases}$ 。簇群的多样程度用 $S_k = \frac{1}{N!} \sum_{m=1}^{N!} p$ 表示,

若 $S_k \leq C_3$, 其中 C_3 是常数, 则表示该簇群具有多样性。

簇群的调整分为两部分: 一部分是服务机分配给该簇群的优解, 使用优解对簇群进行一定的调整, 称之为簇群间调整; 另一部分是根据簇群多样性程度自动调整, 称之为簇群内调整。

假设该簇群接收到 n 个优解, 记为 $y_i(i=1, 2, \dots, n)$, 而簇群内个体记为 $p_i(i=1, 2, \dots, N)$ 。

簇群间调整流程为:

Step1: 是否接收到优解, 若是则转 Step2, 否则继续簇群内进化;

Step2: 优解 y_i 是否来自于自身簇群, 若是则不予调整; 否则转 Step3;

Step3: 若 $y_i \approx p_i(i=1, 2, \dots, N)$, 则不予调整, 否则转 Step4;

Step4: 将 y_i 的第1个基因与第1个基因值 $=k$ 的基因交换, 改变后的染色体记为 y_i' , 用 y_i' 替换 p_i 。

Step5: 重复 Step2、Step3 和 Step4, 直至所有的优解都处理完毕。

簇群内调整流程为如下几步。

Step1: 计算群体的多样性, 若群体不具有多样性, 则转 Step2, 否则继续簇群内进化;

Step2: 若 $p_i \approx p_{i-1}(i=N, N-1, \dots, N/3)$ 则重新生成一条第1个基因值 $=k$ 的染色体, 记作 p_i' , 用 p_i' 替换 p_i 。

Step3: 重复 Step2, 直至 $p_i \neq p_{i-1}(i=N, N-1, \dots, N/3)$ 或 $i < N/3$ 。

2 仿真与结果分析

本文算法在 VS2015 上实现, 结合 RCP 技术和 TCP/IP 协议实现并行机制。硬件采用4台电脑, 一台为服务器, 另三台作为并行机。

2.1 理论仿真

本文在理论上采用 FT10 对算法进行验证, FT10 问题有10个工件, 每个工件具有10道工序, 由于实验设备有限, 实验参数设置如表1所示。

表 1 参数设置

No.	Range	PopSize	P_c	P_m
1	1 ~ 4	40	0.4	0.05
2	5 ~ 7	30	0.4	0.05
3	8 ~ 10	30	0.4	0.05

其中，Range 表示领域范围，PopSize 是种群规模， p_c 交叉概率， $p_c \in [0.1, 1]$ ， p_m 是变异概率， $p_m \in [0.01, 0.1]$ 。

PGA 与本文算法的比较如图 2 所示。由图 2 可知，本文算法在第 50 代左右解趋于稳定，而 PGA 一直处于波动状态。本文算法和 PGA 都在 45 代左右找到了最优解。所以本文算法不仅保证了寻找解的速度，而且提高了解的稳定性。

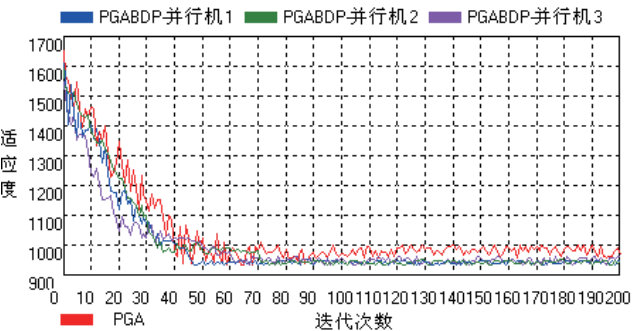


图 2 PGA 与 PGABDP 的比较

2.2 实际仿真

2.2.1 数据及参数配置说明

本文实际仿真部分使用 3 个工件，每个工件下由若干的零件和配件组成，零件由若干道工序组成。3 个工件的组成结构如图 3 所示。其中，方框内的数字表示工程编号（工程可能是零件的一道工序，也可能是一个完整配件），结点的连接关系表示工程的前后约束关系，如 102 与 101、103 的关系就是 102 的前驱工程是 101，后继工程是 103；101、102、103 表示了一个零件的 3 道工序。

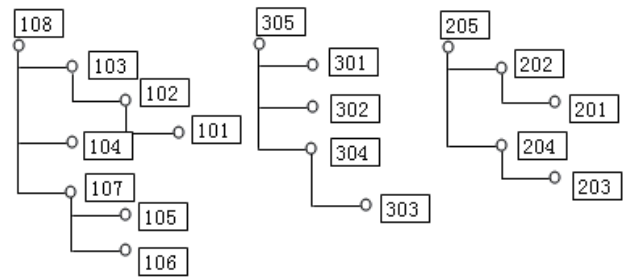


图 3 工件结构

每个工程主要由加工或装配所在的部门、需要的数量、完成 1 个需要的工作时间和完成 1 个需要的工作量组成。每个工程的详细信息如表 2 所示。工程信息中涉及的部门是一种虚拟部门，部门中只有人或机械。而实际部门中不仅有机

械，而且有人，所以算法在进行实际验证时，首先将部门拆

解为虚拟部门。上述工程信息中涉及的部门初始信息如表 3 所示，休息日信息如表 4 所示。

表 2 工程信息

编号	种别	内外区分	人部门编码	机械部门编码	数量	时间 / 天	人负荷 / (人·天)	机械负荷 / (台·天)
101	机械	加工	A110A	M100LV	1	3	9	3
102	机械	加工	A110A	M100LV	1	2	4	2
103	机械	加工	A110A	M100LV	1	3	12	3
104	人	外购	—	—	2	5	25	—
105	人	装配	A120A	—	2	2	20	—
106	人	装配	A110A	—	2	4	24	—
107	人	装配	A110A	—	2	5	20	—
108	人	装配	A110A	—	1	6	18	—
201	机械	加工	A110A	M100LV	1	5	25	5
202	人	装配	A120A	—	1	3	15	—
203	机械	加工	A110A	M100LV	1	6	24	6
204	人	装配	A120A	—	1	5	5	—
205	人	装配	A120A	—	1	3	12	—
301	人	装配	A120A	—	3	3	15	—
302	人	装配	A120A	—	1	4	28	—
303	机械	加工	A110A	M100LV	1	5	10	5
304	机械	加工	A110A	M100LV	1	5	10	5
305	人	装配	A120A	—	1	4	24	—

表 3 部门初始信息

部门编码	部门种别	工作时间	单价	能力
M100LV	机械	1	5	5
A110A	人	1	10	10
A120A	人	1	10	10

表 4 休息日信息

年	月	休息日
2021	9	4,5,11,12,18,19,25,26
2021	10	1,2,3,9,10,16,17,23,24,30,31
2021	11	6,7,13,14,20,21,27,28

部门的初始信息主要由部门编码、工作时间、单价和能力组成。其中，部门编码与工程信息中的部门编码保持一致；工作时间是指 1 天的工作时间，它依据算法实际设定的时间单位而进行改变，譬如时间单位若是天，工作时间就是 1，时间单位若是小时，则工作时间就是 8；单价是指一个人或一台机械做单位时间的工作量所需要支付的费用；能力是指这个部门拥有的人数或机械数。

算法启动所必需的参数为排产方式。其中，排产方式 1 是指排产方式的基准，共有 2 种方式：（1）顺推：根据一个产品的出图日期或设定日期，顺着 0 时间往后逐步得出工程的开工时间和完工时间；（2）倒推：根据一个产品的交货期，逆着时间逐步往前得出工程的完工时间和开工时间。

排产方式 2 是指在排产方式 1 下各个部门的能力分配

规则,共有3种方式:(1)均分:每个时间单位干的工作量尽量保持一致;(2)变动:每个时间单位干的工作量在一个设定的范围内变化;(3)满能力:尽量占用每个时间单位的能力。

本文分别采用顺推和均分方式进行。

2.2.2 求解结果

通过本文算法进行求解,其中一条近优解的染色体为:2,2,2,2,1,3,1,3,2,1,3,3,1,3,1,1,1,1,对应的加工装配顺序为201,202,203,204,101,301,102,302,205,103,303,304,104,305,105,106,107,108。相应能力负荷甘特图如图5所示。通过图4和图5可以看出,本文算法在解决加工和装配并存的实际日程展开问题上是有有效的。

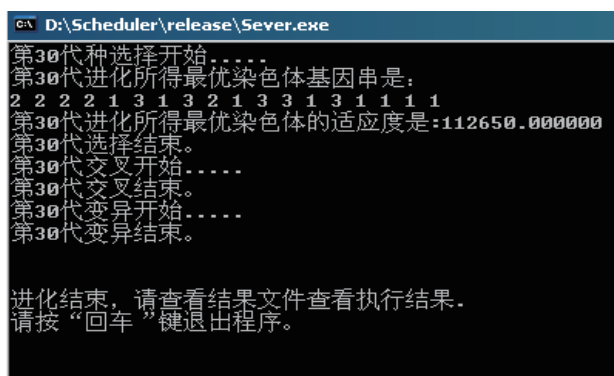


图4 实际问题寻优过程

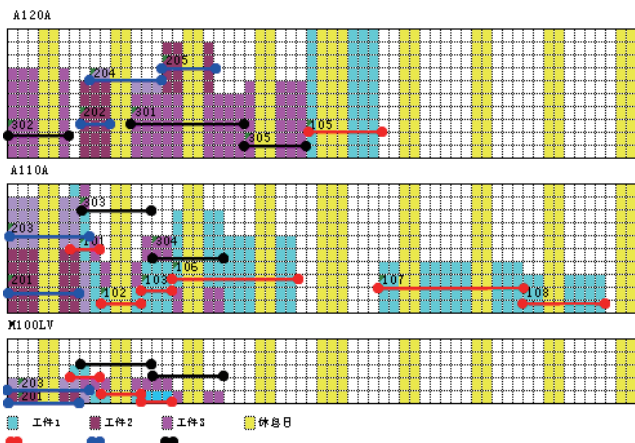


图5 优解对应能力负荷甘特图

3 结论

本文针对装配和加工同时存在的综合作业调度问题,在分析“集群”现象和遗传算法的编码规律的基础上,提出了簇群并行遗传算法(PGABDP)。该算法生成初始种群后,依据簇群原理,将相似的种群分配在同一台并行机上,以提高收敛速度。同时,不断将每台并行上的优解收集到服务器上,然后进行二次分配,以促使各并行机进化,避免过早收敛。

最后使用理论数据和实际数据对算法进行了仿真验证,实验结果表明,本文算法能够快速找到最优解且有利于提高解的稳定性。下一步将着重研究比较有效的分类规则,以期达到更好的效果。

参考文献:

- [1] 王凌. 车间调度及其遗传算法[M]. 北京:清华大学出版社,2003.
- [2] 何婷婷,戴文华,焦翠珍. 基于混合并行遗传算法的文本聚类研究[J]. 中文信息学报,2007,21(4):55-60.
- [3] KRZYSZTOF P. Hybrid multi-evolutionary algorithm to solve optimization problems[J]. Applied artificial intelligence, 2020, 34(7): 550-563.
- [4] DMITRI I A, DI W, TAO W, et al. A parallel genetic algorithm framework for transportation planning and logistics management[J]. IEEE access, 2020,8:106506-106515.
- [5] 王成栋,朱永生,张优云. 自适应伪并行遗传算法及其性能分析[J]. 小型微型计算机系统,2004,19(3):65-69.
- [6] 王玉芳,缪昇,马铭阳,等. 改进混合遗传算法的作业车间调度研究[J]. 现代制造工程,2021(5):32-38.
- [7] 王林平,贾振元,王福吉,等. 综合作业调度问题基于可行域的遗传算法[J]. 大连理工大学学报,2011,51(2):205-209.
- [8] KUANG T, HU Z, XU M. A genetic optimization algorithm based on adaptive dimensionality reduction[J]. Mathematical problems in engineering, 2020:8598543.
- [9] AHMED A A, OCTAVIA A D, ANA G A, et al. Task scheduling for mobile edge computing using genetic algorithm and conflict graphs[J]. IEEE transactions on vehicular technology, 2020, 69(8): 8805-8819.
- [10] VAHID J, VAHAB N, MOHAMMAD H R. Analysis of parallel genetic algorithm and parallel particle swarm optimization algorithm UAV path planning on controller area network[J]. Journal of control, automation and electrical systems, 2020, 31:129-140.

【作者简介】

隗昊(1993—),男,山东济南人,讲师,博士,研究方向:自然语言处理、群智能优化算法。

李正光(1980—),通信作者(email: lizhengguang2004@163.com),男,四川资阳人,讲师,博士,研究方向:文本挖掘、群智能优化算法。

(收稿日期:2024-01-18)