

## 2023 “华为云杯” 人工智能应用大赛赛题

### 一、赛事背景

流水线的发明对于制造行业来说是一次技术变革，极大提高了工作效率。当今许多制造厂家都是采用流水线进行生产，例如汽车制造、芯片制造、服装加工等。流水线均衡则是依照流水线作业的工程顺序，将作业分割或者结合，使各个工位的负荷均匀，以提高生产效率。传统的流水线均衡是由流水线编排人员人工来实现的，这严重依赖于人员的专业素养与编排经验，在调配各生产要素时难以做到全局最优。流水线均衡的关键不仅要考虑工序、工位和设备的安排，还需要均衡各个工位的工作负荷，提高流水线的整体均衡程度和生产效率。

本次赛题设置，面向实际工业生产环境，希望参赛者通过AI模仿编排人员的工位分配、工序拆分和合并等能力，解决瓶颈工序，让生产更加顺滑高效。

### 二、比赛任务

本赛题是针对真实工业场景中流水线生产。流水线是常见的一个生产系统，由 $m$ 个工位组成，工件或任务沿着生产线从一个工位移动到另一个工位，每个工位完成指定的工序。由于工艺要求，不同工序之间一般存在先后顺序，并且完成每个工序需要一定的设备以及有专业技能的人员。工序分配问题为根据生产线上需要生产工单的工艺路线以及产线上的资源（包含工位、设备、员工等），完成工序的分配/指派的问题，如图1。这是一类典型的装配线均衡问题，即将工序进行合理地分配到产线的各个工位上，在满足各种约束的条件下，将生产任务尽可能平均地分配到各个工位/机器/员工上，解决“工作超负荷”和“工作闲置”情况，进而提升生产效率。参赛选手可以通过对生产过程进行建模与求解，输出流水线生产作业计划。

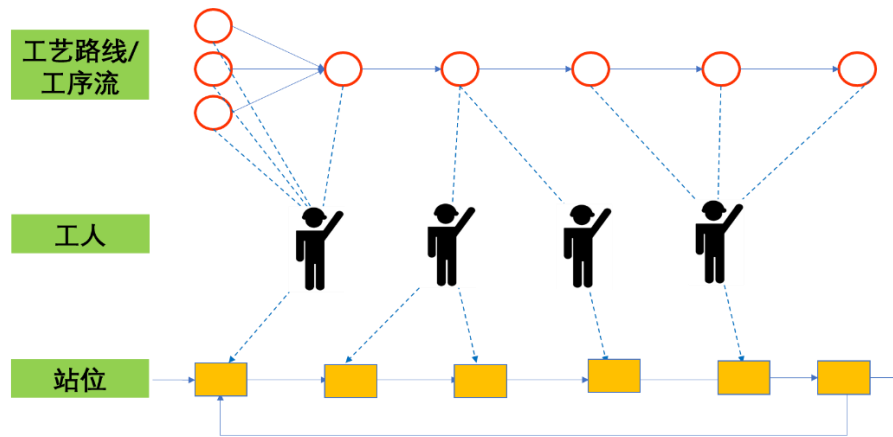


图1 工序分配问题示意图

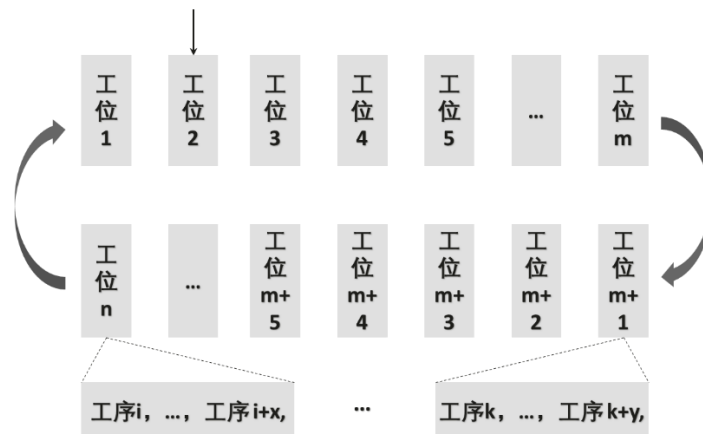


图2 环形产线以及工序分配示意图

比赛考虑加工工件通常含有多道工序（几十到上百种不等），单个工件的单个工序不可分割（即必须一次性完成，不可只做一半），有标准加工时间、所需设备类型等属性。

加工工件的所有工序中，部分工序存在严格的先后关系，部分工序之间的顺序可以调整，不同工序之间的前后顺序可以用树来表示。在表示做工顺序的图中，同一分支（同一部件）上的工序必须按顺序完成，不同分支（不同部件）的工序先后顺序可以由节点工序来表示。工序流的简单示意图如下：

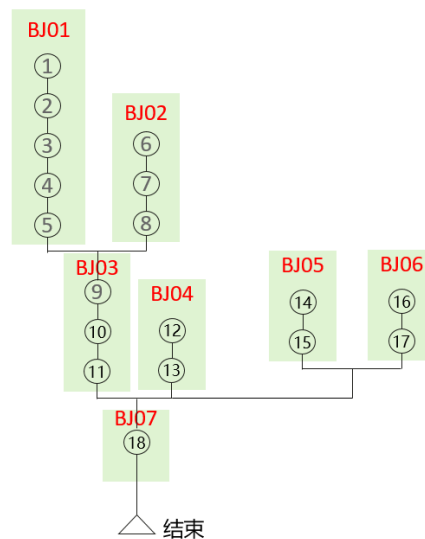


图3 工序流顺序示意图

在图3中，部件BJ01上的工序（工序1-5）必须按顺序完成，即做完工序1才可做工序2；对于不同部件如BJ01和BJ02上的工序，如工序1和工序7，由于其同属于一个父节点（工序9），其做工的先后顺序没有严格要求，可以完成工序1后再做工序7，也可以完成工序7后再做工序1，但这2道工序必须在节点工序9之前完成，BJ01和BJ02的节点工序均为工序9，而部件BJ03、BJ04、BJ05、BJ06的节点工序均为工序18。

只要不违反先后顺序，在为工人进行工序分配时，皆可做自由的工序组合。如工人A可以做1, 2, 6工序，工人B可以做3, 7, 8工序，工人C可以做4, 5, 9, 10工序。其中，由于工人的技能效率不同，不同员工做一道工序的实际完成时间会不同，实际完成时间为工序的标准工时/员工加工此工序的效率。例如，工序1的标准工时是100s，员工A做工序1的效率为200%，则其实际做工序1需要的时间为 $100s/200\% = 50s$ ；员工B做工序1的效率为50%，则其实际做工序1需要的时间为 $100s/50\% = 200s$ 。

比赛考虑的流水线为环形产线，环形产线的特点是允许工件反复进入同一个工位。如图4所示，工位的顺序是1、2、…、6，若工序1-7必须按顺序逐个完成，按照图中右侧的分配结果的话，则在完成工序6后才可再次进入工位1完成工序7（重复入站），即工位1发生了一次重复进站。生产上尽量减少重复进站。

其次，大多工序使用的普通设备允许移动，但部分工序使用的设备是固定在某几个工位上的，不可移动、增加。如图5所示，如果工序2使用的设备为固定设备，只有工位3才有，则在产线实际运行时，在工位1完成工序1后需要跳过工位2，在工位3完成工序2，

由此在工位有限的情况下，员工F只能分配在工位2，在第二圈运行时完成工序6，在第三圈运行时由员工A完成工序7。因此在分配工序到员工及工位时，需要同时考虑工艺流的顺序、工位的顺序、固定设备的位置、员工的技能。

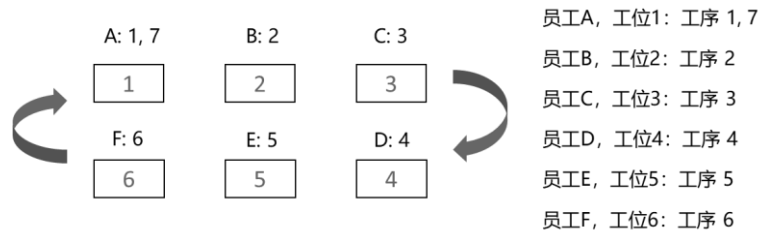


图4 环形产线以及工序分配结果示意图-1

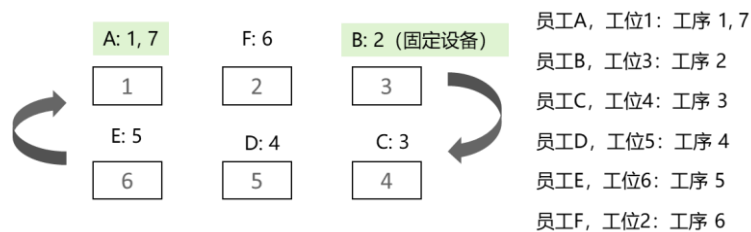


图5 环形产线以及工序分配结果示意图-2

在生产线上进行加工时，一般会对工件进行批量生产加工。如果在分配工序时，某个员工的工序/工序组合需要的加工时间（节拍）相对于其他员工较大，容易产生瓶颈，可以将其生产的部分工件（如50%）拆分给其他人做，以达到节拍均衡。**拆分后的做工人员数一般情况下不超过3人（可配置）。**一个工单中的所有工序，允许进行拆分的工序组合总个数存在上限要求。需要注意的是，拆分的工序数或员工数越多，需要的工位数和设备数越多，这些需要计入成本考虑。

为了实现节拍均衡，会将多个工序分配给同一个人做。但由于工位有可摆放的设备数量限制，需要为此员工分配多个工位，该员工做完部分工序后转移至另一工位完成其余的工序。由于员工从一个工位移动到另一个工位需要走动时间，给员工分配相邻工位能优先减少走动时间，提供工作效率。

工件在环形产线上运转圈数越多，移动距离越长，会增加耗时，因此，应合理安排工位和做工顺序，减少工件在产线上的绕行圈数。

比赛考虑目标和因素汇总如表1。

表1 场景目标和考虑因素

类型	目标与考虑因素
----	---------

主要目标（初赛）		最小化最大节拍（score1）与节拍波动率（score2）
次数目标（复赛）		<p>最小化不在相邻工位进行工作的员工数量（如果员工分了多工位，为减少人工跑动，工位尽量为相邻工位）</p> <p>最小化人员移动次数</p> <p>最小化稀缺设备使用数量（部分设备类型属于稀缺设备）</p>
约束	工艺规则	1、存在强先后关系的工序不能调整工序流顺序。例如：假设工序 A 必须在工序 B 之前完成，不能分工成工序 B 在工序 A 之前进行。
		2、某个员工在单个工位上完成的工序组合。由于批量生产，可以将此员工的部分工件的整个工序组合拆分给别的员工进行加工，不可只拆分工序组合中的部分工序。例如小王做工序 A,B,C，可将 A,B,C 整组工序拆分给小李进行帮扶，但不能只将 B,C 给小李做。一个工单最多允许将 max_split_num（总数）个工序组合进行拆分给多人生产。
		3、由于实行批量生产，单个工序/工序组合原则上全部由 1 个员工生产，但当该工序时间较长有可能造成瓶颈时，可将该工序的批量生产分给最多 max_worker_per_oper（此参数描述允许单工序允许分配的最大员工数量）个员工。例如，max_worker_per_oper=2，要生产 1000 件产品，可将 500 件给小李，500 件给小王，单个工序拆分为多人生产。
	人员规则	1、给定的员工必须参与分工，即每个员工都有分配的工序
		2、员工要完成此工序时必须具备对应技能
		3、考虑员工操作能力限制，一个员工最多安排 max_station_per_worker 个工位
		4、同类工序为相似度较高的单工序组成的集合。单个工序需分配给具备对应工序技能的员工。若所有员工均无此单个工序技能，则才考虑有对应同类型工序能力的员工。
		5、允许提前指定某个工序必须由某个员工完成（该员工一定具备该工序的技能），同时可以指定或不指定工位
	设备规则	1、由于工位空间限制，一个工位不能超过 max_machine_per_station 台设备

		2、部分设备由于体积较大，只能独占工位，不允许再安排其他设备再此工位
		3、相同类型设备，工序对设备规格有要求，即有可能两不同工序所需同一设备类型，但需在不同的设备上完成。例如，工序 1 的"machine_type"为"m-688067-01", "machine_type_2"为 "A",工序 2 的"machine_type"为"m-688067-01","machine_type_2"为"B",此时，工序 1 和工序 2 需要采用 2 台 m-688067-01 设备。
		4、产线中存在固定设备，不能移动、增加、减少
	工位规则	1、一个工位最多分配一个人
		2、在环形产线中，相邻工位包括两侧及对面的工位。对面的工位由于距离近也便于到达也视为相邻工位。
	其他约束	1、工件在环形产线上进行加工，最大转圈次数存在限制，要小于 max_cycle_count 次
		2、节拍均衡：员工安排了不同工序的任务，所有员工的节拍时间（分配工序所花时间）都在平均节拍的上下正负 volatility_rate 以内
3、产线希望每个工位都尽量减少重复入站的次数。所有工位的重复入站的总数量小于 max_revisited_station_count。（固定设备所在工位的重复入站数无限制，其他设备所在站位重复入站数上限不超过 2 次）		
计划输出	工序分配方案，即工序、员工、工位的分配关系	

### 三、数据集

本次比赛数据全部来自于真实工业场景，分为训练集和测试集，源自相同的问题分布。训练集在比赛开始时公开给供参赛者训练他们的模型，测试集将被隐藏以用于评估参赛者效果与泛化性能，并且只会在比赛结束时公开。

训练集共60个生产场景，数据采用 json 格式，本次比赛的要求即为每一个生产场景输出对应的流水线生产作业计划。测试集共有30+个生产场景，数据采用 json 格式。

说明：本次大赛参赛者只能使用组委会提供的数据，不允许使用其它数据，**赛题所使**

用的训练数据集可拉至此页面底端 <数据集> 附件进行下载，并提供部分训练生产场景的对应输出供参赛选手参考。

#### 四、评价方法

每个生产场景的评价指标采用  $score = (1 - score1) * w1 + score2 * w2$

1)  $score1$  = 所有员工的平均节拍/最大节拍

2)  $score2$  = 员工节拍的标准差/平均节拍

最终的评价指标为  $avg(score)$

#### 五、规则

1. 参赛者可以团队合作。
2. 参赛者可以根据机器学习/强化学习模型、人工设计的算法或两者的任意组合提交代码；
3. 参赛者可使用Scip、Highs等开源求解器，禁止使用 CPLEX、Gurobi等商业求解器。
4. 提交的代码必须用Python编写，并且必须在我们提供的评估环境中运行。
5. 在比赛结束时，获胜者需要描述他们使用的方法。

#### 六、提交说明

1. 提交时间：2023年6月26日~8月28日 18:00
2. 提交方法：在大赛平台“提交作品”页面提交所有作品材料，每支团队每日有2次提交机会。
3. 模型提交判分后需等待一定时间（判分系统运行时长与参赛选手提交的模型有关），判分系统完成判分后，可在大赛平台“提交作品”页面查看得分。
4. 提交模型后请勿在AI Gallery中下架模型，否则会导致判分失败。
5. 如果使用集成学习，要求被集成模型个数不超过三个。
6. 初赛评审：比赛成绩位于前10名的参赛队伍需要提供可以复现最好成绩的源代码（训练和测试），以及必要模型或者算法原理介绍文档和支撑跑出最好成绩的代码使用文档。
7. 初赛评审阶段将会对选手的代码进行审核，以确保没有作弊行为，否则将取消比赛资



格。若比赛成绩取消，则排名顺延。所提交的代码和文档等数字资产，知识产权归参赛选手所有。所有代码和资料仅用于比赛公平性验证。