# 基于阈值坏场景集的一致并行机两阶段鲁棒调度

**摘要：**本文讨论了利用离散场景表示不确定加工时间的一致并行机鲁棒调度问题。对于每个调度解，根据阈值选择出坏场景集，并计算该解在坏场景集下的一个罚函数值。整个鲁棒优化目标为一个双目标优化模型，将阈值和坏场景集下的罚函数值作为优化对象，在保留了最终解抗风险能力的同时兼备了对整体优良性能的追求。为了求解该模型，本文提出了一个两阶段算法。在第一阶段将求解一致并行机场景均值的问题转换成单一均值场景下的确定性一致并行机问题，在此基础上求解阈值。第二阶段针对坏场景集的特点采用了基于合并场景的邻域构造方式，并将其与布谷鸟算法（CSA）结合进行求解。仿真结果表明本文设计的两阶段算法对该双目标模型求解的有效性，最终求得的结果在优良性能和鲁棒性上能为不同偏向的决策者提供更多的选择。

**关键词**：鲁棒调度模型，一致并行机调度，双目标优化，两阶段算法，布谷鸟算法，合并场景领域

## 绪论

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，广泛存在于日常的生产加工中。通过确定工件在机器上加工的序列来使某项性能指标达到最优。传统的对于一致并行机问题的研究通常在确定性环境下进行，加工时间等生产参数都是已知且固定不变的。然而由于加工能力，现场环境等各种因素的影响，实际的生产环境充满了不确定性。寻找合适的方法处理不确定参数，建立合适的求解模型来保证不确定环境下的求解质量成为了研究的热门。

场景方法是一种有效的处理不确定参数的方法。通过某个区间的连续场景[1]或者有限个离散场景的集合[2]来描述不确定性。各场景之间分布相对独立，不需要依赖具体的概率分布模型或者模糊规则，目前在调度问题中有着广泛的应用。[3]-[5]

Li等[6]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以所有场景的期望性能作为优化目标[7]，考虑整个不确定环境下的平均性能。基于场景的鲁棒优化模型则是以增加系统整体的鲁棒性为目标。其中比较常见的为最坏场景模型[8]，该模型通过提升所有场景下表现出的最坏性能，来提升整个系统在不确定环境下的抗风险能力。然而这两种模型都存在不同的局限性。随机优化场景在考虑整体的期望性能时往往会忽视一些极端场景，使得最终的决策缺乏抗风险能力。最坏场景模型仅关注单一的性能最坏的场景，缺少了对其他场景性能的积极追求和考量，最后得到的结果太过保守。为了兼顾整体期望性能和抗风险能力， Wang[9]提出了基于坏场景集的鲁棒调度模型。以一个基准性能为阈值，筛选出表现性能较差的坏场景，并以这些坏场景为基础设定不同的鲁棒优化目标。如计算所有的坏场景与基准性能的罚函数，计算所有坏场景下的性能均值等。同时Wang[10]将这些基于坏场景集的鲁棒模型用于作业车间调度，并将计算结果与最坏场景模型和期望场景模型进行了对比，实验结果表明根据基于阈值的坏场景集模型求得的解在保持鲁棒性的同时还具有对优良性能的追求。

求解基于阈值的坏场景鲁棒模型的一个关键步骤在于基准性能的确定。合适的基准性能才能保证所求解的合理性。因此我们希望该取值能尽可能的精确。Wang等[10]在模型提出的最初，没有给出具体的求解方式，后来Wang[11]证明基准性能的合理取值范围应该位于最坏场景性能和最佳期望性能之间，并以此设计了一个两阶段算法进行求解。在第一阶段采用GSA算法求得最佳期望性能的近似值，并以此作为基准性能的起始值进行第二阶段坏场景的筛选。同时还加入了反馈检验的过程来避免第一阶段求得的近似值与真实值相差过大。

本文将对Wang[10]的两阶段算法进行改进，并将其用于基于阈值坏场景集的一致并行机鲁棒调度问题的求解。在第一阶段，本文提出了均值场景的概念，并利用一致并行机调度问题的特性，将基于场景的一致并行机期望模型转化为单一均值场景下的确定性一致并行机问题，然后利用已有的精确算法[12]进行求解，将求得的值作为第二阶段基准性能的起始值，相比近似算法减少了反复检验的过程以及提升了准确度。在第二阶段，将在合理的取值范围内求解不同的基准性能下对应的坏场景和罚函数。为了提升求解的效率，我们针对坏场景集设计了一种基于合并场景的邻域构造方式，并将其与布谷鸟算法（CSA）结合用于第二阶段的双目标优化问题，同时优化基准性能和对应的罚函数值。

本文的剩余部分安排如下：在第二节，我们将对基于阈值的双目标一致并行机鲁棒调度模型进行描述，在第三节中将详细介绍本文设计的两阶段算法，第四节是所提算法的仿真和测试。最后的第五节是总结与展望。

## 2.问题描述

本文研究不确定环境下的一致并行机调度问题，采用离散场景的方式来描述不确定的加工时间。涉及到的相关符号参数如下表示：为机器的数量，表示第台机器，为所有机器的集合。为工件的数量，表示第个工件，为所有工件的集合。 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。表示所有的可行调度解集合，来表示一种具体的可行调度方案，如果选择在上加工，则，否则。表示所有场景的集合，表示场景集中场景的个数。表示一个具体的场景，表示在下各工件的加工时间。表示机器的完工时间，表示在场景下，调度解的最长完工时间makespan。如果仅存在一个场景，当场景固定时，加工时间也确定。此时可以将记作，整个问题也可以看作是对确定性一致并行机问题DIP[12](Deterministic identical parallel machine scheduling)进行求解：

DIP:



### 2.1 基于阈值坏场景集的一致并行机两阶段鲁棒调度模型

本文求解的一致并行机鲁棒模型主要考虑坏场景集下的性能。参考文[10]，坏场景集的定义如下：对于调度解，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:



将在所有坏场景集下的性能相对的波动量作为惩罚值，以此为衡量标准，定义鲁棒度量函数：



以作为优化目标的鲁棒模型被记作PSP：

(PSP):



观察式(2)，式(3)可以发现，在场景集确定的情况下，调度解在不同的值下会产生不同的，从而影响PSP的求解结果。值增加，相应的坏场景的数量减少，得到的值也会降低，反之亦然。因此的取值要尽可能的合理。Wang[11]已经证明，对于场景集，其合理的值取值范围应该在对应的最佳期望性能和最佳最坏场景性能之间，即当时，为有效合理的鲁棒度量函数：





本文求解的鲁棒模型BIPS（bi-objective identical parallel machine scheduling），则是同时将阈值和作为优化目标，方便决策者在实际生产中根据不同的生产能力和抗风险能力，设定不同的值和最符合实际情况的鲁棒解：

(BIPS)：











### 一阶段

为了求解BIPS，需要对场景集的最佳期望性能进行精确的求解，并以此作为阈值的起始值进行下一步的计算。然而根据我们查找的文献，目前在一致并行机调度领域，尚未有求解离散场景下一致并行机最佳期望性能的精确算法。因此本文定义了一种均值场景，将原有的多场景下的一致并行机期望makespan模型转换为单一均值场景下的确定性一致并行机调度问题进行求解。

**定义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。即， 。

用ECP表示如式(5)所示的多场景下的一致并行机期望makespan调度问题（Expected-case scheduling）。则有

**定理1**. 对于场景集下的不确定一致并行机调度问题，其ECP模型可以转换为单一均值场景下的DIP模型的形式。

即可以将式(5)转换为式(1)的形式进行求解(证明见附录):



## 4.二阶段调度

在上一节中介绍了基于阈值的一致并行机鲁棒调度模型BIPS和一致并行机的均值场景模型。在此基础上，本文设计了一个两阶段算法来对BIPS模型进行求解。

### 3.1 两阶段算法求解框架

在第一阶段将对一致并行机的ECP问题进行求解，利用上一节提出的定理将其转化为均值场景下的确定性一致并行机调度问题DIP，然后利用已有的割平面法[12]进行求解。在第二阶段，将第一阶段求得的作为PSP阈值的基准值，然后在合理的取值范围内逐步增加值，求得不同阈值下的。根据Wang[11]的论证，我们无需再求解的具体值，而是可以将作为第二阶段算法的终止条件。即当值过大，对于每个解都不存在坏场景了，此时算法终止。因为PSP问题的目标函数相对复杂，在第二阶段，本文设计了一种基于合并场景邻域的布谷鸟算法UNCSA(union scenario neighborhood cuckoo search)进行求解。布谷鸟算法有着强大的全局搜索能力，自提出以来被广泛地运用在各种优化问题和调度问题中[13]-[15]，而本文在标准布谷鸟算法的基础上加入了一种基于合并场景的邻域构造方式来增加其局部搜索的能力，具体操作会在下文进行详细说明。

整个求解BIPS的两阶段算法流程图如图1所示：



### 3.2 求解第二阶段的布谷鸟算法

一致并行机调度问题属于NP-hard问题[16]，随着问题规模和目标函数复杂性的增加，越来越多的研究人员开始采用智能优化算法进行求解。遗传算法（GA）[17]，模拟退火算法（SA）[18]，禁忌搜索算法（TS）[19]，粒子群算法（PSO）[20]等在其中都得到了应用。布谷鸟算法是一种提出时间较晚的群搜索智能优化算法，凭借着莱维飞行可以在复杂的解空间里进行高效的全局搜索[21]，自提出以来就被应用到调度问题中，Leha[22]，dipa等[23]将其成功地运用优化目标为最小化makespan的一致并行机调度问题中，实验结果表明优于模拟退火算法，粒子群算法等其他的智能优化算法。

标准的布谷鸟算法在一开始产生种群大小为的初始种群，包含个个体，之后第代的种群利用莱维飞行更新每一个个体得到下一代的种群：



其中为步长因子，为固定的常数[21]，通常取。

是服从莱维分布的随机步长：



在更新完新解后选部分性能较坏的解，每个坏解有的概率被随机产生的新解替换。

依靠莱维飞行，布谷鸟算法拥有高效的全局搜索能力，但是在每次迭代过程中仅用随机产生的新解来进行部分解的替换，这使其在局部搜索能力上略有欠缺。本文尝试在抛弃坏解的步骤中加入邻域搜索的机制，用坏解的邻域解来替代随机产生新解，从而提高求解的质量和效率。

#### 3.2.1 基于合并场景的邻域构造

在并行机调度问题中，常见的邻域构造方式有移动[24]，交换[25]等。相比其他类型的并行机问题，一致并行机调度的邻域操作往往围绕加工时间最长的关键机器[1]进行。用表示关键机器，一致并行机围绕进行邻域解构造的操作大致可以分为以下几类：交换(exchange)，随机移动(move)，插入(insert)。其中交换操作又可以分为一对一交换(exchange1)和多对多交换(exchange2)[26] -[29]。此外还有循环使用以上多种邻域构造方式的操作[30]。相比之下，多对多交换和插入方式通常针对加工时间最长的工件，在工件的选择上更具针对性。多对多交换是从上随机选择加工时间最长的和，将其与另一台非关键机器上的和进行交换位置。插入操作则是上选择加工时间最长的工件，再选择一台，将插入到的加工队列中。本文将采用以上两种操作产生邻域解。

以上的一致并行机邻域解构造方式一般都是在确定性环境下进行的。在不确定环境中，同一个解在不同场景下会呈现出不同的关键机器，因此同样的邻域操作在不同场景下也会产生不同的邻域解。以一个3机器7工件的一致并行机调度问题为例，图2展示了对该问题的一个解采用插入方式在不同场景下产生的邻域解。其中(a)为在场景下进行插入操作的过程，(b)为在场景下利用插入操作产生邻域解的过程：

因为第二阶段的目标函数涉及多个坏场景，为了充分考虑不同场景对邻域解产生的影响，参考Wang[10]在求解离散场景下的作业车间调度问题时，提出了一种基于合并场景的邻域构造方式，以此来处理不同场景下每个解的关键路径不同造成的影响。本文对常规的一致并行机邻域产生方式进行拓展，采用合并场景的构造来使邻域解的产生更加合理。

对于一个解，其在场景下产生的邻域用表示，合并场景领域表示在场景集下的合并场景邻域：



本文的目标函数与坏场景集有关，将围绕建立合并场景邻域。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邻域用表示：



整个基于坏场景产生合并场景邻域的伪代码如图3所示：



#### 3.2.2 基于合并场景邻域的布谷鸟算法

上一节中为了消除不同场景对邻域解产生的影响，结合本文的目标函数，我们设计了一种基于坏场景集的合并场景邻域构造方式。这一节中将把该邻域构造方式加入标准布谷鸟算法中，设计了一种基于合并场景邻域的布谷鸟算法 (UNCSA)， 具体步骤如下所示：

在一致并行机调度问题中，主要采用基于机器的编码[32](encode based on machine)和基于工件的编码[23] (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组成一个子集，所有子集按机器序号排列形成一个解。

以图2为例，按照基于工件的编码方式，图2中的当前解的编码为，在下通过插入方式获得的d邻域解为，在下通过插入方式获得的邻域解。

本文将在每个场景下利用LPT规则[31]产生的解作为初始的种群。

C.离散化：

标准的布谷鸟算法最初的设计是为了求解连续的优化问题。而一致并行机问题的解是具体的工件编号，属于离散的优化问题。因此我们在标准布谷鸟算法的基础上增加了离散化的步骤，实现了离散调度解和连续位置解之间的相互转换。参考文[30]，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：



而对于连续位置值到离散调度解，可以采用SPV规则[30]进行。SPV(smallest position value)规则可以将连续变量转换成离散的加工序列。

D. 精英保留策略

因为本文采用合并场景邻域产生大量的邻域解，为了使每一代的精英解得到保留，同时控制种群数量大小，本文加入了精英保留策略。记每一代种群大小为，对于每一代种群，其产生的合并场景邻域解用表示，合并，计算性能后选出个最优的解作为下一代的种群。

E.UNCSA步骤

整个UNCSA的步骤如下。首先利用LPT规则[31]产生个解作为初始种群，再利用式(19)对得到其对应的位置解，之后的每一次迭代过程中先对位置解进行莱维飞行，然后通过SPV规则进行离散化得到的子代种群。通过计算中每个解的适应值，选择部分性能较差的个体，用合并场景邻域产生的邻域解代替。在利用精英保留策略得到下一代的种群。UNCSA的流程图如图4所示。





F.终止准则

UNCSA的终止准则和标准的布谷鸟算法相似[21]，都是以最大迭代次数作为终止条件。

## 4.仿真与分析

为了测试整个算法，本文将通过大量实验仿真来进行验证。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。仿真用计算机的CPU为Intel(R) Xeon(R) Gold 5188 ,运行内存为256G。

参照文[12][18]的测试算例，选取了测试问题的规模如表1所示。同时参考[33]中离散场景的产生方式来对加工时间进行不确定化处理。每个工件的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340个测试算例。在实际测试过程中发现，同样的参数组合仿真多次的结果相近，因此本文接下来的仿真都将只取某个参数组合的一个测试算例结果作为展示。



### 4.1 第一阶段算法仿真测试

在第一阶段，本文采用了文[12]的割平面法来对一致并行机均值场景下的DIP问题进行求解。为了确保算法能对第一阶段问题进行有效的求解，本文对不同参数的算例进行了仿真测试。表2是选取了3种不同规模的算例在不同时间参数下的仿真结果，从测试结果来看，在算例规模相同的情况下，无论是较小的规模(3\*20)，还是较大规模(100\*500)，不同的时间参数主要影响加工时间的取值，从而影响最后的。同时对运算时间和迭代次数也有一定影响，算例规模越大，在增大时其运算时间和迭代次数的绝对变化量越大。但对于同一个规模来说，这种变化量远不如运算规模对运算时间造成的变化明显，因此在接下来的不同规模的仿真结果对比时，本文将只展示的结果。



表3是不同规模的算例在时间参数下的仿真结果，可以看到，最后求得的主要受工件数和机器数的比值n/m的影响较大。更大的n/m意味着平均一台机器上会安排更多的工件进行加工，整体的运行时间肯定会增加。而整体的运算时间和迭代次数随着规模的增加而增加。在大规模的算例下这种变化更加明显。当算例达到一定规模后，如(500\*1000)，此时精确算法的求解时间已经很大，此时的求解需要消耗大量的时间资源和计算机资源，但实验已经表明该精确算法对于相当大规模的算例都可以进行求解。在实际问题中可以进行应用。本阶段的求解结果也将在下一阶段使用。



### 4.2 第二阶段UNCSA仿真测试

本文在第二阶段设计了采用合并场景邻域的布谷鸟算法UNCSA，在本节中将对该算法进行仿真测试。根据3.1节中设计的两阶段求解流程，第二阶段的算例参数与第一阶段相同，第二阶段中不同算例下的阈值的基准值为对应的第一阶段算例的计算结果。UNCSA自身的算法参数设计如下：种群大小步长因子，，坏解抛弃概率。

#### 4.2.1 UNCSA的终止准则测试

本小节中将对UNCSA算法的终止准则进行测试。选取了部分规模的算例测试时其在不同最大迭代次数下的CPU时间和最终的。测试结果如表4所示。

从表4中可以看出，在不同的规模下，最大迭代次数的增加都会导致求解时间的增加，但同时也能提升最终的结果的求解质量。但最大迭代次数增加到一定程度后解的改善就变得相对换暖。为了更清楚地观察这种趋势，本文选取了两种规模的算例(m=3,n=20)和(m=5,n=100)，根据它们的迭代次数和每代求得的最佳值绘制了迭代过程的收敛图。如图5图6所示，结合表4可以发现整个求解过程在250代左右接近收敛，此时再增加迭代次数，会进一步增加CPU时间但是获得的结果增益不大。因此本文将UNCSA的终止条件设为。



#### 4.2.2 UNCSA算法和其他算法的比较

本小节中将采用UNCSA算法来求解本文提出的BIPS模型第二阶段，并将求解结果与未采用合并场景邻域的标准CSA[21]，PSO[20]，GA[17]，SA[18]进行比较。

我们选取了两种规模差异较大的算例(m=3,n=20)和(m=100,n=200)进行结果展示，仿真的结果如表5所示。从表中可以看出，在解空间较为复杂的情况下，虽然SA所需时间最少，但其求解结果远不如其他几种并行搜索算法(CSA,PSO,GA)。而CSA的求解效率在其中又高于其他两种算法，在求解质量和求解时间上均占优势。相比CSA，本文设计的UNCSA由于采用了合并场景的邻域进行局部更新，相比标准的CSA增加了对多个坏场景的邻域构造过程，所以会多消耗一点求解时间。在未明显增加求解时间的情况下，UNCSA的求解结果更具优势，解的质量得到了进一步的提升。同时还可以发现，在阈值逐渐提升的过程中，由于坏场景的数量在逐渐变少，相对的求解时间也会减小。而UNCSA除了目标函数外，邻域构造方式也与坏场景密切相关，此时UNCSA的CPU时间减少更加明显。

图7和图8是根据不同算法在不同阈值下求得的目标函数值绘制的前沿图，从图中可以更清楚地看出UNCSA在求解BIPS模型时相比其他算法解的质量更占优。

## 5.总结和展望



本文提出了一种基于阈值坏场景集的一致并行机鲁棒调度模型BIPS，将阈值和与之下求得的坏场景集罚函数值作为优化指标。方便决策者根据现实的情况决定生产标准和直顶生产方案。

为了求解该模型，本文设计了一个两阶段的求解算法。在第一阶段将求解一致并行机场景均值的问题转换成单一均值场景下的确定性一致并行机问题，再利用已有的割平面进行求解，并将所得的结果作为第二阶段。在第二阶段设计了一种基于合并场景邻域的布谷鸟算法UNCSA，在标准的布谷鸟算法基础上加入了基于坏场景的合并场景邻域构造方式，增加了其局部搜索的能力。

为了测试UNCSA的求解效果，本文仿真了大量不同规模的算例，并于已有的智能优化算法进行对比。仿真结果显示基于坏场景集的合并场景邻域构造方式能够大大提升求解效果，使得UNCSA在求解本文的BIPS模型上更具优势。

在之后的研究中，我们可能会考虑将这种模型应用到其他的调度问题中，或是将这种邻域构造方式与其他的算法结合进行问题的求解。



参考文献：

1. 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.
2. Naji W. Robust and stable optimization for parallel machine scheduling problems[D]. Grenoble Alpes, 2018.
3. 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.
4. Yang J , Yu G . On the Robust Single Machine Scheduling Problem[J]. Journal of Combinatorial Optimization, 2002, 6(1):17-33.
5. 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.
6. Li Z , Ierapetritou M G . Robust Optimization for Process Scheduling Under Uncertainty[J]. Industrial & Engineering Chemistry Research, 2008, 47(12):4148-4157.
7. 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.
8. Daniels R L , Kouvelis P . Robust Scheduling to Hedge Against Processing Time Uncertainty in Single-Stage Production[J]. Management Science, 1995, 41(2):363-376.
9. WANG Bing, YANG XiaoFei, LI QiaoYun,. Bad-scenario Set Based Risk-resisting Robust Scheduling Model基于坏场景集的抗风险鲁棒调度模型[J]. 自动化学报, 2012, 38(2):270-278.
10. 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.
11. 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.
12. Mokotoff E . An exact algorithm for the identical parallel machine scheduling problem[J]. European Journal of Operational Research, 2004, 152(3):758-769.
13. 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.
14. Ouaarab A, Ahiod B, Yang X S, et al. Discrete cuckoo search algorithm for job shop scheduling problem[C].2014 IEEE international symposium on intelligent control (ISIC). IEEE, 2014: 1872-1876.
15. Hanoun S, Nahavandi S, Creighton D, et al. Solving a multiobjective job shop scheduling problem using Pareto Archived Cuckoo Search[C].Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012). IEEE, 2012: 1-8.
16. Garey M R, Johnson D S. Computers and intractability[M]. San Francisco: freeman, 1979.
17. 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.
18. 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.
19. 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.
20. Kashan A H , Karimi B . A discrete particle swarm optimization algorithm for scheduling parallel machines[J]. Computers & Industrial Engineering, 2009, 56(1):216-223.
21. Yang X S, Deb S. Cuckoo search via Lévy flights[C].2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). IEEE, 2009: 210-214.
22. Laha D , Behera D K . An Improved Cuckoo Search Algorithm for Parallel Machine Scheduling.[C]. International Conference on Swarm. Springer International Publishing, 2014.
23. 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.
24. L P Cota, M N Haddad, M J F Souza and V N Coelho. A heuristic algorithm for solving the unrelated parallel machine scheduling problem. 2014 IEEE Congress on Evolutionary Computation (CEC). IEEE, Beijing, 1855–1862.
25. De Paula, M.R., Ravetti,M.G., Mateus, G.R., Pardalos, P.M., 2007. Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search. IMA Journal Management Mathematics 18, 2, 101–115.
26. Ibrahim A , Khaled B , 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, 2018(pt.4):3586731.1-3586731.8.
27. Haleel A J . IDENTICAL PARALLEL MACHINES SCHEDULING USING GENETIC ALGORITHM[J]. THE IRAQI JOURNAL FOR MECHANICAL AND MATERIALS ENGINEERING, 2018, 18(2):321-330.
28. Jing C , Jun-Qing L . Efficient variable neighborhood search for identical parallel machines scheduling[C]. Control Conference. IEEE, 2012.
29. Sevkli M , Uysal H . A modified Variable Neighborhood Search for minimizing the makespan on identical parallel machines[C].International Conference on Computers & Industrial Engineering. IEEE, 2009.
30. 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.
31. Graham R L. Bounds on multiprocessing timing anomalies[J]. SIAM journal on Applied Mathematics, 1969, 17(2): 416-429.
32. 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):p.399-403.
33. 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.

## 附录

定理1证明：

**证明.**

DIP模型的数学规划的形式表示如下：

(DIP):









用表示第个场景下的makespan。则式(4)所示的场景集下的一致并行机ECP问题可用数学规划形式表达如下：

(ECP)：









…





将式(16)-式(19)相加并进一步简化，表示成：







根据均值场景的定义，均值场景下的加工时间，将其带入式(22)，则有：

(ECP)：









式(24)-(27)形式与式(11)-(14)DIP的表示形式相同。

□

通过求解式(24)-式(27)，可以得到均值场景下的最优解，然后将代入式(9) 求得，即为要求的多场景下一致并行机的ECP的最优均值性能。