1. 绪论
   1. 引言

在企业生产制造过程中，生产调度问题是一直需要解决的核心问题之一。通过求解生产调度问题，可以在一定的时间内对有限的生产资源进行合理的分配，制定合适的生产计划来满足特定的性能指标[1]。其本质是提高生产效率，保证生产的稳定高效，优化资源配置，降低生产成本，从而提高企业的经济效益。因此对生产调度问题的研究有着重大的现实意义。

根据生产过程中的工艺约束和设备特征，一般可以将生产调度问题分为单机调度问题(Single Machine Scheduling Problem, SMSP)、并行机调度问题(Parallel Machine Scheduling Problem, PMSP )、流水车间调度问题(Flow Shop Scheduling Problem, FSSP)和作业车间调度问题(Job Shop Scheduling Problem, JSSP)。并行机调度作为经典的生产调度问题，广泛的存在于各种生产制造领域，是联系不同调度问题的纽带之一。而一致并行机调度(Identical Parallel Machine Scheduling, IPMS)又是并行机调度问题中最普遍的一种类型。在一致并行机调度中，所有的机器加工能力相同，这种生产特性符合现实中的多种场景，如芯片制造[2]，螺母加工[3]，轮毂生产[4]等。对一致并行机调度问题进行研究，可以优化生产过程，提供可靠的生产加工方案，有着十分重要的理论意义和实用价值。

经典的对于一致并行机调度问题的研究通常都在确定性环境下进行，及各种生产要素都是固定已知的。然而实际的生产环境中充满了各种不确定因素[5][6]，如工人熟练度不同，机器故障，现场意外干扰等。此时传统的确定性优化模型与实际的加工模型之间存在一定的差距，求解确定性调度模型得到的方案难以达到生产预期。因此在具体求解时需要充分考虑不确定因素，建立合适的不确定性一致并行机调度模型来减少其造成的影响。

除了不确定因素外，在现实的生产过程中往往需要面对多重的目标和期望。如投资者希望节约成本；生产部门希望缩短工期；客户希望交货期准确等。这些不同的目标和期望之间往往存在冲突，难以同时满足。相比单一优化目标的生产调度问题，研究多目标的一致并行机调度问题，可以帮助决策者更好地权衡不同目标，做出更符合实际生产情况的生产计划，近年来也逐渐成为研究的热点。

本论文将重点研究加工时间不确定的一致并行机调度问题。在本章中将首先介绍确定性一致并行机调度问题的研究现状，然后详细说明不确定因素的处理方法和不确定性一致并行机调度的研究现状。接着阐述多目标一致并行机调度问题的研究现状，对不同的优化目标和相关求解算法进行介绍说明。最后给出本文的内容安排。

* 1. 确定性并行机调度问题研究现状

作为经典的生产调度问题之一，并行机调度问题可以看作是联系单机问题和更复杂的多机调度问题间的重要纽带，在过去的数十年里，国内外研究学者对并行机调度问题进行了深入的研究，产生了大量的学术文献，为并行机调度的发展打下了坚实的基础。

McNaughton[7]在1959年发表的论文是对于并行机最早的研究之一。他研究了带有截止日期的并行机调度问题，并在文中对并行机问题提出了三种性能指标。分别为依赖完工时间的性能(Completion Time Based Performance, CTB)，依赖截止日期的性能(Due-Date Based Performance, DDB)和依赖加工流水时间的性能(Flowtime Based Performance, FTB)。虽然其并未在论文中给出具体的求解算法，但在此之后，并行机调度问题开始进入研究者的视野。

关于并行机调度问题的研究在过去的几十年里不断地得到充实和完善。与现实的各种应用场景的联系也逐渐密切。随着求解时目标函数的多样化，求解时约束和规模的增加，求解的难度也在与日俱增。而与之对应的求解算法也在不断地进步。从最初的一些简单优化规则，到利用运筹学的一些精确算法，再到融合了不同学科领域技术的启发式算法，并行机调度问题的求解规模和求解效率也在逐渐提升。发展至今，在并行机调度的求解算法上有着丰硕的研究成果。这些求解算法主要可以分为两大类，精确算法和近似/启发式算法。

精确算法通常建立在数学规划的基础上，利用穷举法，分支定界法，或者割平面法等数学方法进行求解。分支定界算法(Branch And Bound algorithm, B&B)是一类有效的求解离散规划问题的优化算法，其本质上是一种隐枚举算法。求解时将问题的解空间分解成不同的子集，通过计算不同子集的上下界抛弃不符合的解来缩小搜索范围。分支定界算法在并行机调度中有着广泛的应用。Dell等[8]采用了分支定界算法对目标函数为最大完工时间的一致并行机问题进行了求解，并对问题的上下界进行了优化。实验结果表明该方法可以求解各种规模的算例，求解结果优于一般的动态规划算法。Shim等[9]在求解可分段加工的一致并行机调度问题式，针对最小化总滞后时间的目标函数计算了新的下界，并以此设计了新的分支定界算法。Haouari等[10]在分支定界算法中加入了更加紧缩的增强型下界方式用于一致并行机最大完工时间的求解，并通过大量的算例证明了该方法的求解效率。除了分支定界法外，Mokotoff[11]设计了一种割平面法。他将一致并行机调度问题用数学规划的形式表示，然后对其进行松弛。通过不断地向松弛模型中增加有效割的方式来求得最优解。该割平面法简单有效，表现出了比分支定界方法更优秀的求解能力。除此之外，Dell等[12]提出了一种分支定价算法，Chen等[13]采用了一种列生成算法来进行一致并行机调度问题的求解。

以上提到的都是用于求解一致并行机问题的精确算法。然而一致并行机调度问题是NP-hard问题[14]，当问题规模增大时，精确算法的求解时间和求解难度呈指数型增加，很难在可接受的时间内获得满意的结果。为了提高求解效率，启发式算法在并行机调度领域的应用逐渐成为研究的重点。根据算法机理的不同，启发式算法可以分为构造启发式算法和迭代启发式算法。构造启发式算法通常基于一些启发式规则进行计算。这类规则问题针对性很强，在求解特定问题时可以简单快速地获得近似最优解。Graham[15]提出了最长加工时间优先规则(longest processing time first, LPT)，即各工件按照加工时长依次放到空闲机器上进行加工，并证明在台机器的并行机调度问题下，通过该规则得到的makespan与最优性能的性能比不大于。LPT规则在并行机调度问题中通常用于一些上界的计算，或是用于产生初始解[11][16]。除了LPT规则外，MULTIFIT算法[17]，COMBINE算法[18]，LISTFIT算法[19]等都是应用于并行机调度问题的构造启发式算法。

相比构造启发式算法，迭代启发式算法凭借着通用性强，对解空间的搜索深入等特点越来越受到研究人员的重视。迭代启发式算法往往从一个初始解或由多个初始解组成的初始种群出发，根据一定的机制或者规则对解空间进行搜索。在每一次的迭代中对当前解或当前种群进行更新，在达到终止标准后停止迭代。根据每次进行迭代的解数量，可以将迭代启发式分为邻域搜索算法和群智能搜索算法。邻域搜索算法是一类串行搜索算法，基于单一初始解出发，每次迭代时对当前解产生一批候选解的集合（邻域），从中选取解进行更新。常见的邻域搜索算法有模拟退火算法(Simulated Annealing, SA)，禁忌搜索算法(Tabu Search, TS)，变邻域搜索(Variable Neighborhood Search, VNS)等。SA是一种模拟物理固体退火过程的算法，利用Metropolis准则来防止算法陷入局部最优。Lee等[16]利用SA对一致并行机调度问题进行求解，采用LPT准则来产生初始解，通过实验仿真证明了SA相比其他构造性启发式算法由更强的求解能力。TS利用禁忌表来“记忆”一定次数的操作，从而防止算法陷入局部最优。Yamashita等[20]采用TS来优化一致并行机调度问题中的工件平均滞后时间，并通过增加禁忌表长度和设计更有效的解移动方式来提升对解空间的探索能力。邻域构造是邻域搜索算法中的重要步骤，VNS对同一个当前解采用多种邻域构造产生不同的邻域解，再从中选出解进行更新。Alharkan等[21]对最小化makespan的一致并行机调度问题采用了五种邻域构造方式来产生邻域解，并分别用随机和LPT规则两种方式产生初始解。仿真结果表明该改进的变邻域搜索算法比起一般的邻域搜索算法可以获得更好的平均makespan。

群智能搜索算法以多个解组成的种群为基础，通过模仿自然界中的一些生物行为或者自然规律对解空间进行全面的探索。如模仿生物进化的遗传算法(genetic algorithm, GA)，参考鸟群捕食的粒子群算法(Particle Swarm Optimization, PSO), 模仿布谷鸟下蛋行为的布谷鸟算法(Cuckoo Search Algorithm, CSA)，模拟蚂蚁觅食行为的蚁群算法(Ant Colony Optimization, ACO)等。群智能搜索算法在并行机调度问题中有着各种应用。Liu等[22]采用了GA对一致并行机调度问题进行求解，利用交叉变异等方式对种群进行更新，通过实验证明求解大规模问题时GA比SA和LPT表现更佳。PSO算法一开始是用于连续优化问题的求解，Kashan等[23]在此基础上加入了离散化的步骤来处理离散的调度解，提出了一种离散化后的PSO算法(Discrete Particle Swarm Optimization, DPSO)用于求解目标函数为makespan的一致并行机调度问题。Neto等[24]采用蚁群算法对部分生产可进行外包的一致并行机调度问题进行了求解，优化目标是外包费用和延期损失的花费总和。并提出了三条种群移动规则来保证每个解朝着优化目标函数的方向进行更新。并在每次迭代中加入了局部搜索的步骤来提升求解质量。Laha等[25]在求解一致并行机调度问题时提出了一种改进布谷鸟算法(Improved Cuckoo Search Algorithm, ICSA)，对传统的CSA加入了离散化的策略使得CSA中用于种群更新的莱维飞行能适用于调度问题。同时还加入了局部更新的步骤来增强算法的搜索能力。实验结果表明布谷鸟算法有着强大的全局搜索能力，相比PSO, SA，GA等能获得更好的求解结果。

* 1. 不确定性并行机调度问题研究现状

上一小节介绍了近些年来并行机调度领域的丰富研究成果。然而大部分的研究都是在理想的确定性环境下进行的，即各种生产参数如加工时间等都是已知且确定的。然而由于存在工人加工熟练度，机器故障，现场意外的加工干扰等不确定因素[26]-[28]，传统的确定性模型和实际加工模型之间存在着一定差距，此时按照求解确定性模型得到的方案进行加工制造难以达到生产预期。因此对不确定并行机调度问题进行研究有着重要的学术意义和实际价值。如何处理并行机调度中的不确定参数，建立合适的不确定模型并设计不确定环境下的高效求解算法逐渐成为了研究的热门。

对于如何处理不确定参数，目前主要有随机分析法，模糊分析法和场景法三种方式。在随机分析法中，不确定参数用随机变量表示，相应的随机变量利用概率分布模型进行描述。通常对实际生产中的大量历史数据进行反复分析，利用得到的统计规律来确定其概率分布模型。常见的模型有指数分布，正态分布等。如Alimoradi等[29]在优化一致并行机调度的总流水时间时，将不确定的工件加工时间用满足正态分布的随机变量表示，并设计了一种分支定界算法进行求解。Ranjbar等[30]提出了两种分支定界算法来求解最大化顾客满意度的一致并行机调度问题，其中每个工件的加工时间满足正态分布。Cai等[31]研究了每台机器都会有随机故障发生的并行机调度问题，优化目标是最小化提前完工造成的花费和总滞后工件数。其中的交货期满足指数分布。

在有些情况下通过对历史数据的分析，只能获得不确定参数的一些近似值，很难得到不确定参数服从的具体概率分布。此时可以使用模糊分析法来进行不确定参数的处理。同随机分析法不同，在模糊分析法中，不确定参数不再满足特定的概率分布模型，而是引入了模糊数的概念，利用模糊数和对应的模糊规则来对不确定参数进行描述和预测[32]。Balin[33]用三角模糊数来表示不确定并行机调度中的加工时间，采用GA来优化最大完工时间的目标。并通过测试不同模糊集下的问题，将GA求得的结果与LPT方法获得的结果进行了比较来体现GA算法的有效性和稳定性。Yeh等[34]在优化不确定并行机调度的最大完工时间时，用梯形模糊数来表示不确定的加工时间，并同时采用了SA和GA两种智能优化算法来进行求解对比。Behnamian[35]用钟形模糊数来描述不确定的加工时间，并采用了面积中心法来进行解模糊。同时设计了一种离散粒子群算法，在粒子群算法中加入了遗传算法的交叉变异操作，并用该改进的算法求解较大规模的模糊一致并行机调度问题。

随机分析法和模糊分析法都需要大量的数据或经验作为基础。然而在现实的生产环境中，经常会出现诸如新工艺，新生产线等历史数据不足的情况[36][37]。不同于随机分析法和模糊分析法，场景法不需要历史数据的积累，适用于在不确定因素的发生概率未知或历史信息不足的情况下描述不确定参数。场景法可以分为区间场景和离散场景两种形式[38]。区间场景下利用不确定参数可在一个上下界内任意取值。而离散场景下一个场景代表一组可能的参数离散值，通过所有可能出现的离散场景集合来描述不确定性。Li等[39]将基于场景的不确定模型分为两类，基于场景的随机优化模型和基于场景的鲁棒优化模型，两种模型体现了不同的决策偏向。基于场景的随机优化模型将整体期望性能作为优化目标[40]，考虑优化整个调度解在所有可能场景下性能的平均水平。如Liu等[41]研究了离散场景下的一致并行机调度问题，不确定加工时间用多个离散场景的集合表示，优化目标是总完工时间在所有场景下的期望值。并且分别采用抽样平均近似法(Sample Average Approximation, SAA)和GA来对不同规模的算例进行求解。而基于场景的鲁棒优化模型则是以增加系统抵抗不确定性干扰的能力为优化目标，比较常见的有min-max模型(最坏场景模型)和min-max regret模型(最大后悔场景模型)。这类模型体现了决策者的风险厌恶偏向，通过优化所有场景下的最坏性能或者最大遗憾值来提升系统的抗风险能力。如Xu等[42]对区间场景下的一致并行机最大makespan的遗憾值进行优化，分别采用了松弛迭代算法和SA来求解不同规模的问题。Drawl等[43]对区间加工时间下的一致并行机调度问题进行了研究，优化目标是最大完工时间的最大遗憾值。Wang等[44]研究了工件可外包的一致并行机调度问题，目标函数是外包消耗花费和非外包加工花费的生产总成本。他们同时采用了离散场景和区间场景来描述不确定的加工时间，利用2-近似算法(2-Approximation Algorithm)分别对区间场景下生产总成本的最大遗憾值和离散场景下的最大生产总成本问题进行了求解。这类传统的鲁棒优化模型因为过度关注单一的最差解，因此最终获得求解结果往往过于保守。针对该不足目前也出现了一些新的鲁棒模型，如Wang等[45]以一个性能值为阈值，以此筛选出性能表现劣于阈值的坏场景，通过综合考虑其他坏场景的性能而不是单一的最坏场景性能来减少最终调度解的保守性，然而这类模型在鲁棒并行机调度中还未有见到具体的使用。

以上提到的随机分析法，模糊分析法和场景法都是并行机调度中处理不确定参数的有效方法。然而随机分析法和模糊分析法都需要大量的历史数据，其概率分布函数或者模糊隶属度函数一般不太容易获得。相比之下场景法的描述方式更加准确且容易实现。而在不确定环境下，获得抵御风险能力的结果体现了很多决策者的保守追求，因此基于场景的鲁棒优化模型逐渐成为了研究的热门。综上考虑，本文将采用离散场景来进行描述一致并行机调度问题中的不确定加工时间，并以此建立合适的鲁棒模型进行求解。

* 1. 混合算法在并行机调度问题的应用

上文介绍了并行机调度问题在确定性和不确定环境下的不同求解方法。包括精确求解方法和启发式算法。其中的迭代式启发式算法，或称为智能优化算法，相比精确算法有着更强的求解效率和通用性，在较大规模或复杂目标函数的并行机调度问题的求解中应用广泛。这些智能优化算法有着各自的特点，对解空间有着不同的搜索方式。目前的研究显示，相比单一的智能优化算法，混合智能优化算法通过结合不同算法的优点，在对一致并行机调度问题的求解中表现得更加出色[46]-[48]。

群智能搜索算法有着强大的全局搜索能力，但其局部搜索能力较差，因此常常采用往群智能搜索算法中加入邻域搜索算法或者局部搜索策略的方式来构造混合算法。通过结合两种搜索方式的优点，混合算法可以对解空间有着更强的搜索能力。

如Wang等[46]在优化有工艺约束的并行机调度问题时，采用了一种混合了禁忌搜索的遗传算法(Tabu-Genetic Algorithm)，用禁忌搜索来增加遗传算法的局部搜索能力。Lee等[47]在求解一致并行机调度问题时，在遗传算法的基础上加入了局部搜索（GA+LS），通过混合算法的形式来提升求解效率，其中每个个体在每次迭代过程中都会有一定的概率被其邻域解代替。和声搜索算法(Harmony Search, HS)是一种群智能优化算法，是对多个乐师创作歌曲的一种模，拟有着良好的全局搜索能力，但其在局部搜索能力上依然有改善的空间。Chen等[48]在和声算法的基础上提出了动态和声算法(Dynamic Harmony Search, DHS)来求解最小化makespan的一致并行机调度问题，把整个和声库分为多个子和声库分别进行独立的进化更新。同时在每次迭代过程中加入了VNS算法，根据并行机调度问题的特点设计了四种邻域构造方式，利用VNS来对每次产生的邻域解进行局部搜索，以此来增强DHS的局部搜索能力。Kashan等[23]在其提出的离散粒子群算法的基础上加入了局部搜索的环节，形成了混合离散粒子群算法(Hybridized Discrete Particle Swarm Optimization, HDPSO )。实验结果表明加入局部搜索后的算法明显提高了求解能力，在对大规模一致并行机问题的求解中无论是求解花费的时间还是求解的结果都优于之前的DPSO。上文提到的布谷鸟算法利用莱维飞行进行每代种群的更新，有着强大的全局搜索能力，但也存在局部搜索能力不强的缺点。Guo等[59]采用了三种面向一致并行机调度问题的邻域结构，并将采用了这些邻域的局部搜索与布谷鸟算法混合，用于对每代种群中性能表现较差的个体进行更新，将布谷鸟算法引向更好的搜索空间，实现了不同搜索方式的优势互补。而加入了局部搜索的布谷鸟算法也比单一的布谷鸟算法表现出了更好的实验结果。

根据以上的混合算法在并行机调度问题中的研究现状，综合布谷鸟算法在各群智能算法中的优异表现，本文也将采用混合算法，采用布谷鸟算法加入局部搜索的方式来对本文研究的一致并行机调度问题进行求解。同时本文还将根据研究的问题模型特点设计面向问题的邻域构造方式，让局部搜索更具效率。混合算法的细节将在第三章中进行详细的描述。

* 1. 本论文的研究内容及章节安排

本论文针对加工时间不确定的一致并行机鲁棒调度问题开展研究，目标函数是最大完工时间makespan。首先采用离散场景来描述不确定加工时间，然后求解一致并行机的坏场景集鲁棒调度模型。这是对传统鲁棒优化模型上进行拓展的一种优化模型，通过关注更多坏场景下的性能来降低最终鲁棒解的保守性。本文将先求解一种单阶段阈值坏场景集惩罚模型，结合一致并行机调度问题和该模型的特点，设计了一种面向问题的邻域构造方式，并将采用该邻域方式的局部搜索与布谷鸟算法结合用于该鲁棒模型的求解。然后求解了一个两阶段阈值坏场景集模型。根据求解精度的需求，分别采用割平面法和混合布谷鸟算法对两个阶段进行求解。并通过仿真实验来证明所提算法和模型的有效性。

基于本文所要研究的内容，各章节安排如下：

第一章绪论，首先介绍了确定性并行机调度问题的研究现状，然后介绍了不确定并行机调度问题的研究现状，简单介绍了处理不确定参数的不同方法。接着介绍了求解并行机调度问题中应用广泛的混合算法，引出本文的研究内容。

第二章中对一致并行机调度问题进行了详细介绍，给出了确定性一致并行机调度问题和场景集下不确定一致并行机优化常见优化模型的数学规划形式。然后介绍了求解一致并行机问题的常规算法，包括割平面法和布谷鸟算法。并对一致并行机调度中常用的邻域构造方式进行了详细的阐述。

第三章是对加工时间不确定的一致并行机调度问题进行研究，采用离散场景表示不确定加工时间，构建了离散场景下的一致并行机阈值坏场景集惩罚模型。针对该模型的特点设计了一种面向模型的合并场景邻域方式，将采用了该邻域的局部搜索加入到布谷鸟算法中形成一种混合布谷鸟算法，并对该邻域构造方式和混合算法的步骤进行了详细的介绍。然后将混合算法与其他三种已有的算法一起用于该模型的求解，通过仿真测试结果证明该混合布谷鸟算法的有效性。

第四章在第三章的基础上，构建了离散场景下一致并行机调度问题的两阶段阈值坏场景集模型。在第一阶段采用精确算法中的割平面法来求解阈值，在第二阶段采用第三章设计的混合布谷鸟算法进行求解。通过与其他算法的比较证明了该求解算法的有效性。

第五章对本文进行了总结和展望。

1. 一致并行机调度问题及其求解算法
   1. 引言

并行机调度问题属于典型的生产调度问题，是联系单机调度问题和其余复杂调度问题的纽带[49]。一致并行机调度是并行机调度中最普遍的加工类型，涉及多台加工能力相同的机器，在实际生产中有着广泛的应用。其在确定性环境下的调度模型和不确定环境下的鲁棒模型都是研究的热门。用于一致并行机问题的求解算法多种多样，可以满足不同精度和规模的需求。在精确算法中，割平面法凭借高效易于操作的特点被广泛地使用。而在启发式算法中，布谷鸟算法作为一种较为新颖的群搜索算法，有着强大的全局搜索能力，经常通过混合局部搜索，增加对邻域解的搜索来提升求解效率，被应用于各种调度问题包括一致并行机调度中。

基于以上内容，本章的内容安排如下：2.2节介绍确定性一致并行机问题及其数学模型。2.3节对不确定环境下的一致并行机鲁棒调度问题进行介绍，包括一致并行机常见的一些鲁棒指标及其鲁棒模型。2.4节是对求解一致并行机的割平面法的介绍，2.5节是对标准布谷鸟算法的介绍，包括了算法原理和基本步骤。2.6节是对一致并行机调度中的局部搜索以及用于一致并行机邻域构造方式的介绍。2.7节是对本章内容的总结。

* 1. 确定性一致并行机调度问题

并行机调度问题涉及个工件， 台机器 ，表示第个工件，为所有工件的集合；表示第台机器，为所有机器的集合。表示在上的加工时间。同时在并行机调度问题中，还有以下的约束：

* 任意一个工件可在任意一台机器上进行加工，工件之间相互独立，没有优先级。
* 一台机器同一时刻只能加工一个工件，一旦开始加工无法中途停止。
* 一个工件只需加工一次。
* 无特殊情况，所有的工件和机器在0时刻都是准备就绪的。

根据加工机器的属性，并行机调度问题可以分为以下三类：一致并行机调度问题，异速并行机调度问题和无关并行机调度问题[50][51]。

一致并行机调度问题中的所有机器没有差别，同一个工件在不同机器上的加工时间完全相同，加工时间只与工件有关。即对于，其在不同机器上的加工时间可统一记作，有。

异速并行机调度中的每台机器各自有一个速度系数。对于机器，其速度系数记作。每个工件有一个基本加工时间，工件在不同机器上的加工时间满足。

无关并行机中的工件加工时间与机器和工件均相关。同一个工件在不同机器上加工，同一台机器加工不同的工件的时间可能都不相同。即不同的之间并无特殊的规律。

本文中研究的一致并行机调度问题，目标函数均为最大完工时间makespan。用表示所有调度解的集合，为一个具体的可行调度解，如果选择在上加工，则，否则。表示机器在调度方案下的完工时间，为在上加工的所有工件的加工时间之和：



一致并行机调度中所有机器中最长的完工时间即为最大完工时间，记作：



当所有工件的加工时间都为确定的数值时，这样的问题被成为确定性一致并行机调度问题。

以makespan为目标函数的一致并行机问题(Identical Parallel Machine Scheduling，IPMS)，用三元法表示为，该问题可表示为如下的数学规划模型：

(IPMS):









* 1. 基于场景的不确定一致并行机随机优化模型

在实际情况中，各工件的加工时间往往是无法确定的。本文采用1.3节介绍过的离散场景来描述不确定的加工时间，研究场景集下的一致并行机调度问题(Scenario Identical Parallel Machine Scheduling, SIPMS)。本文中用表示所有离散场景的集合，为场景集中的场景总数。表示一个具体的场景，为一组可能的加工时间取值。表示调度解中的机器在场景下的完工时间，表示在场景下调度解的makespan：



可以看出，在给定场景后，以式为目标函数的SIPMS问题即为上节提到的表达为的确定性一致并行机调度问题。

场景下的不确定一致并行机调度问题存在不同的优化模型，可以分为基于场景的随机优化模型和基于场景的鲁棒优化模型[39]。体现了不同的决策偏向。

随机优化模型通常以所有场景下的期望性能作为优化指标，因此也称为期望模型[52]。每个场景赋予发生概率，用表示调度解在中的所有可能场景下的期望性能指标(Expected-performance Criterion)，以作为优化指标的期望模型(Expected-performance Criterion Model, ECM)的表达式如下所示：

(ECM)



而在本文研究的离散场景中，所有的场景均以“纯”场景的方式出现，即不同场景之间没有发生概率的差别，或者说不同场景出现的概率相同。此时优化的期望性能表现为所有场景下的均值性能。用表示调度解在中的所有可能场景下的均值性能指标(the Mean-performance Criterion)，以作为优化指标的均值模型(the Mean-performance Criterion Model, MCM)的表达式如下所示：

(MCM)



SIPMS下的MCM模型可以转换为单一场景下的确定性IPMS问题进行求解。

**定义2.1** 以每个工件的加工时间为该工件在中所有场景下加工时间的均值所构造的场景称为场景集下的均值场景，用表示，简记为。则， 。

**定理1** 对于场景集下的不确定一致并行机调度问题，其MCM模型可以转换为均值场景下的IPMS模型的形式。

**证明：**用表示第个场景下的makespan,则场景集下的一致并行机MCM模型的数学规划形式表达如下：









…





将~相加，表示为：



对作进一步简化：



根据均值场景的定义，均值场景下的各工件加工时间，则离散场景集下的MCM模型，可以转化为求均值场景下的IPMS的表达形式：









□

根据以上的结论，MCM问题可以看作单一场景下的，通过已有的求解IPMS的方法求解~，将求得的最优解代入中，即可得到所有场景下的最佳均值性能。

* 1. 基于场景的不确定一致并行机鲁棒优化模型

上节提到的随机优化模型将所有场景下的期望性能作为优化目标，体现了偏积极性的决策偏向。而基于场景的鲁棒优化模型则是以增加整个系统抵抗不确定性干扰的能力为优化目标，把抗风险作为基本的决策偏向。

目前对于一致并行机鲁棒优化模型的研究以min-max模型(最坏场景模型)和min-max regret模型(最大后悔场景模型)为主[53][54]。

* + 1. 最坏场景模型

对于一个可行调度解，其在场景集中表现性能最差的场景被称为的最坏场景(the Worst-case Scenario)，记作：



表示表示调度解在中的所有可能场景下最坏的性能 (the Worst-scenario performance Criterion):



以作为鲁棒优化指标的模型，即为最坏场景模型(the Worst-scenario performance Criterion Model，WCM)：

(WCM)



WCM是最传统的鲁棒调度模型，通过优化最坏场景下的性能来提升调度解性能下降的风险，体现了决策者的悲观偏向。因为只关注单一最坏场景，因此最终得到的鲁棒解过于保守。

* + 1. 最大后悔场景模型

Kouvelis等[38]在最坏场景模型的基础上提出了最大后悔场景模型，是一种基于偏差鲁棒指标的鲁棒优化模型。

用表示场景下的最优调度解。为场景下的最优性能。则对于一个可行调度解，其在下的表现性能与该场景下的最优性能的偏差即为在下的后悔值(regret)，记作：



而在所有场景下最大的后悔值被称为的最大后悔值(the Worst-case Regret)：



以作为鲁棒优化指标的模型，即为最大后悔模型(the Worst-case Regret Model，WRM)：

(WRM)



相比WCM，WRM用与场景下最优性能的相对偏差代替了性能的绝对差值，包含了各场景下最优性能的追求，同时也降低了最终解的鲁棒性。但依然只关注后悔值最大的单一场景。同时WRM的求解需要计算每个场景下的最优性能，对于在确定性环境下就为NP-hard的一致并行机调度问题来说无疑增加了求解难度。

* 1. 求解一致并行机调度问题的割平面法

Mokotoff[11]在2004年提出了一种求解的割平面法。同其他求解确定性一致并行机调度的精确算法如分支定界法相比，该方法有着更高的求解效率，并可与其他算法结合用于求解更复杂的目标函数。如Xu等[42]在求解区间场景下的一致并行机最大后悔模型时就采用了该割平面能法来确定每个极点场景下的最优makespan。

* + 1. 算法简介

该割平面法是先确定目标函数的上下界限，令目标函数等于当以前下界并逐步增加下界的值。在给定的情况下对原始的IPMS模型进行松弛求解，看所得的结果是否为整数解。如果不是整数解，则根据求解结果往松弛模型中增加有效不等式(Valid Inequality)。通过有效不等式对松弛可行域进行切割，通过缩减可行域的范围形成让求解结果向最终的整数解不断接近。

根据2.2节中的IPMS的数学规划模型，IPMS问题的解可行域为：









通过将式这一约束条件松弛为，得到松弛可行域：









则IPMS模型的表示形式从最初的变为了，其中的即为有效不等式。每次求解过后通过增加下界的值和加入有效不等式的方式对松弛可行域进行切割，直到得到最终的整数解。

极少情况下会遇到无法增加有效不等式但还未求得最终解，此时根据当前的下界利用分支定界法[8]继续进行求解。

* + 1. 有效不等式的计算

Mokotoff[11]给出了有效不等式的具体证明过程，本文不对证明做详细介绍，只介绍有效不等式的产生方法。

设求解某次的IPMS松弛问题得到的非整数解为，表示在机器上的加工工件集合。将这些工件的加工时间相加，用表示在上的额外时间消耗。用表示中加工时间大于的工件集合。

对于每台机器，当时，可以增加如下的有效不等式作为约束：



下面以一个6工件2机器的一致并行机调度问题为例来展示有效不等式的产生过程，各工件的加工时间为。在时利用CPLEX在松弛可行域下对IPMS的线性规划模型进行求解，求得的结果如下所示：

根据求解结果，确定各机器上的加工工件，，，根据额外时间消耗可以得到。由此可以确定两个有效不等式：





* + 1. 算法步骤

利用割平面法求解IPMS分为以下几个主要的步骤：

Step1：将所有的工件按照加工时间的降序进行编号，使得，这是为了用于接下来的目标函数的初始上下界计算。

Step2：计算目标函数的初始上下界。其中下界(Lower Bound)根据以下的公式计算：



上界(Upper Bound)为利用启发式规则求得的结果，根据Mokotoff[11]给出的示例，采用LPT规则求得。

Step3：如果初始的，则直接跳转step6输出最终的结果，其中为利用LPT规则求得的调度解，为对应的最大完工时间makespan，即为此时的上下界。如果不相等，则将IPMS问题变为其对应的松弛问题，并加入约束。

Step4：根据已有的约束求解IPMS的松弛问题，如果松弛解为整数解，则完成计算，跳转step6输出最终的结果。否则根据式添加有效不等式，同时。重复Step4，直到无法继续产生有效不等式，此时根据当前的，采用分支定界法继续求解。

Step5：输出最终的结果。

整个割平面法的步骤的流程图如图3.1所示：



* 1. 布谷鸟算法

布谷鸟算法(Cuckoo Search Algorithm, CSA)是一种近些年来较为新颖的群智能优化算法，由Yang和Deb[55]于2009年提出。算法设计的灵感来源是自然界中布谷鸟的寄生性产卵行为，并利用莱维飞行(Lévy Flights)进行种群更新。布谷鸟算法的操作简单，全局搜索能力强，在各种优化领域包括生产调度领域都有出色的表现[56]-[58]。在一致并行机调度问题的求解上也有成功应用的案例[25][59]。下面将对布谷鸟算法进行简单的介绍。

* + 1. 算法简介

布谷鸟，又称杜鹃，因叫声类似“布谷”而得名，是自然界中一种常见的鸟类。同其他鸟类不同，自然界中的布谷鸟不会筑巢和育雏。它们会将蛋下在精心挑选的宿主鸟窝中，让宿主来替它们完成孵蛋和哺育的工作。布谷鸟会尽可能地对自己的蛋进行伪装，使蛋的颜色，大小，花纹等都与宿主的蛋相似，而布谷鸟的雏鸟也会模仿宿主雏鸟的叫声来获取更多的食物。但当宿主一旦发现了布谷鸟的寄生情况，布谷鸟的蛋或者幼崽会被宿主抛弃[62]。

生物学家们发现，布谷鸟的飞行轨迹表现出了莱维飞行的特性.除布谷鸟外，自然界中的许多其他鸟类或者昆虫的运动行为也满足这一特征[63]-[67]。这是一种方向服从均匀分布，步长满足莱维分布(Lévy Distribution)的随机游走，而莱维分布同时也是一种重尾幂律分布(Power-law Step-length Distribution with a heavy tail)。相比普通的随机游走，莱维飞行可以更有效地对周边环境进行搜索。

根据布谷鸟搜索宿主鸟窝进行寄生下蛋的行为和莱维飞行，Yang和Deb[55][68]提出了布谷鸟算法。为了简化算法和计算，他们制定了以下三条规则：

1）每只布谷鸟每次只选择一个随机的窝下一个蛋；

2）每代中产生最优秀个体的鸟巢会被保留到下一代；

3）每代种群中会有占总数伪装较差的蛋被宿主以一定概率发现抛弃，为了保持蛋的数量不变，布谷鸟会选择一个新窝重新下蛋。

根据以上规则，布谷鸟算法中每个个体对应一个鸟窝或者蛋，即为一个解，每代种群数量即为每代解的数量。鸟窝或蛋的优劣程度对应解的性能好坏。布谷鸟寻找鸟巢下蛋的过程就是在解空间里搜索新解的过程。蛋被抛弃寻找新窝的过程则为每代种群中性能较差的解以一定概率被新解代替的步骤。布谷鸟行为和利用布谷鸟算法求解优化问题可以有表2.2所示的映射关系：

布谷鸟算法中莱维飞行的步长满足莱维分布，如下式所示，其中为给定的常数：



实际计算过程中采用Mantegna算法来产生服从莱维分布的随机步长[69]，该算法的有效性和合理性已经得到了证明[71]。产生步长的公式如下所示：



上式中的为两个满足正态分布的随机数，其中



随机数满足的正态分布的标准差的计算公式为：



为标准的Gamma函数，对于给定的表现为常数。为圆周率。

* + 1. 算法步骤

上一节介绍了布谷鸟算法的背景原理，本节将对标准布谷鸟算法的具体步骤进行介绍。

在布谷鸟算法中，用表示当前的迭代代数，表示当前代种群，每代种群包含了个解，为对应的目标函数。整个布谷鸟算法的流程图如图2.2所示，主要包括了以下几个步骤：

Step1：随机产生个个体组成初始代种群。并记录下其中性能最好的解。 Step2：对每代种群中的每个解利用式得到下一代的解，组成下一代种群：



其中为步长因子，通常取[59][68] , 为满足莱维分布的随机步长，式~为其具体的计算公式。

Step3：每代种群更新完毕后，计算更新后每一个解的性能，排在后的坏解会被随机产生的新解代替。同时根据最新种群中各解的性能更新最优解。

更新种群和替换坏解的步骤Step2和Step3会重复执行，直到当前代数达到最大迭代次数即时停止。

总体来说，作为一种近年来兴起的群智能优化算法，布谷鸟算法具有以下几个特点：

1）自组织性[72]。设置好算法参数和完成种群的初始化后，布谷鸟算法便不再需要外部的指令控制。整个种群中不存在领导个体，个体之间不存在互相控制，算法通过保留最优解和淘汰坏解的策略使得种群逐渐趋向最优解。

2）并行性。布谷鸟算法是一种并行搜索算法，整个种群有着共同的优化目标，不会因为个体的缺陷而受到影响.

3）全局搜索能力强。依靠莱维飞行的种群更新机制，保留最优解的机制，淘汰劣势个体机制的有机结合，布谷鸟算法具有强大的全局搜索能力。莱维飞行的飞行步长呈现频繁的短距离和偶然的长距离，相比遗传算法，粒子群算法等其他群智能优化算法，这一方式有助于算法在对解空间进行全局高效搜索的同时跳出局部最优[73]。保留最优解的方式可以对优势个体保持跟踪。而淘汰劣势个体能够在进行全局搜索时维持种群的质量。

4）易于实现，混合方便。布谷鸟算法的算法参数较少，步骤简单，公式明确。作为一种开放式算法，布谷鸟算法容易于其他搜索方式进行融合，优势互补，形成性能更好的混合算法[72]。

* 1. 一致并行机调度问题的邻域构造方式

在本文的第一章介绍过求解并行机调度问题的混合算法，通常采用将群智能优化算合和局部搜索相结合的方式来提升求解效率。通过结合群智能算法在搜索大规模解空间时的优势和局部搜索在局部深度搜索上的的优点，混合算法相比单一的迭代启发式算法对解空间由更强的搜索能力。如上文介绍的布谷鸟算法，其依靠莱维飞行有着强大的全局搜索能力，但是局部搜索能力相对欠缺，加入局部搜索后可以对种群中表现较差的个体进行更新，实现不同搜索方式的优势互补，将布谷鸟算法引向更广的搜索空间[58]-[61]。

邻域构造是局部搜索中的一个关键步骤。根据所求问题或模型的特点设计面向问题(模型)的邻域构造方式，可以产生性能更佳的邻域解，从而进行更高效的搜索。

在一致并行机调度问题中，每个解中加工时间最长的机器被称为关键机器，其余的机器被称为非关键机器[42]，最常用的方式是通过对关键机器上的工件进行邻域操作来产生相应的邻域解。这些邻域操作可以分为交换(Swap)和插入(Insert)两种方式[24]，而交换方式又可以根据交换工件的数量分为一对一交换(Swap)，非对称交换(Asymmetric Swap)和两两交换(Double Swap)。插入操作是将关键机器上的一个工件移动到另一台非关键机器上加工，一对一交换是选择关键机器上的一个工件，与另一台非关键机器上的工件交换加工位置；非对称交换将关键机器上的一个工件与某台非关键机器的两个工件进行加工位置的交换；两两交换则是交换关键机器上的两个工件与某台非关键机器上的两个工件。非关键机器和进行插入或者交换的工件都是随机选择的。

这里以一个7工件3机器的确定性并行机调度问题为例展示这几种邻域操作方式，各工件的加工时间如表2.3所示，其中工件在上加工，在上加工，在上加工。

图2.3通过甘特图展示了四种邻域解产生的过程。根据表2.3所示的各工件加工时间，当前解中为关键机器，和为非关键机器。a)图所示的插入操作将关键机器上的工件移动到一台非关键机器上进行加工；b)图为一对一交换，选择上的工件与上的交换加工位置；c)图中的操作方式为非对称交换，从关键机器上选择了一个工件放到上加工，再将原本在上加工的两个工件和移到了。d)图展示了两两交换的操作过程，将中的与非关键机器上的两个工件进行位置的交换。



* 1. 本章小结

本章首先介绍了一致并行机调度问题，给出了具体的定义和相关的表示符号。重点介绍了最具代表性的以最大完工时间为目标函数的确定性一致并行机调度问题，给出了其数学规划模型。然后对基于场景的不确定一致并行机调度问题进行了叙述，介绍了几种常见的基于场景的随机优化模型和鲁棒优化模型，并给出了均值场景的定义，以此实现了一致并行机场景均值模型和确定性一致并行机调度问题间的转换。

除了求解模型外，本章还对求解一致并行机调度问题的精确算法和启发式算法分别进行了介绍，重点介绍了精确算法中的割平面法和启发式算法中的布谷鸟算法。对这两种算法的的算法原理和基本步骤都进行了详细的介绍。然后简单提及了启发式算法中采用群智能搜索算法和局部搜索结合的混合算法，并对局部搜索中的关键步骤邻域构造方式进行了具体的描述，通过甘特图的方式展示了四种并行机调度问题中最常用的邻域构造方式。

目前对于不确定一致并行机调度问题的研究，在求解模型上依然以本章介绍的传统鲁棒模型为主。但是这类模型保守性较强，存在一定的缺陷。在求解算法上，割平面法和布谷鸟算法在一致并行机问题的求解中都有成功的应用。割平面法作为一种精确算法，适合在求解规模较小精度要求高的场合使用。而作为一种群智能搜索算法，布谷鸟算法有着强大的全局搜索能力，对于复杂并行机问题的求解有着显著的优势，并可以通过与局部搜索混合的方式来提升其求解效率。根据以上对不确定一致并行机调度问题的介绍，本文将在下面的章节中在已有鲁棒模型的基础上提出新的鲁棒模型，更好地处理鲁棒优化过程中保守性和积极性间的对立统一。同时本文将采用割平面法和混合了局部搜索的布谷鸟算法对新的一致并行机鲁棒调度模型进行求解，其中局部搜索的邻域构造是在常规的一致并行机邻域基础上面向该鲁棒模型特点设计的全新邻域构造方式。

1. 一致并行机阈值坏场景集惩罚模型及其求解算法
   1. 引言

本文在第二章中介绍过离散场景下一致并行机调度的几种传统鲁棒优化模型，包括最坏场景模型和最大后悔模型。然而因为过度关注目标性能表现最差的单一最坏场景，忽视了其他场景对性能造成的影响，导致该类模型存在一定的局限性，最终求得的鲁棒解保守性过高。同时在对最大后悔模型进行求解时需要获得每个场景下的最优性能，对于在确定性环境下已经是NP-hard问题的一致并行机调度来说，这无疑大大增加了求解难度。考虑到以上几点，本文尝试在离散场景的不确定一致并行机问题中求解一种全新的阈值坏场景集惩罚模型，通过关注更多坏场景下的性能来降低最终解的保守性。本文根据问题特点采用了一种面向问题的合并场景邻域构造方式(United-scenario Neighborhood, UN)，并将使用了该邻域的局部搜索与布谷鸟算法进行混合，用该基于合并场景邻域局部搜索的布谷鸟算法(UN based CSA, UNCSA)对这一鲁棒模型进行求解。

本章的3.2节将介绍一致并行机的阈值坏场景模型。3.3节对面向问题的合并场景邻域进行介绍，详细说明采用该邻域构造方式的原因和邻域构造的步骤。3.4节对本文提出的UNCSA算法的步骤和算法组成部分进行详细的说明。3.5节是仿真和分析，将UNCSA算法和已有的其他三种智能优化算法进行对比，以此来验证算法和模型的有效性。3.6节对本章的内容进行概括和总结。

* 1. 离散场景下一致并行机阈值坏场景集惩罚模型

本文研究用有限个离散场景表示不确定加工时间的一致并行机调度问题(Scenario Identical Parallel Machine Scheduling, SIPMS)。2.4节中对传统的SIPMS下的鲁棒模型WCM和WRM进行了介绍，通过关注单一最坏场景下的性能或者最大遗憾值来获得具有抗风险能力的鲁棒解。本文在此基础上关注更多性能表现较差的场景，以这些场景下的性能为鲁棒衡量指标，设计新的用于SIPMS的鲁棒模型。

根据Wang等[45]的研究内容，首先给出以下的定义：

**定义 3.1** 在SIPMS中，同一个可行解在不同的场景下表现出不同的性能。给定一个阈值性能，令的表现性能劣于的场景被称作的一个坏场景。所有坏场景的集合被称为的阈值坏场景(Threshold Bad-scenario Set, TBS)，记作：



从TBS的定义可以看出，对于一个可行解，其最坏场景也包含在中。

将每个坏场景下的性能与阈值的差值的平方项作为坏场景没达到标准的惩罚值。将在所有坏场景中的惩罚值相加，定义阈值坏场景集惩罚值：



本文以优化为鲁棒指标，定义了一个阈值坏场景集模型PTM (PT Model)：

(PTM):



从式和可以看出，对于PTM，阈值作为衡量坏场景的标准，体现了决策者的主观偏向。根据Daniels等[74]和Wang等[45]的建议，当阈值的取值不优于所有场景下的最优期望性能时PTM模型可以表达决策者抵御坏场景下不合格性能的抗风险偏向。本文研究的均为出现概率相同的“纯场景”，根据2.3节的介绍，在“纯”场景下的期望性能可以表示成均值性能。用表示最佳均值性能，即当时，求解SIPMS下的PTM可以获得抵御坏场景下未达标性能的鲁棒解。

相比WCM模型，PTM模型关注包括最坏场景在内的更多坏场景，因此得到的鲁棒解的保守性大为降低。同时又不需要求解每个场景下的最优性能，相比于WRM模型，降低了求解的难度。

在本章中将先假定阈值对于决策者是先验的，本文会在第四章的两阶段阈值坏场景集模型的求解过程中给出其具体求解方法。

* 1. 面向坏场景集惩罚模型的合并场景邻域

本章将采用布谷鸟算法混合局部搜索的方式对SIPMS的PTM进行求解。而邻域构造是局部搜索中的关键步骤。PTM模型涉及多个坏场景，而单一场景下的SIPMS问题可以看作是确定性一致并行机问题。因此本文先参考已有的确定性一致并行机调度问题设计单一坏场景下的邻域构造方式，再根据SIPMS的PTM问题的特点，设计一种面向问题的合并场景邻域构造方式用于局部搜索。

在第二章的2.7节中，本文介绍了一致并行机调度问题中常用的邻域构造方式。这几种方式通常随机选择进行交换或者插入的工件。相比于其他的并行机调度问题，一致并行机调度中同一个工件在不同的机器上的加工时间是保持不变的。根据这个特点，本文在原有的邻域构造方式基础上进行拓展，设计了两种面向一致并行机调度问题的邻域构造方式，更加有针对性的选择进行插入和交换的工件。

第一种邻域构造基于原本的插入方式。在进行插入操作时，选择关键机器上加工时间最短的，插入到当前总加工时间最短的非关键机器的加工队列中进行加工。第二种邻域构造采用一对一交换的方式，选择关键机器上加工时间最长的，与总加工时间最短的非关键机器上加工时间最短的进行加工位置的交换。

以图2.4中的当前解的加工序列为例，采用本文提出的方式选择工件进行插入或交换产生邻域解。 根据表2.3的各工件加工时间，关键机器为，加工时间最短的非关键机器为。则选择上加工时间最短的，插入到上进行加工。而在进行交换操作时，选择关键机器上加工时间最长的，与上加工时间最短的进行交换得到邻域解。

图3.1展示了利用本文设计的方式选择工件产生邻域解的过程。从图中可以看出邻域解相比当前解在最大完工时间上的改善。相比随机选择工件，本文提出的操作方式能让加工时间的分配更加均匀，从而减少当前解的最大完工时间，产生相对优秀的邻域解。

本文求解SIPMS下的PTM问题，需要考虑同一个解在不同坏场景下的表现性能。但不同场景下的工件有着不同的加工时间，导致了同一个解在不同场景下呈现出不同的关键机器，即使是相同的操作也会产生完全不同的邻域解。

表3.1是一个7工件3机器的一致并行机问题在两个场景下的加工时间，加工序列以图3.1中当前解为例。图3.2展示了不同场景下进行的插入操作。在下，关键机器为，对应的加工时间最短的工件为，插入操作是将移动到总加工时间最短的。在下的关键机器为，则选择将上的插入到总加工时间最短的上加工。

图3.3展示了不同场景下的交换操作。在下，选择关键机器上加工时间最长的，与上加工时间最短的交换加工位置。而在的关键机器为，总加工时间最短的非关键机器为，则将上加工时间最长的工件与上加工时间最短的交换加工位置。



通过图3.2和图3.3，可以清楚的看到即使采用同一种邻域操作规则，同一个当前解在不同的场景下也会产生完全不同的邻域解。因此在求解PTM模型时需要充分考虑不同场景对邻域操作造成的影响。

Wang等[45]在求解离散场景下的作业车间调度问题时，在单一场景下基于关键路径这一概念来构造邻域。与关键机器相似，在不同的场景下同一个调度解会呈现出不同的关键机器导致产生完全不同的邻域解。为了克服不同场景对邻域解造成的影响，他们提出了合并场景邻域这一概念，将所有单一场景下的邻域解集合起来形成合并场景邻域。

**定义3.2：**对于一个解，其在场景下产生的邻域解用表示，合并场景邻域表示在场景集下的所有场景产生的邻域集合：



其中不同场景下产生的重复邻域解需要被删除。

参照这一做法，本文在上文提出的两种单场景一致并行机邻域解构造方式的基础上，根据PTM模型的特点，围绕坏场景集建立面向SIPMS下PTM问题的合并坏场景邻域。

本文中用分别表示在场景下采取插入(Insert Operation)和交换操作(Swap Operation)后产生的邻域解，则。在其坏场景集下的合并坏场景邻域用表示：



同样在中，不同坏场景下产生的重复候选邻域解会被删除。

整个基于坏场景集产生合并坏场景邻域的伪代码如图3.4所示：

相比单一场景下的邻域构造，合并坏场景邻域将不同坏场景下的邻域综合起来进行考虑。这一操作方式扩大了每代邻域解的个数，同时充分考虑了不同场景对邻域操作造成的影响，使算法在局部搜索的过程中有更大的几率获得性能更好的邻域解。

* 1. 基于合并坏场景集邻域的混合布谷鸟算法(UNCSA)

布谷鸟算法最初的设计多用于求解连续优化问题[69]。自Tein等[75]将布谷鸟算法进行改进用于解决护士排班问题后，布谷鸟算法开始在离散优化问题尤其是调度问题中得到广泛的应用[23][25]。在用于调度问题的求解时，通常需要加入离散化的步骤来实现离散调度解和连续位置解之间的转换。离散化常用的方法有最大位置规则（Largest Position Value, LPV）[76],最小位置值规则（Smallest Position Value，SPV）[58]等。本文将采用SPV规则来对布谷鸟算法进行离散化的操作，SPV规则操作简单，已经被成功用于多种调度问题[77]-[79]，具体的操作方式会在下文进行详细介绍。

作为一种全局搜索能力较强的智能优化算法，布谷鸟算法中用于个体位置更新的步长服从莱维分布，这使得个体容易在较大的搜索区域内进行跳跃。虽然在每次迭代过程中会筛选出部分差解进行更新，但采用的是随机产生新解进行替换的方式，这也导致了算法的局部搜索能力较差。因而越来越多的研究人员尝试在布谷鸟算法中加入局部搜索的步骤，在保留种群中优秀个体的同时对性能较坏的个体的邻域进行深度搜索，进而提升整个种群的性能，将整个算法引向更好更广的搜索空间。混合算法通过结合两种搜索方式的特点，形成优势互补，相比单一的算法有着更高的搜索质量和效率。

本文将先对布谷鸟算法进行离散化的处理，使其能够用于一致并行机调度问题的求解，然后根据3.3节的内容，采用一种面向SIPMS下的PTM问题的合并坏场景邻域构造方式，将使用了这种邻域构造方式的局部搜索与离散化后的布谷鸟算法混合用于求解。接下来将对这种混合了基于合并场景邻域局部搜索的布谷鸟算法(UN based CSA, UNCSA)进行详细的介绍和仿真测试。

* + 1. 编码和解码

在一致并行机调度问题中，主要有基于机器的编码[22](Encode Based on Machine)和基于工作的编码[58](Encode Based on Job)两种编码方式。基于机器的编码中的每一位数码表示具体的机器序号，因此会出现重复数码的情况。而本文采用的布谷鸟算法后续会加入离散化的步骤，需要解中每个维度的数码都不同，因此本文采用基于工件的编码方式。

对于一个涉及台机器，个工件问题的一致并行机问题采用基于工件的编码。机器为，工件为。每个解由位整数组成，每个数字代表对应工件的下标。在同一台机器上加工的工件按照加工顺序组成一个子集，所有子集按机器序号排列成一个解。

以表3.1所示的7工件3机器一致并行机调度问题为例，如果一个调度解为，则表示工件在上加工，在上加工，在上加工。其对应的在下的生产加工甘特图如图3.5所示：

* + 1. 初始种群的产生

对于群智能优化算法，良好的初始解可以提升求解的效率。LPT规则作为一种启发式规则，可以快速产生相对优秀的一致并行机调度解，在用智能优化算法求解一致并行机调度问题时常被用作产生初始解[25]。

LPT规则在一致并行机调度中的使用方式比较简单，即将所有工件按照加工时间非增序进行排列，然后每次将当前加工时间最长的工件安排到序号最小的空闲机器上进行加工。

以表3.1的加工时间为例。在场景下，七个工件按加工时间非增序进行排序如下：。一开始三台机器都处于空闲状态，先将放置在上进行加工，然后和分别在上进行加工。根据各工件的加工时间，最先进入空闲状态，下一个待加工工件放置上加工，和分别在上加工。轮到最后一个工件时，都处于空闲状态，优先放置在序号较小的上加工。在下利用LPT规则得到的调度解为，对应的甘特图如下所示：

通过类似的操作，在下利用LPT规则得到的调度解为，对应的甘特图如图3.7所示：

本文求解个离散场景下的一致并行机调度问题，混合布谷鸟算法每代的种群大小为。如果算法设定的种群大小场景个数时，选择在前个场景下利用LPT规则产生的解作为初始的种群。如果种群大小>场景个数,则前个初始解利用LPT规则生成，后个解通过对前面已产生的初始解进行扰动产生。这样可以根据问题背景产生相对优秀的初始解。

* + 1. 连续化和离散化

布谷鸟算法的最初设是为了求解连续优化问题，而一致并行机调度问题的最终解为具体的工件排序，属于离散的优化问题。因此本文在标准的布谷鸟算法基础上加入了连续化和离散化的步骤，以此来实现离散的调度解和布谷鸟算法运行过程中的连续解之间的转化。

本文中用表示每代的调度解种群，其中的每个个体为一个离散的调度解。用表示连续位置解种群，里面的每个连续解个体与一一对应。在使用LPT规则生成初始调度解种群后，将其连续化得到初始的连续解种群。在混合布谷鸟算法之后的迭代过程中，对每代的连续解种群中的个体采用莱维飞行更新得到下一代的，再通过对进行离散化得到下一代的调度解种群。

对于初始调度解种群中的个体，其对应的连续解通过下式计算得到：



本文将连续化的结果保留3位有效数字。以一个调度解为例，按式进行计算，则。

最终对应的连续解为。

每一代中的连续解通过离散化转为调度解的步骤根据SPV规则实现。SPV规则的操作步骤比较简单，对于一个连续解，先将的各位数值按照升序进行排序，排序后位于第位的数值在原本中的位置序号即为对应的离散解的。表3.2为SPV规则应用的一个具体例子，假设某个连续解，先对各位数值按升序进行排序，得到b)表所示的结果。排序后的第一位0.16，为中的数值，则对应的离散调度解的第一位，排序后的第二位为原本中的数值0.59，则。以此类推，则该连续解利用SPV规则得到的离散调度解。

* + 1. 基于合并坏场景邻域的局部搜索

在3.3节中对本文设计的面向SIPMS下的PTM问题的合并坏场景邻域构造进行了介绍，本小节中介绍混合布谷鸟算法UNCSA中对该邻域进行局部搜索的具体步骤。

在UNCSA中，每一次迭代过程中会筛选出性能表现较差的后的解。用表示一个坏解，首先确定的坏场景集，然后根据3.3中的合并坏场景邻域产生步骤形成在其所有坏场景下的合并坏场景邻域。接着将中的所有邻域解按性能排序，选出其中表现性能最好的邻域解，记作。给定一个概率，如果邻域解的性能优于进行局部搜索的当前解，则用代替。否则以的概率将更新为。

这一利用基于合并坏场景邻域的局部搜索步骤记作(Local Search based UN)，整个局部搜索的具体操作如图3.8的伪代码所示：

* + 1. 终止准则

在智能优化算法中，终止准则起着平衡求解效率和求解结果的作用。合适的终止准则可以使算法在合理的时间成本消耗下获得性能相对优秀的解。本文采用的UNCSA算法的主框架是布谷鸟算法，参考其他已有的布谷鸟算法[25][59]，采用最大迭代次数作为算法终止条件。但具体的数值需要结合实际问题的计算效果才能决定，本文会在下文的仿真实验中确定的具体数值。

* + 1. 算法步骤

综合上面几小节的内容，整个用于求解SIPMS下的PTM问题的UNCSA算法总流程图如图3.9所示，包含了以下几个步骤：

Step1：输入阈值，场景集，初始化迭代次数，最大迭代次数等算法参数。

Step2：利用3.4.2节的方法产生初始调度解种群，并记录下种群中的最优解。将中的每个个体进行连续化得到初始的连续解种群。

Step3：利用莱维飞行对当前代的连续解种群进行更新产生下一代连续种群。同时同使用SPV规则得到下一代调度解种群。

Step4：计算中各解的性能，筛选出性能较差的后个坏解。计算每个坏解的坏场景，构造其合并坏场景邻域。

Step5：对每个坏解的合并坏场景邻域执行进行局部搜索，更新。并将对中调度解个体所作的交换插入操作同步到的对应个体中，对也进行更新。并根据更新后的种群判断是否要更新最优解。同时迭代次数。

Step6：重复Step3~Step5，直到满足终止准则后停止迭代输出最终结果。

* 1. 仿真实验与分析

本节中将采用UNCSA算法来求解离散场景下加工时间不确定的一致并行机调度问题的阈值坏场景集惩罚模型，优化目标为所有坏场景下的惩罚值之和。首先将对算法和算例的相关参数进行测试设置，然后将UNCSA算法与采用了普通局部搜索的布谷鸟算法ICSA[25]，采用了遗传算法加局部搜索的GA+LS[47]，还有传统的模拟退火算法SA[42]进行比较，通过大量的仿真实验数据来展示本文提出的UNCSA算法在求解SIPMS下的PTM问题时的有效性和优势。

本文中的所有仿真程序均采用C++语言编写实现，在Windows10操作系统环境下采用Visual Studio 2017软件运行。仿真所用计算机的CPU为Intel(R) Xeon(R) Gold 5188 ,运行内存为256G。

* + 1. 算例设置

参照文[11][16][25]中的测试算例设置，本文设置了30种不同规模()的算例，并将其分为三种类型：小规模算例,中规模算例和大规模算例，每种类型包含10个不同规模的算例。同时本文还将参考Kouvelis等[80]产生离散场景的方式来对加工时间进行不确定化处理。每个场景下工件的加工时间从区间中产生，其中为控制加工时间取值的参数，在中选取。测试算例的规模和加工时间的设置如表3.3所示：

每个算例的加工时间涉及16个场景，即。每组算例组合进行20次测试，一共进行2400个算例测试。

* + 1. 算法参数设置

整个UNCSA算法需要设置以下几个算法参数，每代的种群大小，莱维飞行时的步长因子，计算莱维飞行时所用的常数，每次迭代后筛选出的坏解百分比，局部搜索时接受性能较差的邻域解的概率以及最大迭代次数。

考虑到初始解的产生方式，本文中的种群大小设置为和场景数相同，即，参照文[55][59]中对布谷鸟算法参数的设置，步长因子，。每代选择的坏解进行局部搜索。局部搜索时性能较差的邻域解接受概率。而对于最大迭代次数的设定，在下文中会通过具体的仿真确定。

* + 1. UNCSA算法终止准则对求解结果的影响

根据上一节的算法参数设置，本节中将对UNCSA的终止准则开展研究。从三种规模类型中各选取两个规模的算例，记录使用UNCSA在不同迭代次数下对PTM模型的求解结果。对于每组参数组合的20次仿真测试，选择目标函数性能值最优的那次进行展示。测试的结果如表3.4所示，其中的为不同参数组合对应的阈值，为使用UNCSA求解PTM模型得到的最优解对应的性能值，为整个求解过程消耗的CPU时间。

从表3.4中可以看出，在不同的问题规模下，最大迭代次数的增加会增加UNCSA算法求解过程中的消耗时间，但同时也会提升UNCSA算法的最终求解结果的性能。在迭代次数相对较少时，UNCSA的求解结果随着代数的增加改善比较明显。而在算法运行到一定的迭代次数后解的改善开始变得缓慢，此时再继续运行UNCSA，只会增加大量的时间消耗而不能显著地提升求解效果。为了更清晰地观察这种趋势，本文从三种不同规模类型中分别挑选了三个规模的算例，绘制了其在时间取值参数时CPU消耗时间随迭代次数的变化曲线以及值和迭代次数的收敛图，如图3.10~图3.15所示。通过观察CPU消耗随迭代次数变化的曲线可以得出，在不同规模的算例中，随着迭代次数的增加，UNCSA算法完成求解需要的时间基本呈线性增加。而结合表3.4和不同算例的求解过程收敛图可以发现，最终的求解结果大约在250代左右开始进入收敛。此时继续运行算法会牺牲大量的CPU时间但并不能对最终的求解结果进行明显的改善。将UNCSA的最大迭代次数设置为250代以上并没有太大意义。综合考虑求解质量和CPU时间的消耗，本文中的UNCSA算法的终止准则将设置为最大迭代次数。

同时，通过观察表3.4还可以发现，参数对算法的影响主要表现为通过改变加工时间的取值范围来改变最终解的目标函数值。而对于算法本身而言，虽然的增大会使得工件的加工时间取值分布空间范围增大，增加整个系统的不确定性，从而略微增加CPU时间的消耗，但对于实际的算法运行趋势和规律没有影响。











* + 1. 四种算法求解对比

为了测试UNCSA算法在求解PTM模型时的有效性和优势，本文将采用UNCSA算法和其他三种求解一致并行机问题的智能优化算法一起进行求解对比：Laha等【】设计的加入了普通局部搜索的布谷鸟算法(ICSA)，Lee等【】设计的加入了局部搜索的遗传算法(GA+LS)，还有Xu等【】采用的模拟退火算法(SA)。其中UNCSA算法的参数设置如3.5.2节所示，并根据上一节的测试结果，最大迭代次数。其余算法的参数设置参考各自的文献。

根据3.5.3节的仿真结果，参数对算法的运行趋势和规律没有影响，因此在下文中，将只对某个固定下的实验结果进行展示分析。

表3.5展示了在时不同算法对不同规模算例的求解情况。

对于每组参数组合的20次仿真测试，选择目标函数性能值最优的那次进行展示.对比了不同算法求解所用的CPU时间和最终求得的值。从表中可以看出，在求解规模较小，如时各算法的求解结果还未看出明显的差距。但整体来看，在对各种不同规模的算例求解中，UNCSA无论在求解时间还是求解结果上都领先于其他算法。相比于同样在布谷鸟算法中加入局部搜索的ICSA，UNCSA能够在更短的时间内求得更好的值。

这一结果说明本文使用的面向PTM问题的合并场景邻域比普通邻域构造对解空间的挖掘更为合理，因此提升了UNCSA的局部搜索质量，使其能在每一次的迭代过程中对解空间进行更加全面深入的搜索。作为同样采用群智能优化算法混合局部搜索的GA+LS算法，虽然比起SA算法能够求得更好的，但其在四种算法中花费了最多的求解时间。而且这一时间上的差距随着问题规模的增加更加明显。这一结果反应了GS和CSA求解效率之间的差距，CSA能够更加快速高效地求解更为复杂的调度问题。而作为典型的串行邻域搜索算法，SA的每次迭代过程只涉及单一解的邻域搜索，因此相比GA+LS消耗相对较少的CPU，求解时间与ICSA相近。但其对解空间进行的探索范围有限，在所有测试算法中最后的求解结果性能最差。

为了更清楚地反应UNCSA相比其他算法的求解优势，本文根据三个规模算例求解的20次结果绘制了箱型图，如图3.16~3.18所示。从图中可以看出，UNCSA算法的求解结果在最优解或是多次求解的平均结果上都明显优于其他三种算法，其次是ICSA算法和加入了局部搜索的遗传算法GA+LS，SA的求解能力最差，所得解与其他算法之间有着明显的差距。通过观察箱型图中解的分布情况，可以发现UNCSA算法所求结果分布稳定，在对同一规模算例的多次运行中求得的最优解和最劣解差距相对不大。而其他三种算法在求解时结果的上下界能看到明显的跨度，甚至在部分规模的求解中还出现了与平均求解结果差距过大的异常值（如ICSA算法对算例的求解，GA+LS算法对算例的求解，SA对算例的求解）。这一情况表明UNCSA算法的搜索方式稳定有效，在多次的实验中都能保证求得相对优秀的结果。

以上的仿真结果充分体现了UNCSA算法相较其他算法在求解PTM问题时的优势。

* 1. 本章小结

本章研究了加工时间不确定的鲁棒一致并行机调度问题，利用离散场景来描述不确定的加工时间。本章中先给出了阈值和坏场景集的定义，然后在此基础上建立了离散场景下一致并行机阈值坏场景集惩罚模型PTM。为了求解该模型，本章设计了一种基于合并坏场景集邻域的混合布谷鸟算法UNCSA。在传统的布谷鸟算法框架上加入了离散化的步骤使其能用于求解一致并行机调度问题。为了进一步提高并加入了局部搜索的步骤，将布谷鸟算法强大的全局搜索能力与局部搜索相结合，混合不同搜索方式的优点来提升算法的求解效率。同时设计了一种面向问题的合并坏场景邻域用于局部搜索，根据问题特点产生更加有针对性的邻域解来提升局部搜索的效率。

本章对所提出的UNCSA算法进行了大量的仿真测试，针对不同规模的一致并行机调度问题进行了求解分析。首先测试了算法求解时的终止条件，之后将UNCSA算法与已有的三种求解一致并行机调度问题的智能优化算法，包括加入了普通局部搜索的布谷鸟算法ICSA，加入了普通局部搜索的遗传算法GA+LS，还有模拟退火算法SA进行了求解对比。利用四种算法求解离散场景下一致并行机调度问题的PTM模型，通过比较求解时间和最终解的性能验证了本文设计的UNCSA算法在求解该模型时的优势，也表明了本文采用的基于合并坏场景的邻域构造方式的有效性和实际价值。

1. 一致并行机两阶段阈值坏场景集模型及其求解算法
   1. 引言
   2. 离散场景下一致并行机两阶段阈值坏场景集模型
   3. 一致并行机两阶段阈值坏场景集模型的求解算法
      1. 求解第一阶段的割平面法
      2. 求解第二阶段的混合布谷鸟算法
   4. 仿真实验与分析
      1. 第一阶段割平面法的性能测试
      2. 两阶段阈值坏场景集模型的仿真测试
   5. 本章小结
2. 总结与展望

参考文献

1. Rodammer F A, Whit K P. A recent survey of production scheduling [J].IEEE Transactions on System Man and Cybernetic, 1998, 18(6):841-851.
2. Rocholl J , Mnch L . Decomposition heuristics for parallel-machine multiple orders per job scheduling problems with a common due date[J]. Journal of the Operational Research Society, 2019:1-17.
3. Lee C H . A New Discrete Electromagnetism-Like Mechanism Algorithm for Identical Parallel Machine Scheduling Problem with Eligibility Constraints in Metal Nuts Manufacturing[J]. Arabian Journal for Science & Engineering, 2017, 42(8):1-12.
4. Gungor, Murat, Unal, et al. A parallel machine lot-sizing and scheduling problem with a secondary resource and cumulative demand[J]. International Journal of Production Research, 2018.
5. Yamamoto M, Nof S Y. Scheduling/rescheduling in the manufacturing operation system environment [J].International Journal of Production Research, 1985, 23(4):705-722.
6. M ignon D J, Honkomp S J, Reklaitis G V. A framework for investigating schedule robustness under uncertainty [J]. Computer and Chemical Engineering, 1995, 19 (Suppl) :S615-620.
7. McNaughton R. Scheduling with deadlines and loss functions[J]. Management Science, 1959, 6(1): 1-12.
8. Dell’Amico M, Martello S. Optimal scheduling of tasks on identical parallel processors[J]. ORSA Journal on Computing, 1995, 7(2): 191-200.
9. Shim S O, Kim Y D. A branch and bound algorithm for an identical parallel machine scheduling problem with a job splitting property[J]. Computers & Operations Research, 2008, 35(3): 863-875.
10. Haouari M, Jemmali M. Tight bounds for the identical parallel machine‐scheduling problem: Part II[J]. International Transactions in Operational Research, 2008, 15(1): 19-34.
11. Mokotoff E. An exact algorithm for the identical parallel machine scheduling problem[J]. European Journal of Operational Research, 2004, 152(3): 758-769.
12. Dell'Amico M, Iori M, Martello S, et al. Heuristic and exact algorithms for the identical parallel machine scheduling problem[J]. INFORMS Journal on Computing, 2008, 20(3): 333-344.
13. Chen Z L, Powell W B. Solving parallel machine scheduling problems by column generation[J]. INFORMS Journal on Computing, 1999, 11(1): 78-94.
14. Garey M, Johnson D. Computers and intractability: A guide to the theory of NP-Completeness[M]. W. H. Freeman and Company 1979.
15. Graham R L. Bounds on multiprocessing timing anomalies[J]. SIAM journal on Applied Mathematics, 1969, 17(2): 416-429.
16. Lee W C, Wu C C, Chen P. A simulated annealing approach to makespan minimization on identical parallel machines[J]. The International Journal of Advanced Manufacturing Technology, 2006, 31(3-4): 328-334.
17. Coffman, Jr E G, Garey M R, Johnson D S. An application of bin-packing to multiprocessor scheduling[J]. SIAM Journal on Computing, 1978, 7(1): 1-17.
18. Lee C Y, Massey J D. Multiprocessor scheduling: combining LPT and MULTIFIT[J]. Discrete applied mathematics, 1988, 20(3): 233-242.
19. Gupta J N D, Ruiz-Torres A J. A LISTFIT heuristic for minimizing makespan on identical parallel machines[J]. Production Planning & Control, 2001, 12(1): 28-36.
20. Yamashita D S. Tabu search for scheduling on identical parallel machines to minimize mean tardiness[J]. Journal of intelligent manufacturing, 2000, 11(5): 453-460.
21. Alharkan I, Bamatraf K, Noman M A, et al. An order effect of neighborhood structures in variable neighborhood search algorithm for minimizing the makespan in an identical parallel machine scheduling[J]. Mathematical Problems in Engineering, 2018.
22. Min L , Cheng W . A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines[J]. Artificial Intelligence in Engineering, 1999, 13(4):399-403.
23. Kashan A H, Karimi B. A discrete particle swarm optimization algorithm for scheduling parallel machines[J]. Computers & Industrial Engineering, 2009, 56(1): 216-223.
24. Neto R F T, Godinho Filho M, Da Silva F M. An ant colony optimization approach for the parallel machine scheduling problem with outsourcing allowed[J]. Journal of Intelligent Manufacturing, 2015, 26(3): 527-538.
25. Laha D, Gupta J N D. An improved cuckoo search algorithm for scheduling jobs on identical parallel machines[J]. Computers & Industrial Engineering, 2018, 126: 348-360.
26. 顾幸生. 不确定性条件下的生产调度 [J]. 华东理工大学学报,2000,26(5),441-446.
27. Yamamoto M, Nof S.F. Scheduling/rescheduling in the manufacturing operating system environment [J].Int J Prod Res, 1985, 23 (4):7052722.
28. Honkomp S J，Reklaitis G V. A framework for investigating schedule robustness under uncertainty [J]. Computers and Chemical Engineering, 1995,19( Supp1):S6152620.
29. Alimoradi S , Hematian M , Moslehi G . Robust scheduling of parallel machines considering total flow time[J]. Computers & Industrial Engineering, 2016, 93:152-161.
30. Ranjbar M, Davari M, Leus R. Two branch-and-bound algorithms for the robust parallel machine scheduling problem[J]. Computers & Operations Research, 2012, 39(7): 1652-1660.
31. Cai X, Zhou S. Stochastic scheduling on parallel machines subject to random breakdowns to minimize expected costs for earliness and tardy jobs[J]. Operations Research, 1999, 47(3): 422-437.
32. Zadeh L A. Fuzzy sets as a basis for a theory of possibility[J]. Fuzzy sets and systems, 1978, 1(1): 3-28.
33. Balin S. Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation[J]. Information Sciences, 2011, 181(17): 3551-3569.
34. Yeh W C, Lai P J, Lee W C, et al. Parallel-machine scheduling to minimize makespan with fuzzy processing times and learning effects[J]. Information Sciences, 2014, 269: 142-158.
35. Behnamian J. Particle swarm optimization-based algorithm for fuzzy parallel machine scheduling[J]. The International Journal of Advanced Manufacturing Technology, 2014, 75(5-8): 883-895.
36. Anglani A, Grieco A, Guerriero E, et al. Robust scheduling of parallel machines with sequence-dependent set-up costs[J]. European journal of operational research, 2005, 161(3): 704-720.
37. Allahverdi A, Aydilek H. Heuristics for the two-machine flowshop scheduling problem to minimize maximum lateness with bounded processing times[J]. Computers & Mathematics with Applications, 2010, 60(5): 1374-1384.
38. Kouvelis P, Yu G. Robust discrete optimization and its applications [M]. Spring, 1997.
39. Li Z, Ierapetritou M G. Robust optimization for process scheduling under uncertainty[J]. Industrial & Engineering Chemistry Research, 2008, 47(12): 4148-4157.
40. Sarin S C, Nagarajan B, Jain S, et al. Analytic evaluation of the expectation and variance of different performance measures of a schedule on a single machine under processing time variability[J]. Journal of combinatorial optimization, 2009, 17(4): 400-416.
41. Liu M, Liu X, Zhang E, et al. Scenario-based heuristic to two-stage stochastic program for the parallel machine ScheLoc problem[J]. International Journal of Production Research, 2019, 57(6): 1706-1723.
42. Xu X, Cui W, Lin J, et al. Robust makespan minimisation in identical parallel machine scheduling problem with interval data[J]. International Journal of Production Research, 2013, 51(12): 3532-3548.
43. Drwal M, Rischke R. Complexity of interval minmax regret scheduling on parallel identical machines with total completion time criterion[J]. Operations Research Letters, 2016, 44(3): 354-358.
44. Wang S, Cui W. Approximation algorithms for the min-max regret identical parallel machine scheduling problem with outsourcing and uncertain processing time[J]. International Journal of Production Research, 2020: 1-14.
45. Wang B, Wang X, Lan F, et al. A hybrid local-search algorithm for robust job-shop scheduling under scenarios[J]. Applied Soft Computing, 2018, 62: 259-271.
46. Wang S, Zou H, Wang S. A Tabu-GA-based parallel machine scheduling with restrained tool resources[J]. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2020: 0954405420928691.
47. Lee W C , Wang J Y , Lee L Y . A hybrid genetic algorithm for an identical parallel-machine problem with maintenance activity[J]. Journal of the Operational Research Society, 2015.
48. Chen J, Pan Q K, Wang L, et al. A hybrid dynamic harmony search algorithm for identical parallel machines scheduling[J]. Engineering Optimization, 2012, 44(2): 209-224.
49. 何嘉. 异速并行机系统生产调度与预防性维护集成研究[D]. 重庆大学.
50. Cheng T C E, Sin C C S. A state-of-the-art review of parallel-machine scheduling research[J]. European Journal of Operational Research, 1990, 47(3): 271-292.
51. Edis E B, Oguz C, Ozkarahan I. Parallel machine scheduling with additional resources: Notation, classification, models and solution methods[J]. European Journal of Operational Research, 2013, 230(3): 449-463.
52. Mulvey J M, Vanderbei R J, Zenios S A. Robust optimization of large-scale systems[J]. Operations research, 1995, 43(2): 264-281.
53. Wang S , Cui W . Approximation algorithms for the min-max regret identical parallel machine scheduling problem with outsourcing and uncertain processing time[J]. International Journal of Production Research, 2020(1):1-14.
54. Drwal M, Roman, Complexity of interval minmax regret scheduling on parallel identical machines with total completion time criterion[J]. Operations Research Letters, 2016.
55. Yang X S, Deb S. Cuckoo search via Lévy flights[A].2009 World congress on nature & biologically inspired computing (NaBIC)[C]. Coimbatore, India: IEEE, 2009: 210-214.
56. Abdel-Basset M, Hessin A N, Abdel-Fatah L. A comprehensive study of cuckoo-inspired algorithms[J]. Neural Computing and Applications, 2018, 29(2): 345-361.
57. Valian E, Valian E. A cuckoo search algorithm by Lévy flights for solving reliability redundancy allocation problems[J]. Engineering Optimization, 2013, 45(11): 1273-1286.
58. Wang H, Wang W, Sun H, et al. A new cuckoo search algorithm with hybrid strategies for flow shop scheduling problems[J]. Soft Computing, 2017, 21(15): 4297-4307.
59. Guo P, Cheng W, Wang Y. Parallel machine scheduling with step-deteriorating jobs and setup times by a hybrid discrete cuckoo search algorithm[J]. Engineering Optimization, 2015, 47(11): 1564-1585.
60. Shehab M, Khader A T, Laouchedi M. A hybrid method based on cuckoo search algorithm for global optimization problems[J]. Journal of Information and Communication Technology, 2018, 17(3): 469-491.
61. Sheng Z, Wang J, Zhou S, et al. Parameter estimation for chaotic systems using a hybrid adaptive cuckoo search with simulated annealing algorithm[J]. Chaos: An Interdisciplinary Journal of Nonlinear Science, 2014, 24(1): 013133.
62. Payne R B, Sorensen M D. The cuckoos[M]. Oxford University Press, 2005.
63. Reynolds A M. Cooperative random Lévy flight searches and the flight patterns of honeybees[J]. Physics letters A, 2006, 354(5-6): 384-388.
64. Schreier A L, Grove M. Ranging patterns of hamadryas baboons: random walk analyses[J]. Animal Behaviour, 2010, 80(1): 75-87.
65. Ramos-Fernández G, Mateos J L, Miramontes O, et al. Lévy walk patterns in the foraging movements of spider monkeys (Ateles geoffroyi)[J]. Behavioral ecology and Sociobiology, 2004, 55(3): 223-230.
66. Austin D, Bowen W D, McMillan J I. Intraspecific variation in movement patterns: modeling individual behaviour in a large marine predator[J]. Oikos, 2004, 105(1): 15-30.
67. Atkinson R P D, Rhodes C J, Macdonald D W, et al. Scale‐free dynamics in the movement patterns of jackals[J]. Oikos, 2002, 98(1): 134-140.
68. Yang X S , Deb S . Engineering Optimisation by Cuckoo Search[J]. International Journal of Mathematical Modelling & Numerical Optimisation, 2010, 1(4):330-343.
69. Yang X S , Deb S . Cuckoo Search: Recent Advances and Applications[J]. Neural Computing & Applications, 2014, 24(1):169-174
70. Yang X S. Nature-inspired metaheuristic algorithms[M]. Luniver press, 2010.
71. Mantegna R N. Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes[J]. Physical Review E, 1994, 49(5): 4677.
72. 叶春明, 李永林, 刘长平. 新型仿生群智能算法及其生产调度应用[M]. 科学出版社, 2015.
73. 陈怡萍. 布谷鸟算法及应用研究[D]. 浙江大学, 2019.
74. Daniels R L , Carrillo J E . β-Robust scheduling for single-machine systems with uncertain processing times[J]. 1997, 29(11):977-985.
75. Lim H T, Ramli R. Recent advancements of nurse scheduling models and a potential path[J]. 2010.
76. Liang J J , Pan Q K , Tiejun C , et al. Solving the blocking flow shop scheduling problem by a dynamic multi-swarm particle swarm optimizer[J]. The International Journal of Advanced Manufacturing Technology, 2010.
77. Fatih Tasgetiren M, Liang Y C, Sevkli M, et al. Particle swarm optimization and differential evolution for the single machine total weighted tardiness problem[J]. International Journal of Production Research, 2006, 44(22): 4737-4754.
78. Tang L , Wang X . An Improved Particle Swarm Optimization Algorithm for the Hybrid Flowshop Scheduling to Minimize Total Weighted Completion Time in Process Industry[J]. IEEE Transactions on Control Systems Technology, 2010, 18(6):1303-1314.
79. Rui Z . An Artificial Bee Colony Algorithm Based on Problem Data Properties for Scheduling Job Shops[J]. Procedia Engineering, 2011, 23(1):131-136.
80. Kouvelis P, Daniels R L, Vairaktarakis G. Robust scheduling of a two-machine flow shop with uncertain processing times[J]. IIE Transactions, 2000, 32(5): 421-432.