**带有柔性项目结构的随机资源均衡问题**

# SRLP-PS

## 问题描述

本文研究的SRLP-PS可以描述如下:

一个项目可以用一个节点式有向网络表示，其中为节点集合，表示项目中的活动，。活动0和是虚活动，分别代表项目的开始和结束。为有向弧的集合，表示活动之间具有完成-开始的零延迟优先关系，即每个活动必须在其所有紧前活动完成之后，才能开始执行。当 时, 称活动是的紧前活动，是的紧后活动。每个非虚活动的工期是一个随机变量，并且遵循一个已知的概率分布。本文用一组有限的情景集合表示不确定的活动工期，项目中所有活动的随机工期的一次采样（实现）构成一个情景，用来表示，情景出现的概率为。一个情景就是一个实际情况的可能。虚活动的工期为0。因为活动的工期是随机变量，所以活动的开始时间也是随机变量，虚终止活动的开始时间对应项目的完成时间。项目需要在给定的项目截止日期之前完成。在项目中，存在种可更新资源，每个时段资源的供应量用表示。非虚活动执行时，在单位时段对第种可更新资源的需求量为。虚活动在执行时不消耗任何资源。资源在时段的使用量用表示，即在时段所有正在执行的活动对资源的总需求量，，其中表示在时段正在执行的活动集合，即。因为受和的影响，因此也是一个随机变量，且每一种情景下，对应不同的资源使用量。每个活动一次性执行，不允许中断。

为了对柔性项目结构进行建模，将活动集合进一步分为三个互斥的子集 (Kellenbrink & Helber, 2015)：必须执行活动集、可选活动集和依赖活动集，,。必须执行活动集中的活动是总是需要执行的活动。除了必须执行活动，其他活动的执行与否是不固定的。因此，一个柔性项目中只需要执行所有活动的子集，且存在多种项目结构，不同的项目结构由不同的活动集合和相应的优先关系构成。

可选活动集可能包含多个互斥的子集，即，。在项目执行时，并非每个可选活动子集都要执行。每个子集都对应了一个选择。每个选择可以被一个活动触发，如果选择被触发，那么一个可选活动将执行，中的其他活动都不执行。活动可以是必须执行活动、可选活动或依赖活动。需要注意的是， 在Kellenbrink & Helber研究的柔性项目结构中只能是必须执行活动或可选活动。选择是拓扑排序的，被选中的活动执行可能会触发下一个选择(。

依赖活动集中活动的执行与否同可选活动有关。一个可选活动的执行可能会导致多个依赖活动执行，即如果可选活动执行，那么所有依赖的活动（这些活动的集合记为）必须都执行。此外，可选活动的依赖活动只能是它后面的活动，即可选活动的编号小于其依赖活动集中活动的编号。需要注意的是，依赖活动和可选活动之间可能不存在优先关系。

考虑到不确定的活动工期和柔性项目结构，SRLP-PS的目标是在尽可能满足优先关系、资源可用量和项目截止日期的约束下，获得一个调度策略，该调度策略指明需要执行的活动和活动的调度顺序。根据获得的调度策略，可以为执行活动安排开始时间，以最小化资源使用量的期望波动。

## 随机规划模型

首先，引入0-1决策变量表示活动在时刻是否开始：如果活动在时刻开始，则，否则。同时，这一变量也隐含实现了是否执行某些活动的功能：如果活动不执行，则在任何时刻，均为0。接下来，建立SRLP-PS的随机规划模型M0：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (M0) | Minimize  = | | | (1) |
|  | Subject to: | | |  |
|  |  | | | (2) |
|  |  |  | | (3) |
|  |  |  | | (4) |
|  |  | ; ; | | (5) |
|  |  | |  | (6) |
|  |  | ; | | (7) |
|  |  | ; | | (8) |
|  | = |  | | (9) |

目标函数(1)第一项衡量了在截止日期内任意两个相邻时间段内资源使用量的变动情况，是资源的权重，表示资源的单位惩罚成本。第二项表示了项目的完成时间超过截止日期的惩罚成本，为单位惩罚成本。为所有活动平均工期的总和，使用这个参数是考虑到活动的开始时间可能会超过截止日期，如果的取值范围为[1,]，则项目的完成时间永远不会超过。我想着换一种建模方式，不用，但好像不能，因为要计算。式(2)确保了虚开始活动0的开始时间为0。式(3)保证了所有必须执行活动都执行。式(4)确保了如果选择被执行的活动触发，那么在选择的可选活动集中必须有一个活动执行。式(5)表示可选活动执行则其相应的依赖活动也都要执行。在不确定的环境中，包含随机变量的约束可能不完全满足，采用机会约束更符合实际情况。式(6)和(7)都是机会约束，其中表示事件发生的概率，和为置信水平，它们反映了决策者对风险的偏好。具体而言，式(6)表示活动之间的优先关系，它要求活动的完成时间早于其紧后活动的开始时间的概率大于等于，其中是一个任意大的正常数，只有活动和都执行时，该约束条件才成立，即活动和之间的优先关系才有效。式(7)确保在整个项目执行中资源的使用量满足其可用量的概率不小于。式(8)用于计算整个项目的资源使用量。式(9)给出了决策变量的取值范围。

## 模型线性化

本节将随机规划模型M0线性化为等价的整数规划模型，以便用商业求解器进行求解。由于M0的目标函数是非线性的，因此本文进一步将目标函数线性化。引入辅助变量表示在情景中，两个相邻时段的资源使用量与之间的偏差，可表示为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | (10) |
|  |  |  |  | (11) |

接下来，引入两个非负变量和，则可线性化为：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | (12) |

则可将目标函数(1)线性化为：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | (13) |

接下来，用确定性情景约束代替随机机会约束。首先，定义一个很大的正数，二元变量和分别表示在情景中，执行活动的开始时间是否晚于其紧前活动的开始时间和项目的资源使用量是否超过资源的可用量。通过定义以上变量，可将机会约束(6)-(7)改写为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  | (14) |
|  |  |  | | (15) |
|  |  | ; ; | | (16) |
|  |  |  | | (17) |

式(14)保证了执行活动的开始时间满足优先关系（），或不满足（）。式(15)确保了在个情景下满足优先关系的约束符合期望概率。式(16)表示资源的使用量不超过给定量（），或超过（）。式(17)确保在所有情景下满足资源可用量的概率不小于。

式(8)可以改写为：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | ;; | (18) |

根据上述计算公式，可建立整数规划模型M1：

|  |  |  |  |
| --- | --- | --- | --- |
| (M1) | Minimize = (13) |  |  |
|  | Subject to: |  |  |
|  | (2)-(5), (9),(14)-(18) |  |  |
|  | , = {0, 1} |  | (19) |

# 计算实验

## 收敛性分析

J5，截止日期为，20个实例，16个实例在所有情景下求出最优解。表3展示了不同情景数量下目标函数的最大值（Max）、最小值（Min）、极差（Gap=Max-Min）、均值（Avg）、标准差（Std）、变异系数（CV）和求解每个实例所需的平均计算时间（CPU），Pr()表示满足优先关系的概率，Pr()表示满足给定资源使用量的概率

表 1 不同情景数下的统计结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 情景数量 | Min | Max | Gap | Avg | Std | CV | CPU(s) | Pr() | Pr() |
| 10 | 14.40 | 69.00 | 54.60 | 30.69 | 16.48 | 0.54 | 1.66 | 90.00% | 90.00% |
| 30 | 14.87 | 84.47 | 69.60 | 34.66 | 20.60 | 0.59 | 8.39 | 94.17% | 94.58% |
| 50 | 14.92 | 82.44 | 67.52 | 35.30 | 20.84 | 0.59 | 19.65 | 92.63% | 94.88% |
| 80 | 14.33 | 79.95 | 65.63 | 35.23 | 20.69 | 0.59 | 38.18 | 93.91% | 94.84% |
| **100** | **14.76** | **80.13** | **65.37** | **35.48** | **20.35** | **0.57** | 52.99 | 93.13% | 97.13% |
| 150 | 14.84 | 81.09 | 66.25 | 35.65 | 20.80 | 0.58 | 143.65 | 96.50% | 97.17% |
| 200 | 14.82 | 80.97 | 66.15 | 35.70 | 20.87 | 0.58 | 239.68 | 92.94% | 95.41% |

情景数越多，越难满足截止日期，惩罚越大，因此目标函数值越大（特别是在惩罚系数很大的情况）

## 实验结果

CPLEX获得调度策略与算法得到的调度策略进行对比

评价指标：

|  |  |
| --- | --- |
|  | (20) |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | ARD | |  | CPU(s) | | |
|  |  | DE | GA |  | DE | GA | CPLEX |
| 7 |  | -7.05% | -7.05% |  | 10.38 | 7.53 | 162.28 |
|  | -7.12% | -7.12% |  | 11.37 | 7.92 | 196.38 |

DE和GA获得的调度策略比CPLEX获得的调度策略好，在小规模数据集上，DE和GA的求解效果相同。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | ARD | |  | CPU(s) | |
|  |  | DE | GA |  | DE | GA |
| 30 |  | 2.29% | -1.39% |  | 23.01 | 20.10 |
|  | 2.13% | -1.03% |  | 22.42 | 21.06 |

问题：

1. 算法中生成初始的AL的生成方式：满足优先关系或者随机生成
   1. 算法原来的局部搜索阶段，也是考虑了优先关系的，如果两个活动有优先关系就不能交换
   2. 如果不考虑优先关系随机生成，则局部搜索阶段，可以任意交换活动的位置
2. 活动的开始时间不依赖于情景
   1. 模型的目标函数为，得到调度策略再仿真，此时的目标函数为。此时，已经不能保证CPLEX得到的调度策略仿真之后仍然是最优解，因为目标函数已经不一样了（模型的目标函数和仿真的目标函数不一样）。相应的实验结果是说明了这一点。
3. 收敛性分析
   1. 直接将CPLEX输出的结果做收敛性分析
   2. 将获得的进度计划转化为调度策略，重新计算每个情景下的目标函数值，再做收敛性分析