遗传算法在任务调度中的应用

吕 收 汪晓雨 冷 洁
海军航空大学青岛校区 山东 青岛 266041

摘 要：随着工业生产的发展，合理利用资源配置，统筹安排工作进度，优化实现任务调度计划，提高设备的利用率已变成多任务调度重点关注问题。近年来遗传算法在解决任务调度问题中得到了很大的发展。本文介绍了遗传算法的基本原理和任务调度的数学模型，详细阐述了遗传算法在任务调度中的编码策略、适应度函数构造及遗传操作实现等方法，并通过改进算法中的交叉和变异操作仿真分析了该算法的有效性。

关键词：遗传算法；任务调度；优化

1 遗传算法基本原理

遗传算法是一种模拟生物学基因遗传的优化算法，其起源可追溯到上世纪六七十年代[1]。经过几十年的发展，遗传算法发展成解决复杂优化问题的主要方法。在任务调度领域，遗传算法的应用尤为突出，主要思想是通过仿照生物进化过程中的遗传、突变、自然选择和交叉配对等操作，来搜索问题的最优解[2, 3]。在具体操作中，遗传算法的初始种群可以通过随机生成一组初始解获取；然后通过适应度函数评估每个个体的优劣，并根据评估结果进行选择、交叉和变异操作，生成新的种群；如此迭代进行，直至满足终止条件，得到问题的最优或近似最优解，如图1所示。

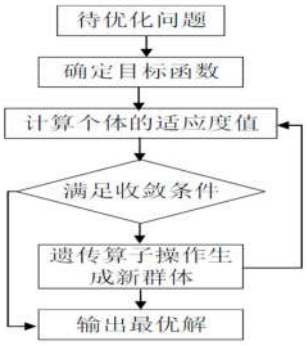

图1 遗传算法原理

遗传算法中问题的解是用有序数字串表示的，遗传算子也是直接对有序数字串进行运算的。遗传运算是仿照生物基因的操作，其任务是根据个体的适应度对其进行一定的操作，进而达到进化论中的适者生存。从优化的角度来分析，遗传运算可以使问题的解通过不断迭代优化，从而逼近最优解。遗传运算包括以下三种遗传基本操作:选择，交叉，变异。选择和交叉基本完成了大部分遗传算法的寻优功能，变异使遗传算法尽量避免陷入局部最优解。

1.1 选择

选择是就是指从群体中进行优胜劣汰的操作[4]。它是基于适应度评估的。适应度越强个体越有可能被选中，其“子孙”在下一代中出现的概率就越大，选择出来的个体就会被放进一个配对库里。目前比较常用的选择方法有最佳个人保留法，期望值法，轮赌盘法，排序选择法，线性标准化法，竞争法等，选择方法应该根据问题的具体情况而选择。

1.2 交叉

交叉是指将两个父代个体的部分结构替换重组而产生新个体，交叉的目的是产生下一代的新个体，遗传算法的搜索能力通过交叉操作可得到较大提升[5, 6]。交叉操作是根据一定的交叉概率在匹配库中随机选择两个个体进行的，交叉位置也是随机的，可以通过一定的方法或原则随机选取位置，交叉概率非常重要，其值影响着问题优化的效果，不宜取太小，其范围为0.6-0.9。

1.3 变异

变异是以极小的变异概率，随机改变种群中个体的有序数字串的值，变异操作的基本过程是:在[0，1]之间产生一个随机数，若随机数小于变异概率，则会执行变异操作。遗传算法中选择和交叉操作可能会造成某一局部排序永久丢失，而通过变异操作与选择、交叉操作相结合，可以保证了传算法的有效性，在变异这一操作步骤中，变异概率的选取非常重要，其值不宜取得过大，一般为0-0.3。

2 任务调度问题的数学模型

假设有n个工件，要在m台设备上操作，每个工件有p_i道工序，则所有工序的集合：

$$P=\begin{pmatrix} \overbrace{P_{j_1 1}P_{j_1 2}\cdots P_{j_1 P_1}}^{P_1} & \overbrace{00\cdots 0}^{P_i-P_1} \\ \vdots & \vdots \\ \underbrace{P_{j_n 1}P_{j_n 2}\cdots P_{j_n P_1}P_{j_n (P_i+1)}}_{P_i+1} & \underbrace{0\cdots 0}_{P_i-P_1-1} \end{pmatrix} \quad (1)$$

其中$J=\{J_1,J_2,\cdots J_n\}$为所有工件的集合；$M=\{M_1,M_2,\cdots M_m\}$为所有设备的集合；$P_{J_i}=\{P_{J_i1},P_{J_i2},\cdots P_{J_iP_i}\}$为工件$J_i$的工序集合；$P$为$n\times\max\{P_1,P_2,\cdots P_n\}$矩阵。$P(i,j)$表示$i$工件的第$j$道工序。$P(i,\bullet)=P_{J_i}$为$i$工件的所有工序按优先顺序的排列。不足$\max\{P_1,P_2,\cdots P_n\}$，那么空出的位置用0补充。

设备顺序阵：

$$J_M=\begin{pmatrix} \overbrace{M_{P_{J_i1}}M_{P_{J_i2}}\cdots M_{P_{J_iP_i}}}^{P_i} & \overbrace{00\cdots0}^{P_i-P_i} \\ \vdots & \vdots \\ \underbrace{M_{P_{J_n1}}M_{P_{J_n2}}\cdots M_{P_{J_nP_i}}M_{P_{J_n(P_i+1)}}}_{P_i+1} & \underbrace{0\cdots0}_{P_i-P_i-1} \end{pmatrix} \quad (2)$$

$J_M(i,\ j)$表示i工件的第j道工序的设备号，$J_M(i,\bullet)$表示i工件的所有工序按优先顺序加工的各设备号的排列。J_M为$n\times\max\{P_1,P_2,\cdots P_n\}$矩阵。如果某工件的工序数不足$\max\{P_1,P_2,\cdots P_n\}$，那么空出的位置用0补充。

加工时间阵：

$$T=\begin{pmatrix} \overbrace{T_{P_{J_i1}}T_{P_{J_i2}}\cdots T_{P_{J_iP_i}}}^{P_i} & \overbrace{00\cdots0}^{P_i-P_i} \\ \vdots & \vdots \\ \underbrace{T_{P_{J_n1}}T_{P_{J_n2}}\cdots T_{P_{J_nP_i}}T_{P_{J_n(P_i+1)}}}_{P_i+1} & \underbrace{0\cdots0}_{P_i-P_i-1} \end{pmatrix} \quad (3)$$

$T(i,\ j)$表示工件i的第j道工序在$J_M(i,\ j)$上的加工时间。T为$n\times\max\{P_1,P_2,\cdots P_n\}$矩阵。如果某工件的工序数不足$\max\{P_1,P_2,\cdots P_n\}$，那么空出的位置用0补充。

任务调度问题有多种优化目标或综合优化目标，任务调度问题的优化目标通常考虑成本和时间两个方面，若从成本考虑，可从存货量、在制品量、设备利用率等方面优化;从时间上考虑，可从最大限度满足交货期、最小完成时间、最小流转时间、最小等待时间等方面考虑[7]。

3 遗传算法在任务调度中的应用

3.1 编码设计

在任务调度中，遗传算法的应用策略至关重要。其中，编码策略设计是首要环节，它直接关系到算法的性能和效率[8]。在选择编码方式时，需充分考虑任务调度的特性和问题规模。编码长度与问题规模的关联密切，通常问题规模越大，所需的编码长度也越长。合理的编码设计能够确保遗传算法在任务调度中的有效应用，为求解复杂调度问题提供有力支持。基于操作的编码将对工件的所有操作进行编码，对工件用不同数字进行编码表示，同一工件的不同操作在染色体上按照操作顺序则用相同的数字编码，数字编码在染色体上的出现的次数表示该编号工件的该次数的操作。

表1 加工时间和工艺约束

项目	工件	操作序列		
		1	2	3
操作时间	J_1	3	3	2
	J_2	1	5	3
	J_3	3	2	3
设备	J_1	M_1	M_2	M_3
	J_2	M_1	M_3	M_2
	J_3	M_2	M_1	M_3

对于3个工件在3个设备上加工的案例，假设染色体为{2 1 2 3 1 1 3 2 3}，O_{ijk}表示第i个工件的第j个工序在第k个设备上加工，该染色体对应的有序操作表为[O_{211} O_{111} O_{223} O_{312} O_{122} O_{133} O_{321} O_{232} O_{333}]，可得出先安排第2个工件的第1个操作，然后安排第1个工件的第1个操作，第2项任务的第2个操作步骤，…以次类推，进而相应的调度如图2所示。

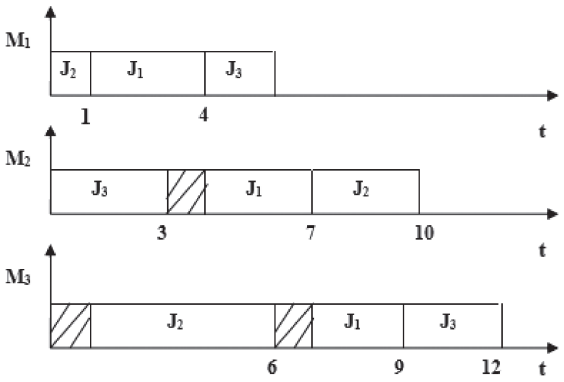

图2 甘特图

3.2 适应度函数设计

在遗传算法应用于任务调度时，适应度函数的构造至关重要。它不仅是评估个体优劣的依据，更是引导算法搜索方向的关键[9]。设计适应度函数时，应遵循明确性、合理性和高效性的原则。典型的适应度函数示例包括基于任务完成时间的函数，如总完成时间最小化函数，以及基于系统负载均衡的函数，如负载均衡度最大化函数。这些函数在具体应用中需结合任务特性和系统环境进行细致分析和调整。

要计算出多个任务的最短工期，首先要对染色体进行解码，计算所有工作任务的最早完工时间，得出目标函数公式（4）：

$$\min\sum_{i=1}^{n}T_{P_{ji(Pi+1)}} \quad (4)$$

3.3 仿真分析

为验证遗传算法在任务调度中应用的有效性，以某一案例验证遗传算法在任务调度中的有效性，该案例的工件、工序、所需时间如表2所示。

表2 案例

工序 时间/min 工件	工序1	工序2	工序3	工序4	工序5
1	6	5	8	16	
2	9	16	10		
3	5	9	7	4	12

对传统遗传算法进行改进，在进行交叉操作时，随机选择某一工件编号，分别在两条染色体中，保留该编号的位置，将除该编号以外的片段按先后排序依次交换复制到剩余的基因位。在进行变异操作时，随机选择染色体，在染色体中随机选择两个基因位，首先判断这两个基因位上的值是否相等，如果相等，则随机重选，直到选到值不相等的两个基因位，然后将两个基因位上的值互换，得到变异后的新染色体。交叉概率设为0.7，变异概率设为0.1，变异概率终止条件条件设置为迭代次数为100进行仿真分析，仿真结果如图3、图4所示。

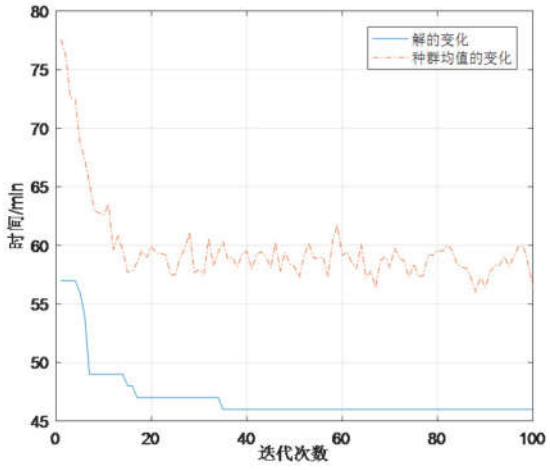

图3 最优解和种群均值的变化

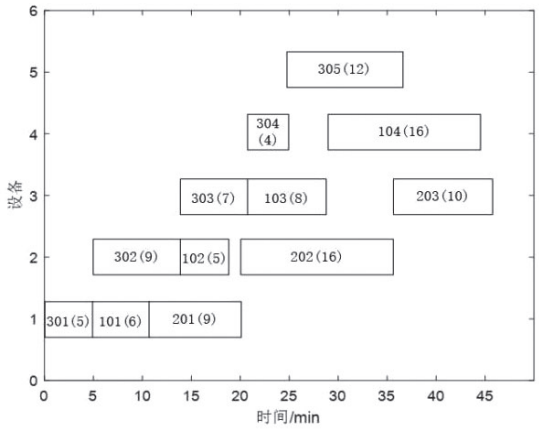

图4 甘特图

由图3可知，在最初阶段，任务完成所需时间随迭代次数的增加而减少；在迭代次数为36时，任务完成所需时间达到最小46min；在此之后，随着迭代次数的增加任务完成所需时间不再变化。种群的任务完成所需时间均值一直高于最优解所需时间，其也会随迭代次数的增加而减少，在迭代次数为36时会在60min上下振荡。按照同一参数对系统进行多次仿真，每次在迭代次数达到一定数值时最优解所需时间会固定为某一值，但是获得的最优解所需时间是不同的，因此遗传算法容易陷入局部最优解，不易获得全局最优解。图4为某次仿真分析获得的局部最优解，因此按照此甘特图计划的任务排序不是耗时最短的排序。

4 总结

本文对遗传算法在任务调度中的应用进行了分析和讨论，遗传算法能够求解任务中多任务优化问题，从而提高资源利用率，减少完成任务所需时间，另外遗传算法相对比较灵活，只需要分别输入任务的各种参数信息，比如各工作所需时间、人力设备资源配置等，就能对各个任务进行调度分析优化。但是遗传算法同样存在缺点，不易获得全局的最优解，而容易陷入局部最优解。

参考文献

[1]遗传算法的基本理论与应用[M].李敏强等著.科学出版社.2002.

[2]改进的自适应遗传算法的研究与应用[D].张大科.昆明理工大学,2019.

[3]改进自适应遗传算法在柔性车间调度的应用[J].王豪渊;刘文涛.自动化应用. 2023, 64 (17).

[4]一种新型自适应遗传算法在多峰函数优化中的应用[J].张大科;钱谦.软件导刊,2018(06).

[5]遗传算法在多目标优化中的应用研究[D].张安英.辽宁工程技术大学,2008.

[6]一种基于改进遗传算法的柔性车间调度方案[J].叶彦斐;童先洲;刘之境.国外电子测量技术,2020(09).

[7]基于遗传算法的柔性车间调度优化[J].郭庆;张明路;孙立新;刘轩.科学技术与工程,2020(29).

[8]柔性制造车间多载具AGV节能调度[J].吴立辉;顾天鑫;石津铭.应用技术学报,2024 ,24 (04)

[9]自适应遗传算法与邻域搜索的柔性作业车间调度方法[J].叶孝生;张舒曼;唐天誉等.智能计算机与应用,2024