计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190900123

求解柔性资源受限项目调度问题的多种群遗传算法

姚 敏
大同煤炭职业技术学院信息工程系 山西 大同 037003

摘 要 柔性资源普遍存在于制造业生产制造的各个环节中,提高了资源利用率和生产效益。以柔性资源为研究对象,建立了以最小化项目完成工期为目标的柔性资源受限项目调度问题的数学模型。针对现有标准遗传算法过早地收敛从而使整个遗传搜索无法求解出全局最优值的缺陷,提出了一种改进的多种群遗传算法来求解该问题模型。算法对作业优先级列表编码,引入交叉移民算子实现多种群间的协同进化,在解码过程中运用一种启发式柔性资源技能分配算法为作业分配资源,同时通过改进的串行调度生成方案对作业调度。最后通过标准算例库 PSPLIB 进行数值试验,验证了所提算法求解该问题的有效性。

关键词:柔性资源;资源受限;项目调度;多种群遗传算法
中图法分类号 TP29;F273;TP31

Multi-population Genetic Algorithm for Multi-skill Resource-constrained Project Scheduling Problem

YAO Min
Department of Information Engineering, Datong Vocational and Technical College of Coal, Datong, Shanxi 037003, China

Abstract Multi-skill resource resource is popularly existing in the process of production and manufacturing, which improves resource utilization and production efficiency. This paper makes the multi-skill resource as the object and presents a mathematical model of multi-skill resource-constrained project scheduling problem (MSRCPSP) with the objective of minimizing the makespan of the project. In order to solve the disadvantage that the existing genetic algorithm converges prematurely and the whole algorithm cannot solve the global optimal value, an improved multi-population genetic algorithm is proposed to solve the problem model. The algorithm is designed to code job priority list and introduced the cross immigration operator to promote the co-evolution of different groups. In the decoding process, a heuristic flexible resource skill allocation algorithm is used to allocate resources for jobs, and an improved serial scheduling generation scheme is used to schedule jobs. Finally, numerical experiments are carried out with standard case library PSPLIB to verify the effectiveness of the proposed algorithm in solving MSRCPSP.

Keywords Multi-skill resource, Resource constrained, Project scheduling, Multi-population genetic algorithm

1 引言

项目调度是项目管理中最重要的一个研究课题。项目调度是在不违反现有的任何约束如时序优先级约束、资源约束等的情况下,为组成项目的所有作业确定一个最优的执行时间。近些年来,由于全球经济发展迅速,科技日益进步,许多公司和组织需要经常处理越来越大和越来越复杂的项目,因此项目调度的重要性日益显现。资源受限项目调度问题(Resource Constrained Project Scheduling Problem, RCPSP)是项目调度问题中一个最为基础和广泛的研究课题,即不仅各个作业满足给定的时序优先级关系,而且各个作业的执行所需的可再生资源的数量有固定的限制,目标为最小化项目工期[1]。在 RCPSP 中,通常假设每个资源都具有单一的能力,即掌握单一的技能。然而,在许多实际应用程序如汽车生产线等中,资源是多技能的,如人力资源或多用途机器等,即每个资源掌握一种或几种技能,同时每项作业执行所需的每项技能可能需要几个资源共同完成。如图 1 所示,在汽车移动装配生产线上,工人 C 可以执行作业 3-1 和 4-1,作业 3-1 的技能 2 需要 3 个单位资源执行,技能 3 需要 2 个单位资源执行。在这种情况下,整个项目的调度更加复杂困难,RCPSP 的这种扩展称为多技能资源约束项目调度问题(Multi-Skill Resource Constrained Project Scheduling Problem, MSRCPSP)。与经典 RCPSP 相比,MSRCPSP 不仅需要决策各个作业的调度方案,还要决策出分配给每个作业的资源种类和数量以及所分配的资源需要执行的技能[2]。

Néron[3] 证明了 MSRCPSP 是个 NP-Hard 问题,并通过基于线性规划和能量推理的两种方法求出了 MSRCPSP 的下界。Bellenguez-Morineau[4] 考虑了柔性资源存在不可使用时间段的情况,并设计了相应的分支定界算法、启发式算法和元启发式算法求解该问题。Li 等[5] 假设每个活动在执行过程中的每个技能只需要一个资源,目标是在满足项目完成的预定期限下最小化与资源相关的总成本。Jia 等[6] 研究了一种有效的柔性资源替代方法,并通过改进的并行调度启发式算法求解资源间可相互替换的 MSRCPSP 问题。Zhang 等[7] 提出了一种改进的遗传算法,即基于局部两作业资源需求的改进串行调度机制的遗传算法。Xie 等[8] 以项目工期和成本为多优化目标,建立资源可用量可变约束下的多模式柔性资源项目调度问题的数学模型,并提出一种基于非支配排序遗传算法的双目标混合遗传算法来求解该问题。Heimerl 等[9] 提出柔性资源存在不同技能层次结构,即异质效率的柔性资源。

本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
通信作者:姚敏(yaomindali@163.com)

对于各个作业而言,其执行时间会受到分配的资源种类和执行技能层次级别的影响。Liu 等[10]针对柔性资源存在时间窗且作业可中断的项目调度问题,建立了相应的整数规划模型,设计了一种分支定界算法来构造搜索树进行求解,并提出了两个有效的剪枝规则。现有文献在求解 MSRCPSP 问题时所设计的算法主要分为精确算法、启发式算法、元启发式算法。其中,精确算法无法求解大规模算例问题,启发式算法由于受启发式规则的限制,并不能针对所有算例求得较优的解。因此,元启发式算法成为了求解 MSRCPSP 问题的主流算法[11],其中遗传算法(Genetic Algorithm,GA)应用最为广泛[2,9,10,12-14]。然而标准的遗传算法由于受到交叉率、变异率、种群规模和迭代次数等的影响,为使得群体过早地收敛到同一非最优状态而使整个遗传搜索无法求解出全局最优值。针对遗传算法的缺点,学者们提出了多种群遗传算法(Multi-population Genetic Algorithm,MPGA)[15]。Cochran 等[16]通过研究多目标并行机调度问题,提出了一种两阶段多种群遗传算法,每个亚群体分别进化,通过精英策略保留每个目标的最佳个体和组合目标的最佳个体。Jiao 等[17]根据货位分配目标函数和货架特点,设计了整数编码方式的多种群遗传算法,并通过遗传算子实现了种群间的协同进化。Lia 等[18]针对多目标模糊柔性车间调度问题,构建了对应的数学模型,并提出了纵横协同的多种群遗传算法,各种群之间通过相互竞争实现优秀个体的迁移共享。Wang 等[19]研究了一种结合模拟退火算法的混合多种群遗传算法,进一步解决了多种群遗传算法进化过程中丧失种群多样性而导致早熟等的问题。

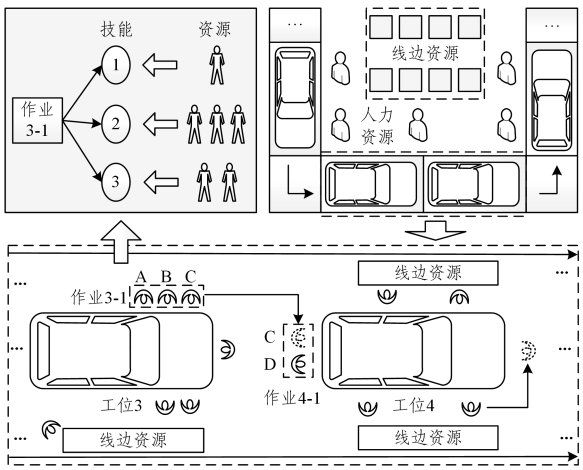


图 1　汽车移动装配生产线简图

Fig. 1　Diagram of automobile moving assembly line

综上所述,本文以汽车移动装配等实际工业生产为研究背景,结合生产中存在的多技能资源情况,提出了柔性资源受限项目调度问题并建立了对应的数学模型。针对该问题模型,设计了一种以作业优先级列表编码的改进多种群遗传算法。算法中引入交叉移民算子,促进种群间信息交流和协同进化。在解码过程中运用一种启发式柔性资源技能分配算法为作业分配资源,同时通过改进的串行调度生成方案对作业进行调度,从而完成整个项目作业的调度计划。

2　问题描述及数学模型

定义任意一个项目由没有闭合环路的节点网络图 $G=(\boldsymbol{V},\boldsymbol{E})$ 表示,其中 $\boldsymbol{V}=\{0,1,\cdots,i,j,\cdots,n+1\}$ 代表作业集合,$\boldsymbol{E}$ 代表作业之间的时序关系。作业 0 和作业 $n+1$ 为虚拟作业,分别代表该项目的开始和结束,两作业不消耗任何资源和执行时间,其余各个非虚拟作业的执行时间为 t_j,$j\in\boldsymbol{V}$。对时间进行离散化处理,时间集合为 $\boldsymbol{D}=\{1,\cdots,d,\cdots,D\}$,$d\in\boldsymbol{D}$。项目需要的技能集合为 $\boldsymbol{S}=\{1,\cdots,s,\cdots,S\}$,$s\in\boldsymbol{S}$,这些技能由资源集合 $\boldsymbol{R}=\{1,\cdots,k,\cdots,K\}$,$k\in\boldsymbol{R}$ 提供执行。作业对技能的需求集合为 $\boldsymbol{VS}=\{r_{js}\mid j\in\boldsymbol{V},s\in\boldsymbol{S}\}$,$r_{js}$ 表示作业 j 对技能 s 的需求量。资源掌握的技能通过资源-技能集合 $\boldsymbol{RS}=\{\delta_{ks}\mid k\in\boldsymbol{R},s\in\boldsymbol{S}\}$ 表示,其中 $\delta_{ks}=1$ 表示资源 k 掌握技能 s,否则 $\delta_{ks}=0$。定义 $\boldsymbol{P}_j$ 为作业 j 的紧前作业集合,$\boldsymbol{S}_j$ 为作业 j 的紧后作业集合,t_{S_j} 和 t_{F_j} 分别表示作业 j 的开始时间和结束时间。

本文考虑以下基本假设:任意作业的所有紧前作业完成后该作业才能执行;所有作业一旦开始不可中断;资源 k 在时刻 d 只能使用一种技能 s 执行一个作业 j;资源不可抢占,即资源 k 执行完作业 j 后,才能被释放去执行另一项作业。

数学模型中的决策变量如下:

x_j^d:0,1 变量,作业 j 在 d 时刻执行则为 1;否则为 0。

y_{jk}^s:0,1 变量,资源 k 使用技能 s 执行作业 j 则为 1;否则为 0。

z_{jk}^d:0,1 变量,资源 k 在 d 时刻执行作业 j 则为 1;否则为 0。

设项目最终完成工期为 T_C,问题目标函数值为最小化项目工期,则目标函数为:

$$T_C=\min\ t_{F_{n+1}} \tag{1}$$

模型约束为:

$$\boldsymbol{RS}\neq\varnothing,\forall k\in\boldsymbol{R},\forall s\in\boldsymbol{S} \tag{2}$$

$$t_j\geqslant 0,\forall j\in\boldsymbol{V} \tag{3}$$

$$t_{F_j}\geqslant 0,\forall j\in\boldsymbol{V} \tag{4}$$

$$t_{S_j}\geqslant t_{S_i}+t_i,\forall i,j\in\boldsymbol{V}\wedge(i,j)\in\boldsymbol{E} \tag{5}$$

$$\sum_{j\in V}\sum_{k\in R}z_{jk}^d\leqslant R,\forall d\in\boldsymbol{D} \tag{6}$$

$$\sum_{j\in V}z_{jk}^d\leqslant 1,\forall k\in\boldsymbol{R},\forall d\in\boldsymbol{D} \tag{7}$$

$$\sum_{s\in S}y_{jk}^s\leqslant 1,\forall j\in\boldsymbol{V},\forall k\in\boldsymbol{R} \tag{8}$$

$$\sum_{k\in R}y_{jk}^s=r_{js},\forall j\in\boldsymbol{V},\forall s\in\boldsymbol{S} \tag{9}$$

$$y_{jk}^s\leqslant\delta_{ks},\forall j\in\boldsymbol{V},\forall k\in\boldsymbol{R},\forall s\in\boldsymbol{S} \tag{10}$$

$$t_jx_j^d-t_jx_j^{d+1}+\sum_{m=d+2}^{D}x_j^m\leqslant t_j,\forall j\in\boldsymbol{V},\forall d\in\boldsymbol{D} \tag{11}$$

$$\sum_{d\in D}x_j^d=t_j,\forall j\in\boldsymbol{V} \tag{12}$$

$$x_j^d,y_{jk}^s,z_{jk}^d\in\{0,1\},\forall j\in\boldsymbol{V},\forall k\in\boldsymbol{R},\forall d\in\boldsymbol{D},\forall s\in\boldsymbol{S} \tag{13}$$

式(1)表示目标函数值最小化项目的完工时间;式(2)确保每一种资源都至少掌握一种技能,不存在空技能情况;式(3)限定作业执行时间非负;式(4)表明任意作业在调度时间表内都有非负的完成时间;式(5)表示时序约束,即作业必须在其所有紧前作业完成后才能进行;式(6)表示资源约束,即任意时刻资源的需求量不得超过资源上限;式(7)表示每种资源在某一时刻只能执行一项作业;式(8)表示每种资源只能使用一种技能执行作业;式(9)表示要满足各个作业对技能的需求量;式(10)表示资源只有在掌握某技能时才能用该技能执行作业;式(11)表示作业一旦开始就不能中断;式(12)表示作业在给定工期内执行完;(13)定义了决策变量的取值范围。

3 算法设计

针对 MSRCPSP 问题，本文设计了一种以作业优先级列表为编码方式的改进多种群遗传算法来求解该问题模型。传统多种群遗传算法中的移民操作可能会因为外来优秀个体成为种群主导个体从而使得种群多样性迅速降低，导致无法跳出局部最优。因此，本文对传统的移民操作进行改进，对移民算子进行交叉处理，得到交叉移民算子，再通过精英策略选择较优的个体替换目标种群中的最差个体，从而达到每次交流都使得目标种群引入外来优秀个体的部分基因，又保留自身种群个体的部分基因，避免外来优秀个体迅速成为主导个体的情况发生，保证了种群的多样性。在解码过程中，针对柔性资源技能分配问题，为避免资源分配时选择的随机性，本文提出一种启发式柔性资源技能分配算法，并通过作业—技能—资源映射关系模型为作业分配相应的资源。同时调用改进的串行调度生成机制对作业进行调度，通过判断局部两作业位置关系，根据不同的情况对作业进行调度，最终得到各个作业的执行时间，使得项目完成工期最短。算法流程图如图 2 所示。

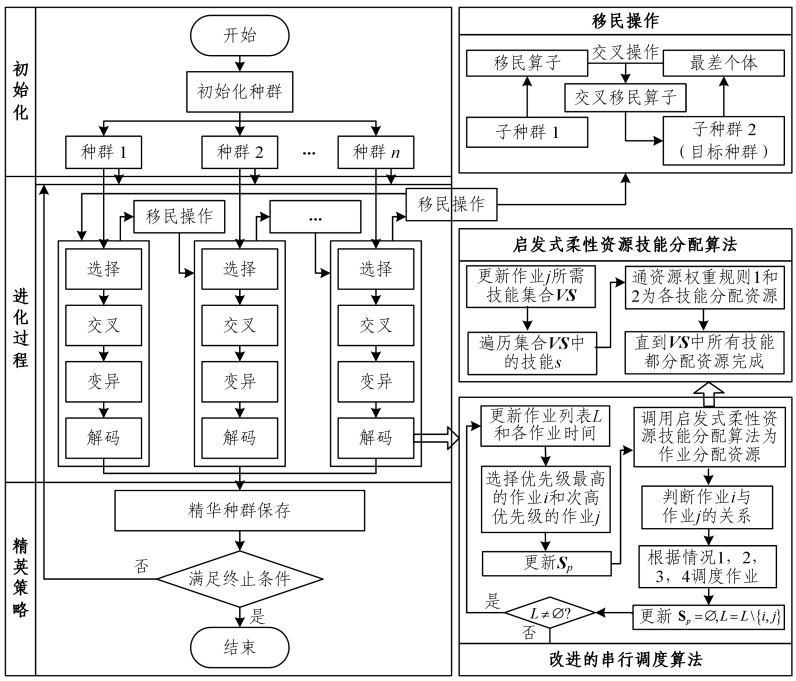


图 2 算法流程图

Fig. 2 Flow chart of algorithm

3.1 改进多种群遗传算法

根据 MSRCPSP 问题特性，选择合适的编码方式，并设计对应的选择操作、交叉操作、变异操作，移民操作和进化终止判断标准。本文提出的改进多种群遗传算法对移民操作进行改良设计，保证了移民操作后种群多样性的持续性。

3.1.1 编码

本文采用实数编码方式对作业优先级列表编码，则编码的长度等于作业的个数 $n+1$，编码的位置代表了作业的优先级 p_j，$j\in\boldsymbol{V}$，p_j 越小，优先级越高；该位置上的数字代表了作业编号。如图 3 所示，作业 2 的优先级 $p_2=1$，优先级最高；作业 7 的优先级最低，$p_7=7$；作业 3 的优先级 $p_3=5$ 等。

2	5	6	4	3	1	7

图 3 编码示意图

Fig. 3 Example of code

3.1.2 选择

MSRCPSP 的目标函数为最小化项目完成时间，因此定义染色体的适应度函数 $f(x)$ 为目标函数的倒数。本文通过轮盘赌选择染色体，即适应度函数值较大的染色体个体被选中的概率较大。设第 n 个子种群中的第 m 个染色体的适应度函数为 $f(x_{nm})$，则该个体被选中的概率为：

$$P_{nm}=f(x_{nm})/\sum_{m=1}^{posize}f(x_{nm}) \tag{14}$$

3.1.3 交叉

从经过选择操作的种群中随机选择两个染色体，通过交替位置交叉法进行交叉操作。如图 4 所示，子代 1 中首先选取父代 1 中第一个基因 2 加入到子代 1 中，然后从父代 2 中选择第一个基因 3，判断其是否与子代 1 中已有的基因重复，若不重复则加入子代 1 中；若重复则舍去，再从父代 1 中选择第二个基因 5 并判断……即轮流选择父代 1 和 2 中的基因并判断是否加入子代中，直到子代的长度达到定义的长度为止。同样，依次轮流选择父代 2 和 1 中的基因，从而得到子代 2。

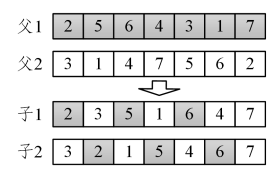


图 4 交叉操作

Fig. 4 Crossover operation

在多种群遗传算法中，不同的子种群设置不同的交叉概率。根据文献[20]中对遗传算法交叉概率的研究验证，本文设置子种群的交叉概率 $P_c=\{0.5,0.6,0.7,0.8,0.9\}$。

3.1.4 变异

本文选择交换变异法对染色体进行变异操作，即从父代个体中随机选择两个基因位置，然后互换这两个位置的基因，如图 5 所示。

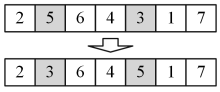


图 5　变异操作

Fig. 5　Mutation operation

考虑多种群遗传算法中不同子种群的差异性,根据文献[20]中对遗传算法交叉概率的研究验证,本文设置子种群的变异概率 $P_m=\{0.1,0.15,0.2,0.25,0.3\}$。

3.1.5　移民操作设计

多种群遗传算法通过移民算子达到各个独立子种群协同进化的目的。传统的移民操作方法用本种群中的最优个体直接替换目标种群中最差的个体。这种方式的优点是种群收敛速度快;缺点是外来优秀个体进入后经过遗传操作会迅速成为本种群的主导个体,从而导致种群的多样性迅速下降。本文提出一种改进移民操作方式(见图 6),即选出本种群中最优染色体作为移民算子,然后与目标种群中的最差染色体进行交叉操作(3.1.3 节),将此类移民算子称为交叉移民算子,选择交叉后个体中较优个体替换目标种群中最差染色体。改进的移民操作可以保证在引入外来优秀基因的同时保持种群多样性的持续性。

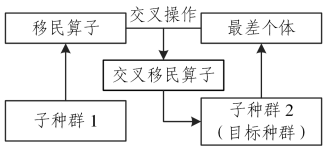


图 6　移民操作

Fig. 6　Immigration operation

3.1.6　进化终止的判断标准

在多种群用于遗传算法中,设置精华种群用于保存每次进化过程中最优的染色体个体,每经过一代遗传操作,将该代最优染色体对应目标函数值 $T_C(\lambda_{best})$ 与精华种群中对应的目标函数值 T_C^{best} 比较,如果 $T_C(\lambda_{best})<T_C^{best}$,则更新 $T_C^{best}=T_C(\lambda_{best})$;如果 $T_C(\lambda_{best})\geqslant T_C^{best}$,则保持 T_C^{best} 不变;当精华种群中最优个体最少保持 50 代时,算法结束。

3.2　启发式柔性资源技能分配算法

为避免柔性资源再分配给作业时的随机性,本文提出一个启发式柔性资源技能分配算法来求解资源分配问题。定义两个启发式规则作为资源选择依据。

定义 $\boldsymbol{Q}_k$,$k\in\boldsymbol{R}$ 表示资源 k 掌握的技能集合;$\boldsymbol{R}_s$,$s\in\boldsymbol{S}$ 表示掌握技能 s 的资源集合;$\boldsymbol{J}_s$,$s\in\boldsymbol{S}$ 表示需要技能 s 的作业集合。

资源权重规则 1:计算各个资源掌握技能的数量 $\alpha_1=|\boldsymbol{Q}_k|$,$k\in\boldsymbol{R}$,$\alpha_1$ 越大代表资源掌握的技能越多,柔性越强,则优先级越低;反之优先级越高。

资源权重规则 2:通过式(15)计算每种资源所掌握技能对整体作业所需技能影响的复杂程度 α_2,α_2 越大,则该资源影响程度越大,优先级越高;反之优先级越低。

$$\alpha_2=|\boldsymbol{Q}_k|\times\max_{s\in\boldsymbol{Q}_k}\left\{\frac{R}{|\boldsymbol{W}_s|}\times\sum_{j\in\boldsymbol{J}_s}(t_j\times r_{js})\right\},k\in\boldsymbol{R}\tag{15}$$

启发式柔性资源技能分配算法步骤如下所示:

步骤 1　选择需要分配资源的作业 j,更新作业 j 所需技能集合 $\boldsymbol{VS}$,转到步骤 2;

步骤 2　遍历集合 $\boldsymbol{VS}$,从第一种技能 s 开始,更新拥有技能 s 的资源集合 $\boldsymbol{W}_s$,转到步骤 3;

步骤 3　通过资源权重规则 1 更新各个资源 $k\in\boldsymbol{W}_s$ 的优先级,选择优先级最高的资源 k,分配给作业 j 的技能 s;如果存在多种资源的优先级相同,则通过资源权重规则 2 更新这些资源的优先级,再次选择优先级最高的资源分配,转到步骤 4;

步骤 4　给下一种技能 s' 分配资源,更新拥有技能 s' 的资源集合 $\boldsymbol{W}_{s'}$,转到步骤 3;如果所有技能均已分配完毕,则转到步骤 5;

步骤 5　算法结束。

3.3　改进的串行调度算法

传统串行进度生成机制过于积极,有时会因为提前某一作业开始时间,从而延迟其他作业的开始时间,进而导致整个项目的完成时间变长。因此,本文提出一种具有前瞻性的考虑局部两作业情况的串行调度方案,通过考虑当前调度作业对下一调度作业开始时间的影响情况,决策当前调度作业和下一调度作业的开始时间。具体情况分析如下:

定义 t_{EST_j},t_{EFT_j},t_{LST_j},t_{LFT_j},$j\in\boldsymbol{V}$ 为由关键路径法求得各个作业的最早开始时间、最早结束时间、最晚开始时间、最晚结束时间;假设作业 i 的优先级高于作业 j,即 $p_i<p_j$。

情况 1　$(i,j)\in\boldsymbol{E},i,j\in\boldsymbol{V}$。此时,两作业间有时序约束关系,且作业 j 的开始时间一定晚于作业 i 的结束时间。则此情况下按照优先级关系,在满足资源约束的情况,依次给两作业安排最早可开始时间,调度两作业即可。

情况 2　$(j,i)\in\boldsymbol{E},i,j\in\boldsymbol{V}$。两作业间存在时序约束,且 $j\in\boldsymbol{P}_i$,此时如果先调度作业 i,则作业 j 的调度区间会进一步缩短,可能导致无法满足作业 j 的资源需求。因此在该情况下,将两作业的优先级进行交换,即更新 $p_i>p_j$,再按照情况 1 调度方式对两作业进行调度。

情况 3　$(i,j)\notin\boldsymbol{E}\wedge(j,i)\notin\boldsymbol{E}\wedge t_{LFT_i}\leqslant t_{EST_j}$,$i,j\in\boldsymbol{V}$。此时两作业不存在时序约束且两作业的调度区间没有重叠区域,此情况下,两作业调度结果互不影响,则按照优先级关系依次调度两作业。

情况 4　$(i,j)\notin\boldsymbol{E}\wedge(j,i)\notin\boldsymbol{E}\wedge t_{LFT_i}>t_{EST_j}$,$i,j\in V$。该情况下两作业虽然不存在时序约束,但是两作业调度区间存在重叠区域,此时先按照情况 1 的调度方式得到作业 j 的完成时间 t_{F_j},然后再按照情况 2 的调度方式得到作业 i 的完成时间 t_{F_i}。当 $t_{F_j}\leqslant t_{F_i}$ 时,则先调度作业 i 再调度作业 j;当 $t_{F_j}>t_{F_i}$ 时,则更新 $p_i>p_j$,先调度作业 j,再调度作业 i。

综上,本文提出的改进串行调度算法的具体步骤如下所示:

步骤 1　初始化,更新作业列表 L 和各个作业的 t_{EST_j},t_{EFT_j},t_{LST_j},t_{LFT_j},$j\in L$,转步骤 2;

步骤 2　从作业列表 L 中选择优先级最高的作业 i 和次高优先级的作业 j,将两作业放入待调度作业集合 $\boldsymbol{S}_p$ 中,转步骤 3;

步骤 3　调用启发式柔性资源技能分配算法为集合 $\boldsymbol{S}_p$ 中的作业分配资源,转步骤 4;

步骤 4　判断作业 i 与作业 j 的关系,分别根据情况 1－4 对两作业进行调度,转步骤 5;

步骤 5　更新 $\boldsymbol{S}_p=\varnothing$,$L=L\backslash\{i,j\}$,如果 $L\neq\varnothing$ 则转步骤 1;否则算法结束。

4 实验结果及分析

为验证本文设计算法的有效性，采用 Windows10 64 位操作系统，Intel Core i7-8700、3.20GHz CPU、16.00GB RAM 试验平台，开发环境为 C＃（Visual Studio 2013），选择 PSPLIB 标准算例库中作业规模为 30 作业、60 作业规模的算例进行试验，并对该算例进行适合 MSRCPSP 的改造，设置技能种类为 4 种，资源数量为 10。为验证本文算法有效性，将其与结合随机分配资源的普通遗传算法（SR）、结合本文提出的启发式柔性资源技能分配算法的遗传算法（SP）、结合随机分配资源的多种群遗传算法（MS）进行对比。MP 算法的参数如下：子种群个数为 25，种群规模为 50，交叉率和变异率为所提出的各指数的全部排列组合，即共 5＊5＝25 种，每一个子种群对应一组交叉率和变异率。

遗传算法中迭代次数影响最终解的质量，对迭代次数进行敏感性分析。图 7 显示了不同迭代次数的敏感性分析结果，最终选择迭代次数为 200 代。

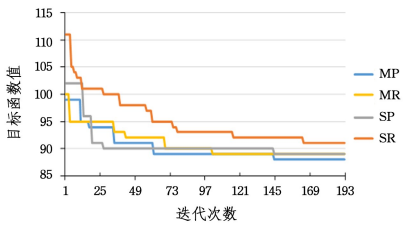


图 7 迭代次数对实验结果的影响

Fig. 7 Influence of iteration on experimental results

对比实验结果数据如表 1－表 3 所列。其中，G_1 为算法 SP 与 SR 求得目标函数值的 GAP，即 $G_1=100*(SR-SP)/SP$；G_2 为算法 MP 与 MR 求得目标函数值的 GAP，即 $G_2=100*(MR-MP)/MP$；GMS 为算法 MP 与 SP 所求目标函数值的 GAP，即 $GMS=100*(SP-MP)/MP$。表 3 单位为 s。

由表 1 可得，在 30 规模算例下，本文设计的启发式算法分配柔性资源要优于随机分配资源方法，在普通标准遗传算法中，启发式分配资源方法优于随机分配方法平均约4.91%，最高提升了约 7.69%；在多种群遗传算法中，启发式分配资源方法优于随机分配方法平均约 3.91%，最高提升了约 7.27%；本文所设计的改进多种群遗传算法比标准遗传算法的求解精度提升了平均约 3.32%最高提升了约 8.33%。综合实验数据结果，由图 8 可得本文所设计的算法在求解 30 规模算例时，所求得解最好。

表 1 30 规模算例的实验结果

Table 1 Experimental results of 30 jobs

No	*SP*	*SR*	*G*1	*MP*	*MR*	G_2	*GMS*
1	52	53	1.92	51	51	0.00	1.96
2	58	61	5.17	55	59	7.27	5.45
3	66	71	7.58	65	69	6.15	1.54
4	68	68	0.00	65	67	3.08	4.62
5	54	56	3.70	52	54	3.85	3.85
6	52	55	5.77	52	54	3.85	0.00
7	58	62	6.90	54	57	5.56	7.41
8	61	65	6.56	61	63	3.28	0.00
9	52	56	7.69	52	53	1.92	0.00
10	52	54	3.85	48	50	4.17	8.33
Avg.	57.3	60.1	4.91	55.5	57.7	3.91	3.32

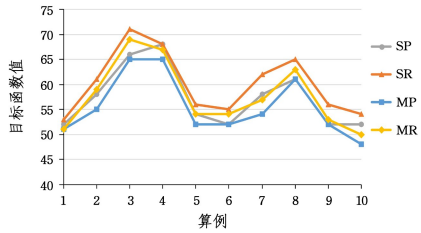


图 8 30 规模算例的实验结果示意图

Fig. 8 Diagram of experimental results for 30 jobs

由表 2 可得，在 60 规模算例下，本文设计的启发式柔性资源技能分配算法在标准遗传算法中求得目标函数值优于随机资源分配算法所求目标函数值约为 2.47%，最高提升了约 5.13%；在多种群遗传算法中求得目标函数值优于随机资源分配算法所求目标函数值约为 3.25%，最高提升了约 5.31%；本文设计的改进多种群遗传算法比标准遗传算法的求解精度提升了平均约 3.56%，最高提升了约 8.65%。综合实验数据结果，由图 9 可得本文所设计的算法在求解 60 规模算例时所求得解最优。

表 2 60 规模算例的实验结果

Table 2 Experimental results of 60 jobs

No	*SP*	*SR*	*G*1	*MP*	*MR*	G_2	*GMS*
1	107	109	1.87	106	107	0.94	0.94
2	114	116	1.75	111	116	4.50	2.70
3	118	121	2.54	112	117	4.46	5.36
4	109	110	0.92	106	109	2.83	2.83
5	117	123	5.13	113	119	5.31	3.54
6	105	110	4.76	101	103	1.98	3.96
7	113	113	0.00	104	108	3.85	8.65
8	105	107	1.90	102	103	0.98	2.94
9	112	114	1.79	108	113	4.63	3.70
10	100	104	4.00	99	102	3.03	1.01
Avg.	110	112.7	2.47	106.2	109.7	3.25	3.56

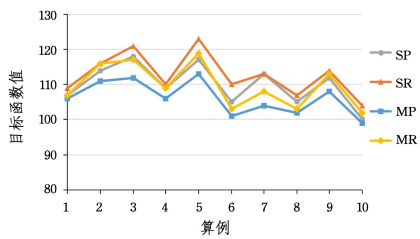


图 9 60 规模算例的实验结果示意图

Fig. 9 Diagram of experimental results for 60 jobs

表 3 对比算法的运算时间结果

Table 3 CPU time of comparison algorithm

No	30				60			
	SP	*SR*	*MP*	*MR*	*SP*	*SR*	*MP*	*MR*
1	2.64	2.72	27.22	25.04	7.19	6.96	68.57	66.07
2	2.37	2.39	24.79	24.54	8.42	6.97	64.72	69.95
3	2.57	2.62	26.67	29.16	7.96	6.69	61.00	62.98
4	2.80	2.74	28.75	28.69	5.93	7.10	55.74	60.24
5	2.72	2.37	28.64	26.55	5.74	6.57	62.20	64.11
6	2.59	2.68	27.72	24.46	5.88	6.46	54.91	57.95
7	2.67	2.84	28.70	28.51	5.87	6.29	57.83	62.05
8	2.87	2.37	30.07	27.93	5.71	6.01	55.86	61.35
9	2.72	2.54	25.71	25.92	6.22	6.71	58.11	60.41
10	2.56	2.63	26.81	27.30	6.04	6.43	56.42	62.09
Avg.	2.651	2.59	27.508	26.81	6.496	6.619	59.536	62.72

由表 3 可得,本文设计的启发式柔性资源技能分配算法并没有增大算法的求解复杂度,在 30 规模和 60 规模的算例中,SP 算法和 SR 算法的求解时间在同一数量级内,同样 MP 算法和 MR 算法的求解时间也在同一数量级内。本文设计的多种群遗传算法虽然在求解时间上大于标准遗传算法,但是仍处于同一数量级内,在可接受范围内。综上所述,本文设计的算法可以有效地求解 MSRCPSP 问题。

结束语　本文以实际生产中存在柔性资源为背景,研究了柔性资源受限的项目调度问题,并针对该问题提出了一种改进的多种群遗传算法,设计了交叉移民算子,避免了算法易于早熟收敛的缺点。同时,构造了一种具有前瞻性的考虑局部两作业情况的改进串行调度算法,并结合启发式柔性资源技能分配算法,有效地解决了柔性资源分配问题和作业调度问题。通过在 PSPLIB 算例库中不同规模下的算例数据实验,相比传统的遗传算法求解精度提升了约 4%,验证了本文设计算法求解 MSRCPSP 问题的有效性。后期可以进一步研究具有不同技能水平特性的柔性资源相关资源受限项目调度问题或资源投入问题等。

参 考 文 献

[1] KADRI R L, BOCTOR F F. An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case[J]. European Journal of Operational Research, 2018, 265(2): 454-462.

[2] REN Y F, LU Z Q, LIU X Y, et al. Project scheduling problem with hierarchical levels of skills[J]. Journal of Zhejiang University (Engineering Science), 2017, 51(5): 1000-1006.

[3] NÉRON E. Lower bounds for the multi-skill project scheduling problem[C]// Proceeding of the Eighth International Workshop on Project Management and Scheduling. 2002: 274-277.

[4] BELLENGUEZ-MORINEAU O. Methods to solve multi-skill project scheduling problem[D]. Springer-Verlag, 2008.

[5] LI H, WOMER K. Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm [J]. Journal of Scheduling, 2009, 12(3): 281.

[6] JIA Y, WANG Z M, ZHANG Y G. Heuristic algorithm for flexible resource-constrained project scheduling problem[J]. Computer Integrated Manufacturing Systems, 2015, 21(7): 1846-1855.

[7] ZHANG M, LU Z Q. Improved algorithm for multi-skill resource constraint project scheduling problem[J]. Computer Integrated Manufacturing Systems, 2016, 22(3): 782-792.

[8] XIE F, XU Z, YU J. Bi-objective optimization for the project scheduling problem with variable resource availability[J]. Systems Engineering-Theory & Practice, 2016, 36(3): 674-683.

[9] HEIMERL C, KOLISCH R. Scheduling and staffing multiple projects with a multi-skilled workforce[J]. OR spectrum, 2010, 32(2): 343-368.

[10] LIU Z Y, YUAN H T, ZHOU C, et al. Branch and bound based approach for preemptive project scheduling with time-window constraints on multi-skilled resources[J]. Systems Engineering-Theory & Practice, 2019, 39(1): 183-199.

[11] MYSZKOWSKI P B, OLECH Ł P, LASZCZYK M, et al. Hybrid differential evolution and greedy algorithm (DEGR) for solving multi-skill resource-constrained project scheduling problem[J]. Applied Soft Computing, 2018, 62: 1-14.

[12] TRITSCHLER M, NABER A, KOLISCH R. A hybrid metaheuristic for resource-constrained project scheduling with flexible resource profiles[J]. European Journal of Operational Research, 2017, 262(1): 262-273.

[13] ZHA H, ZHANG L. Fuzzy flexible resource constrained project scheduling based on genetic algorithm[J]. Transactions of Tianjin University, 2014, 20(6): 469-474.

[14] RANJBAR M, KIANFAR F. Resource constrained project scheduling problem with flexible work profiles: a genetic algorithm approach[J]. Scientia Iranica. Transaction E, Industrial Engineering, 2010, 17(1): 25.

[15] XING L N, CHEN Y W, YANG K W. Multi-population interactive coevolutionary algorithm for flexible job shop scheduling problems[J]. Computational Optimization and Applications, 2011, 48(1): 139-155.

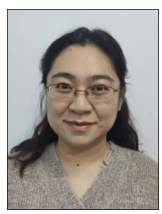
[16] COCHRAN J K, HORNG S M, FOWLER J W. A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines[J]. Computers & Operations Research, 2003, 30(7): 1087-1102.

[17] JIAO Y L, ZHANG P, TIAN G D, et al. Slotting optimization of automated warehouse based on multi-population GA[J]. Journal of Jilin University(Engineering and Technology Edition), 2018, 48(5): 1398-1404.

[18] LIU A J, YANG Y, XING Q S, et al. Multi-population genetic algorithm in multiobjective fuzzy and flexible Job Shop scheduling[J]. Computer Integrated Manufacturing Systems, 2011, 17(9): 1954-1961.

[19] WANG D, GUI W X. A multi-population genetic algorithm based on cross accessibility evaluation[J]. Journal of Guangxi University(Natural Science Edition), 2015, 40(6): 1508-1516.

[20] MAGHSOUDLOU H, AFSHAR-NADJAFI B, NIAKI S T A. A multi-objective invasive weeds optimization algorithm for solving multi-skill multi-mode resource constrained project scheduling problem[J]. Computers & Chemical Engineering, 2016, 88: 157-169.



YAO Min, born in 1982, lecturer. Her main research interests include computer application, data center network, cloud computing and computer software.