求解阈值坏场景集一致并行机调度问题的改进布谷鸟算法

**摘要：**本文讨论了利用离散场景来表示不确定加工时间的一致并行机鲁棒调度问题。以一个阈值为基准筛选出每个调度解的坏场景集，通过求解阈值坏场景集惩罚模型来提升调度解对坏场景下不佳性能的抗风险能力。为了求解该模型，本文提出了一种基于合并场景邻域的改进布谷鸟算法(UNCSA)，在标准的布谷鸟算法中加入了邻域搜索的操作，并根据离散场景的特点采用了基于合并坏场景的邻域构造方式，以此来提升整个算法的局部搜索能力。本文进行了大量的实验，将所设计的算法与其他三种已有的智能优化算法进行仿真对比，最终的实验对比结果表明在对该鲁棒模型的求解上，本文设计的UNCSA算法以及采用的合并坏场景邻域相比其他算法和邻域构造方式更有效，可以获得更好的求解结果。

**关键词**：离散场景，鲁棒调度模型，一致并行机调度，布谷鸟算法，合并场景领域

1. 引言

一致并行机调度问题(identical parallel machine scheduling, IPMS)是一类常见的生产调度问题，涉及多台可同时进行工作且加工能力相同的机器。通过确定各个机器上的工件及其对应的加工顺序，来使得某项性能指标达到最优。一致并行机的加工特性符合多种生产场景，广泛地存在于日常的各种生产活动中[1]-[3]。对于一致并行机调度问题的研究通常在理想的确定性环境下进行，各种生产参数如加工时间等都是已知且固定不变的。然而由于存在加工熟练度，机器故障等各种不确定因素的影响，传统的确定性优化模型与实际加工模型间存在一定差距，此时求解确定性模型得到的方案难以达到生产预期。如何处理一致并行机调度中的不确定加工时间，建立合适的不确定模型并设计不确定环境下的高效算法逐渐成为了研究的热门。

场景方法是一种有效的处理不确定参数的方法。同随机分析法[4]和模糊分析法[5]不同，场景方法通过某个区间的连续场景[6]或者有限个离散场景的集合[7]来描述不确定性。各场景的分布相对独立，不需要预先确定具体的概率分布模型或者模糊规则，适合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性参数。目前在各类调度问题中有着广泛的应用[8]-[10]。

Li等[11]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，两种模型都有各自的决策偏向。基于场景的随机优化模型将整体期望性能作为优化目标[12]，考虑优化整个调度系统在所有可能场景下性能的平均水平。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。其中比较常见的有min-max模型(最坏场景模型)和min-max regret模型(最大后悔场景模型)。这两种模型通过优化所有场景下的最坏性能或者最大遗憾值来体现决策者的风险厌恶偏向，在调度类问题中应用广泛。如Kouvelis等[10]求解了一个双机器流水车间调度的最大后悔场景模型，同时采用了区间场景和离散场景表示不确定的加工时间。Drwal[13]研究了区间场景加工时间的单机调度问题，优化目标是加权滞后工件数的最大遗憾值。在这两种传统鲁棒优化模型的基础上，Wang[16]等提出了一种阈值坏场景集鲁棒优化模型。该模型以一个基准性能为阈值，筛选出性能较差的坏场景，对坏场景施加一定的惩罚，将所有坏场景下的惩罚值作为优化目标。并将这类阈值坏场景集惩罚模型成功应用于不确定作业车间调度问题中。实验仿真结果表明求解该模型能够获得抵御坏场景下不佳性能的鲁棒解。

目前对于鲁棒一致并行机调度问题研究仍以优化最坏场景下的性能为主。如Xu等[6]对区间场景下的一致并行机最大makespan遗憾值进行优化，采用了松弛迭代法和模拟退火算法(simulated annealing, SA)来求解不同规模的问题。Wang等[14]研究了可外包的一致并行机调度问题，目标函数是外包消耗和非外包加工花费的生产总成本。通过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。Drwal等[17]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。这类传统的鲁棒优化模型过度关注性能表现最差的单一最坏场景，而忽视了其他场景的影响，导致最终的求解结果过于保守。同时该类模型需要求解每个场景下的最佳性能，增加了求解的难度。考虑到以上几点，本文尝试在鲁棒一致并行机调度中求解Wang等[16]提出的阈值坏场景集优化模型，通过关注更多坏场景下的性能来降低保守性，这也将是对鲁棒一致并行机调度问题的一次全新探索与挑战。

本文要求解的鲁棒模型旨在对抗一致并行机的makespan在离散场景下性能降低的风险。在对一致并行机makespan的求解中，最初有学者采用一些精确算法进行求解。如Dell等[20]提出了一种分支定界算法来求解最小化makespan的一致并行机问题，Mokotoff[21]将优化makespan的一致并行机调度问题用数学规划的形式进行描述并设计了一种割平面法来进行求解。随着求解规模的增加，越来越多的研究人员开始采用高效的智能优化算法进行求解。如Kashan和Karimi[22]设计了一种离散粒子群算法(discrete particle swarm optimization, DPSO)用于求解目标函数为makespan的一致并行机问题，Kim[23]分别采用了遗传算法(genetic algorithm, GA)和模拟退火算法求解具有准备时间以及工件可拆分的一致并行机调度问题。

随着研究的逐渐深入，为了进一步提高求解效率，研究人员开始尝试将多种算法结合或者在常规的智能算法上加入局部搜索的步骤来对一致并行机调度问题进行求解。Behnanmian[24]在粒子群算法中加入了遗传算法的交叉变异操作，提出了一种离散粒子群算法DPSO用于求解加工时间符合钟形分布的不确定一致并行机调度问题，目标函数为最大完工时间makespan。Chen等[25]在采用和声搜索算法(harmony search, HS)求解一致并行机调度问题时，加入了变邻域算法来提升局部搜索的能力。Laha等[32]和Guo等[29]都采用了布谷鸟算法(cuckoo search, CS)对一致并行机调度问题进行求解，并均在布谷鸟算法中加入了局部搜索产生邻域解的步骤。Lee等[30]在求解最小化有维修行为的一致并行机调度问题时，提出了一种混合算法，在遗传算法的基础上加入了局部搜索（GA+LS），以此来提升求解效率。然而这些算法的局部搜索部分中的邻域解都由一致并行机调度中通用的邻域构造方式产生，尚未有根据求解问题的特异性设计的邻域构造方式出现。

根据一致并行机调度问题的研究现状，本文将对不确定环境下的一致并行机调度问题开展研究工作。本文采用离散场景来描述不确定的加工时间，求解一致并行机的阈值坏场景集鲁棒惩罚模型。为了更好地求解该模型，本文设计了一种改进的布谷鸟算法UNCSA（united scenario cuckoo search algorithm），通过在布谷鸟算法中加入基于合并坏场景的邻域构造方式，提升了整个算法在坏场景集下的局部搜索能力。根据了解到的研究现状，该鲁棒模型和邻域构造方式都是首次应用于一致并行机鲁棒调度问题。

本文将通过大量的实验仿真结果以及与其他算法的对比展示本文所设计的算法在求解该模型上的高效性。

本文的剩余部分安排如下：在第二节将对一致并行机的阈值坏场景集模型进行描述，第三节中将详细介绍本文设计的UNCSA算法，第四节中是算法的仿真和对比，第五节是总结与展望。

1. 问题描述

一致并行机调度问题一般描述如下：个工件，台加工速率相同的机器，任意一个工件可在任意一台机器上进行加工，工件之间互相独立，没有优先级。一台机器同一时刻只能加工一个工件，并且一旦开始加工就无法中途停止。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。表示第个工件的加工时间，与机器无关，无论工件在哪一台机器上加工均保持不变。为所有调度解的集合，表示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：



一致并行机调度中最常见的目标函数为所有机器中最长的完工时间makespan，记作：



当所有工件的加工时间都确定时，这样的问题被成为确定性一致并行机调度问题。

本文研究用有限个场景表示不确定加工时间的一致并行机问题(scenario identical parallel machine scheduling, SIPMS)，其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。

本文研究在坏场景集下一致并行机调度解的性能。参考文[16]，对于一个调度解，给定一个基准性能，或称为阈值，场景集中令表现性能劣于的场景即为的一个坏场景，所有坏场景的集合被称作的坏场景集，记作:



为了对抗坏场景下性能降低的风险，Wang等[16]对无法达到预期的坏场景性能施加了一定的惩罚值，定义了阈值坏场景集惩罚值，将所有坏场景下实现的性能与阈值的差值的平方项作为度量：



本文求解的阈值坏场景集惩罚模型PTM（PT model）则是将作为最终的优化目标：

(PTM):



由式和式可以发现，阈值作为衡量坏场景的标准，体现了决策者的主观偏向。其值可以凭经验或者其他方式获得，在本文中不给出具体的求解方法。根据Wang等[16]的建议，阈值的合理取值范围应不优于所有场景下的最佳期望性能。即当时，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：



确定性一致并行机问题是NP-hard问题[18]，可以看作SIPMS在单一场景下的一种特例。而在其基础上建立的涉及多个场景的PTM模型必定也为NP-hard问题[19]。为了在合理的时间内对PTM问题进行求解，本文根据模型的特点采用了一种基于坏场景的合并场景邻域构造方式，并将其与布谷鸟算法结合，设计了一种UNCSA算法。在下一节中将对该邻域构造方式和算法进行详细的介绍。

1. 求解算法

布谷鸟算法是近些年来较为新颖的群智能优化算法之一，由Yang和Deb[36]在2009年提出。其通过模仿布谷鸟在自然界中寻找其他鸟窝筑巢生蛋的行为，来模拟寻找优良性能解的过程。作为一种比较新颖的群搜索智能优化算法，布谷鸟算法能够对各种复杂的解空间进行较为全面的搜索，在多种优化问题的求解上都表现出了优秀的求解性能[29]-[34]。本文求解的PTM模型相较传统的鲁棒调度模型更为创新，鲁棒优化目标更为复杂。考虑到求解算法的求解效率，本文选择更为新颖高效的布谷鸟算法作为求解PTM的基础算法框架。

布谷鸟算法的步骤相对简单。在布谷鸟算法中，用表示当前的迭代代数，表示当前代种群，每代种群包含个解。初代种群随机生成，之后利用莱维飞行对中的每个个体进行更新，如果更新后的个体的性能优于原个体，则进行替换，否则保留原解。莱维飞行更新公式如下所示：



其中为步长因子，为固定的常数[36]，通常取。是服从莱维分布的随机步长，采用Mantegna算法进行计算：



是位于之间的参数，一般取，服从正态分布，为标准的Gamma函数：





利用莱维飞行完成每一代种群的更新后，计算新一代的种群中每个解的性能并进行排序，选出部分性能表现较差的坏解(数量占种群数的25%~30%)，每个坏解会被随机产生的新解代替。

以上步骤会重复进行，直到达到最大迭代次数时停止。

在以上介绍的标准布谷鸟算法求解步骤中，依靠莱维飞行的种群更新方式，布谷鸟算法相比其他群搜索智能优化算法，如遗传算法，粒子群算法，人工蜂群算法等在求解优化问题时具有更强的求解能力[37]-[39]。然而在布谷鸟算法的其余步骤中依旧存在可以改进的空间。为了提升布谷鸟算法局部求解的能力，越来越多的研究人员开始尝试在布谷鸟算法中加入局部搜索的步骤，来取代原本利用随机解代替坏解的过程[29]-[32]。受到这些改进算法的启发，本文也将在保留了布谷鸟算法莱维飞行这一优秀搜索机制的基础上，往传统的布谷鸟算法里加入局部搜索的环节。

作为局部搜索中的重要组成部分，合理的邻域构造方式可以提升局部搜索的效率和质量。本文的优化目标是离散场景下一致并行机的PTM模型，若使用常规的邻域产生方式，很难在已有的算法上获得进一步的突破。已有的对并行机问题的邻域构造方式通常围绕关键机器进行，随机选择关键机器上的工件进行交换插入等邻域操作。本文根据一致并行机与其他并行机调度问题的区别，设计了新的单场景下用于一致并行机的领域解产生操作。同时在此基础上，根据求解的PTM模型的特点，特意采用了合并坏场景邻域构造方法来处理不确定参数对邻域构造带来的影响，这一邻域构造方式也将是本文所设计算法的核心和主要创新点。

在本节接下来的部分，将对该合并坏场景邻域构造方式以及采用了该方式进行局部搜索的布谷鸟算法UNCSA进行详细的介绍说明。

* 1. UNCSA算法中的邻域构造方式
     1. 单场景下的一致并行机邻域构造

确定性一致并行机问题可以看作是单场景下SIPMS的特例，因此单场景下一致并行机的邻域构造方式可以参考已有的确定性一致并行机调度问题。在确定性并行机调度问题中，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[6]。并行机问题的邻域通常围绕关键机器进行构造。其中的邻域操作可以分为交换(swap)和插入(insert)[25]，而交换又可以分为一对一交换(swap)，非对称交换(asymmetric swap)和两两交换(double swap)。以一个3机器7工件规模的问题为例，为关键机器，和为非关键机器，图1展示了四种邻域构造方式。如图1所示，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；非对称交换下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。

以上的邻域构造方式，通常都是在关键机器上随机选择工件进行交换或者插入操作。考虑到相比于其他类型的并行机问题，在一致并行机调度中，同一个工件在不同机器上的加工时间是固定不变的。根据这个特点，本文设计了新的两种邻域构造方式，采用更有针对性的方式选取插入和交换的目标工件。

第一种邻域构造在进行插入时，选择关键机器上加工时间最短的，插入到总加工时间最短的非关键机器上。第二种邻域构造在进行交换操作时，选择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交换。

相比于随机选择工件，这样的操作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。

* + 1. 求解PTM模型的合并坏场景邻域构造

上一小节介绍了本文设计的单场景下一致并行机的邻域构造方式。但本文求解的PTM模型涉及多个坏场景，而同一个解在不同场景下会呈现不同的关键机器，因此在不同场景下同样的邻域操作也会产生不同的邻域解。以一个3机器7工件规模的问题为例，如图2所示，对同一个解均采用插入操作，在不同的场景下产生了不同的邻域解。其中a)为在场景下进行插入操作的过程，b)为在场景下进行插入操作的过程：在下，关键机器为，对应的加工时间最短的工件为，插入操作为将移动到总加工时间最短的。而在下的关键机器为，则选择上的插入到总加工时间最短的上加工。

因此除了根据一致并行机的问题特点设计邻域操作外，在求解PTM模型时邻域的产生方式还需要充分考虑不同场景对邻域操作造成的影响。 Wang等[16]在求解离散场景下的作业车间调度问题时，在单一场景下采用了基于关键路径的邻域构造方式。为了解决不同场景下关键路径不同导致邻域解不同的影响，Wang等[16]提出了合并场景邻域的概念，在合并场景下进行邻域解的构造：

对于一个解，其在场景下产生的邻域用表示，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：



本文研究的PTM模型以坏场景集为基础。参考文[16]的做法，本文将对单场场景下的一致并行机邻域产生方式进行拓展，围绕建立合并坏场景邻域。

本文用分别表示在场景下采取插入和交换操作后产生的邻域解，则。在其坏场景集下的合并坏场景邻域用表示：



同时在中，不同坏场景下生成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所示：



* 1. 加入合并场景邻域搜索的布谷鸟算法UNCSA

为了消除不同场景对邻域解产生的影响，在上一小节中本文设计了一种基于坏场景的邻域构造方式，这一节中将对加入了该合并坏场景邻域搜索的布谷鸟算法UNCSA的各关键步骤作进一步的详细介绍。整个UNCSA算法的流程图如图4所示。

* + 1. 编码解码和初始种群

在一致并行机调度问题中，主要有基于机器的编码[25](encode based on machine)和基于工件的编码[31](encode based on job)两种编码方式。考虑到本文设计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都不同，因此采用了基于工件的编码方式。对一个涉及台机器，个工件问题的一致并行机问题采用基于工件的编码，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。

以图 2 为例，按照基于工件的编码方式，当前解对应的编码为，在下通过插入方式产生的邻域解为，在下获得的邻域解为。

在传统的布谷鸟算法中，往往通过随机生成的方式产生初始种群[36]，而良好的初始解可以提升求解的效率。参考Laha[32]在求解确定性一致并行机问题时采用LPT规则[40]生成解作为初始解的方式，在本文中，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，后个解则通过对前面已有的初始解进行扰动产生。

* + 1. 连续化离散化

布谷鸟算法最初的设计是为了求解连续优化问题，而一致并行机调度问题的解是具体的工件排序，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，实现了离散调度解和连续位置解之间的相互转换。

对于每一个离散调度解，其对应的连续位置解可以通过下式计算得到：



而对连续解进行离散化得到离散解，常用的方法有最大位置规则（largest position value, LPV）[41],最小位置值规则（smallest position value，SPV）[31]等。参考文[31]，本文中采用SPV规则进行离散化操作，得到离散的加工序列。

* + 1. UNCSA中的局部搜索

在前文中我们介绍了基于坏场景集的合并场景邻域构造方式。UNCSA算法中构造合并坏场景邻域并进行邻域搜索的详细步骤如图5所示：

对于每一代种群中的每个坏解，首先求出其坏场景集和在所有坏场景下的邻域解集合，然后从中选择出性能最好的一个解，如果的性能优于当前解，则用代替。否则以一定的概率接受。

PTM模型的求解涉及多个坏场景，同样的邻域解在不同的坏场景下会有不同的表现。因此相比于普通的邻域搜索，UNCSA中的合并坏场景邻域搜索更加深入，能够充分考虑不同坏场景下的邻域解对PTM模型目标函数的适应性，找寻到更优的解。通过结合布谷鸟算法强大的并行搜索能力和有针对性的合并坏场景邻域构造方式，UNCSA可以更高效地获得更优的解。

* + 1. UNCSA的终止条件

UNCSA的终止条件和其他布谷鸟算法类似[31][32]，都以最大迭代次数作为终止条件。在下一节的实验仿真中我们将会确定的具体数值。

本节介绍了本文针对PTM模型特意设计的合并坏场景集邻域构造方式以及结合了该邻域和布谷鸟算法的UNCSA。在下一节中将进行大量算例的测试，来验证本文所提出的UNCSA算法的有效性和求解效率，

1. 仿真与分析

本节中将对求解PTM模型的UNCSA进行仿真测试。首先将测试不同迭代次数下UNCSA的求解结果，以此确定合适的终止条件。然后通过比较UNCSA算法和其他已有的三种智能优化算法的求解结果，展示UNCSA在求解PTM模型时的优势。

除外，本文中的UNCSA算法参数设置如下：种群大小，步长因子，，局部搜索时接受坏解概率。

本文中的所有仿真程序均采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。仿真用计算机的CPU为Intel(R) Xeon(R) Gold 5188 ,运行内存为256G。

参照文[21][32]中的测试算例规模，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，每种类型各10种算例。不同规模算例的工件数和机器数设置如表1所示。同时参考[10]中离散场景的产生方式来对加工时间进行不确定化处理。所有算例中的工件加工时间均从中产生，其中为控制加工时间取值的参数，在中选取。每个参数组合下生成个场景的加工时间作为一个测试算例。同时对于每个参数组合，均产生5个测试算例，一共测试600个测试算例。在实际测试过程中发现，同样参数组合下的5个算例仿真结果相近。因此本文接下来的仿真都将只取具体参数组合的一个测试算例结果作为展示。

* 1. UNCSA终止准则测试

在本小节中将对UNCSA的终止准则进行测试，从不同规模类型的算例中各选取了两个规模，在固定时间参数时用UNCSA在不同的最大迭代次数下对PTM进行求解。测试的结果如表2所示，其中的为不同算例组合参数组合对应的阈值，为求解PTM模型得到的最终性能值，为整个求解过程消耗的CPU时间。

从表2中可以看出，对于UNCSA，增加最大迭代次数会提升最终结果的求解质量。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。而在迭代到一定代数后解的改善速度开始减缓，之后对UNCSA继续进行迭代并不能有效地提升求解结果，只会无意义地增加时间消耗。为了更清楚地观察这种趋势，本文在不同规模类型中分别挑选了三个规模的算例，绘制了与迭代次数的收敛图，如图6~图8所示。





结合收敛图和表2可以发现，采用UNCSA算法对PTM模型进行求解时，大约会在250次迭代后进入收敛，此时再牺牲CPU时间并不能明显改善解的质量。综合考虑求解质量和CPU时间的消耗，本文中的UNCSA算法的终止准则设置为最大迭代次数。

* 1. UNCSA与其他算法的比较

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[32]设计的加入了普通局部搜索的布谷鸟算法(ICSA)，Lee等[30]设计的加入了局部搜索的遗传算法(GA+LS)，还有Xu[6]等采用的模拟退火算法(SA)。

表3展示了在时不同算法对不同规模算例的求解情况。对比了求解所用的CPU时间和最终求得的值。从表中可以看出在不同规模的算例下，作为串行搜索算法的SA虽然消耗相对较少的CPU，但很难对解空间进行有效的探索，在所有测试算法中求解结果最差。而同样作为并行搜索算法，在都加入了普通邻域进行局部搜索的情况下，ICSA所消耗的CPU时间和最终的结果均优于GA+LS。这表明了布谷鸟算法的求解优势。而采用了基于合并坏场景邻域构造方式的UNCSA，无论是在求解时间还是求解结果上都领先于其他算法。实验结果表明本文设计的UNCSA和合并坏场景邻域构造方式，对一致并行机的PTM模型更具针对性，求解效率更高。

图9~图11是用四种算法求解三个规模算例时的求解过程收敛图。从图中可以进一步看出UNCSA算法的优势。相比普通的邻域构造方式，基于坏场景集的邻域构造方式在每次迭代过程中可以搜寻更多邻域解。随着算法的进行，当普通邻域搜索对解的改善速度越来越缓慢，相应的算法早早进入收敛时，UNCSA依旧可以对解进行一定程度的更新。





1. 总结和展望

本文求解了一致并行机的阈值坏场景集惩罚模型PTM，其中不确定的加工时间利用离散场景表示。给定一个阈值，筛选出场景集中的所有坏场景，通过优化该阈值下所有坏场景性能造成的惩罚值来提升系统的抗风险能力。

为了更好地求解该模型，本文在传统的布谷鸟算法基础上加入了邻域搜索的部分，提出了一种改进的布谷鸟算法UNCSA。除了根据一致并行机调度问题的特点设计了单一场景下新的邻域解产生操作外，还结合PTM模型的问题背景，采用了基于合并坏场景的邻域构造方式，将当前解在所有坏场景下产生的邻域解集合起来统一进行筛选更新。本文将UNCSA算法和其余的三种智能优化算法进行了比较，仿真结果表明UNCSA算法兼顾了布谷鸟算法和合并场景邻域的优点，在求解一致并行机的PTM模型时更具优势。

未来的研究工作可能会朝着以下的方向进行：

1. 在本文的求解过程中，PTM模型的阈值是预先确定的，并没有给出具体的求解方法。之后会先设计算法确定阈值，然后再对PTM模型进行求解。
2. 基于阈值的坏场景集模型最初在作业车间调度问题上应用，本文首次将其运用到一致并行机问题中，之后会尝试用于其他的调度问题。
3. 本文将基于坏场景集的邻域构造方式与布谷鸟算法进行了结合，未来可以尝试与更多的智能优化算法进行结合。

参考文献：

1. 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.
2. 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.
3. 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.
4. Al-Khamis T, Rym M’Hallah. A two-stage stochastic programming model for the parallel machine scheduling problem with machine capacity[J]. Computers & Operations Research, 2011, 38(12):1747-1759.
5. Dubois D, Prade H . Qualitative possibility theory and its applications to constraint satisfaction and decision under uncertainty[J]. International Journal of Intelligent Systems, 1999, 14(1):45-61.
6. 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.
7. Naji W. Robust and stable optimization for parallel machine scheduling problems[D]. Grenoble Alpes, 2018.
8. Kala R, Lamboray C, Vanderpooten D. Lexicographic α-robustness: An alternative to min–max criteria[J]. European Journal of Operational Research, 2012, 220(3): 722-728.
9. Yang J , Yu G . On the Robust Single Machine Scheduling Problem[J]. Journal of Combinatorial Optimization, 2002, 6(1):17-33.
10. 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.
11. Li Z , Ierapetritou M G . Robust Optimization for Process Scheduling Under Uncertainty[J]. Industrial & Engineering Chemistry Research, 2008, 47(12):4148-4157.
12. 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.
13. Drwal M , Jerzy Józefczyk. Robust min–max regret scheduling to minimize the weighted number of late jobs with interval processing times[J]. Annals of Operations Research, 2020, 284(2).
14. 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.
15. Wang B, Xia X, Meng H, et al. Bad-scenario-set robust optimization framework with two objectives for uncertain scheduling systems[J]. IEEE/CAA Journal of Automatica Sinica, 2017, 4(1): 143-153
16. 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.
17. Drwal M, Roman, Complexity of interval minmax regret scheduling on parallel identical machines with total completion time criterion[J]. Operations Research Letters, 2016.
18. Garey M R, Johnson D S. Computers and intractability[M]. San Francisco: freeman, 1979.
19. Lenstra J K , Kan A H G R . Computational Complexity of Discrete Optimization Problems[J]. Annals of Discrete Mathematics, 1979, 4(08):121-140.
20. Dell’Amico, Mauro, Martello S . Optimal Scheduling of Tasks on Identical Parallel Processors[J]. Orsa Journal on Computing, 1995, 7(2):191-200.
21. Mokotoff E . An exact algorithm for the identical parallel machine scheduling problem[J]. European Journal of Operational Research, 2004, 152(3):758-769.
22. Kashan A H , Karimi B . A discrete particle swarm optimization algorithm for scheduling parallel machines[J]. Computers & Industrial Engineering, 2009, 56(1):216-223.
23. Kim J G, Song S, Jeong B J. Minimising total tardiness for the identical parallel machine scheduling problem with splitting jobs and sequence-dependent setup times[J]. International Journal of Production Research, 2020, 58(6): 1628-1643
24. Behnamian, J. Particle swarm optimization-based algorithm for fuzzy parallel machine scheduling[J]. International Journal of Advanced Manufacturing Technology, 2014, 75(5-8):883-895.
25. 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.
26. Venter G , Jaroslaw S S . Particle Swarm Optimization[J]. AIAA Journal, 2003, 41(8):129-132.
27. Karaboga D . An idea based on honey bee swarm for numerical optimization[J]. 2005.
28. Dorigo M , Maniezzo V . Ant system: optimization by a colony of cooperating agents[J]. IEEE Trans. on SMC-Part B, 1996, 26(1):29.
29. 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.
30. 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.
31. 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.
32. 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.
33. Dasgupta P , Das S . A Discrete Inter-Species Cuckoo Search for flowshop scheduling problems[J]. 2015.
34. Li X, Yin M. A hybrid cuckoo search via Lévy flights for the permutation flow shop scheduling problem[J].International Journal of Production Research, 2013, 51(16): 4732-4754.
35. Yang X S , Deb S . Cuckoo Search: Recent Advances and Applications[J]. Neural Computing & Applications, 2014, 24(1):169-174.
36. Yang X S , Deb S . Engineering Optimisation by Cuckoo Search[J]. International Journal of Mathematical Modelling & Numerical Optimisation, 2010, 1(4):330-343.
37. Gandomi A H , Yang X S , Alavi A H . Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems[J]. Engineering with Computers, 2013, 29(1):17–35.
38. Civicioglu P , Besdok E . A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms[M]. Kluwer Academic Publishers, 2013, 39(3):315–346.
39. Yang X S, Deb S. Multiobjective cuckoo search for design optimization[J]. Computers & Operations Research, 2013, 40(6): 1616-1624.
40. Graham R L. Bounds on multiprocessing timing anomalies[J]. SIAM journal on Applied Mathematics, 1969, 17(2): 416-429.
41. 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.