经典运筹学问题与模型

(Models for Classical Operations Research Problems)

主讲人：唐加福

东北财经大学 管理科学与工程学院

TEL：0411-8471 1310 jftang@mail.neu.edu.cn

感谢东北大学系统工程研究所课程组

第九章 生产调度问题与模型

- 生产系统布局与设计
- 生产调度问题概述
- 流水车间（Flow shop)调度问题
- Job-Shop 调度问题
- 机器调度问题

生产系统的布局与设计

■ 生产线布局的类型及原则

类型：1 工艺对象专业化布置 （Process Layout）
2 产品对象专业化布置 （Product Layout）
3 混合布置（Hybrid Layout）
4 成组单元布置 （Cell Layout）

原则：1 符合生产过程流向
2 便于运输
3 创造安全、良好的工作环境
4 便于工人操作和工作地（工具、图纸、工位）的布置
5 充分利用车间面积
6 充分考虑机床精度和工作特点

■ 工艺对象专业化布置

如何设置各经济活动单元，改善物流的有序性？

特点：

- ✓ 生产的产品品种较多、生产量不是很大、断续生产；
- ✓ 采用工艺对象专业化的生产组织方式；
- ✓ 设备按照其具有的功能进行排列和布局

机械制造厂的车床、铣床、钻床、磨床等分类布局；

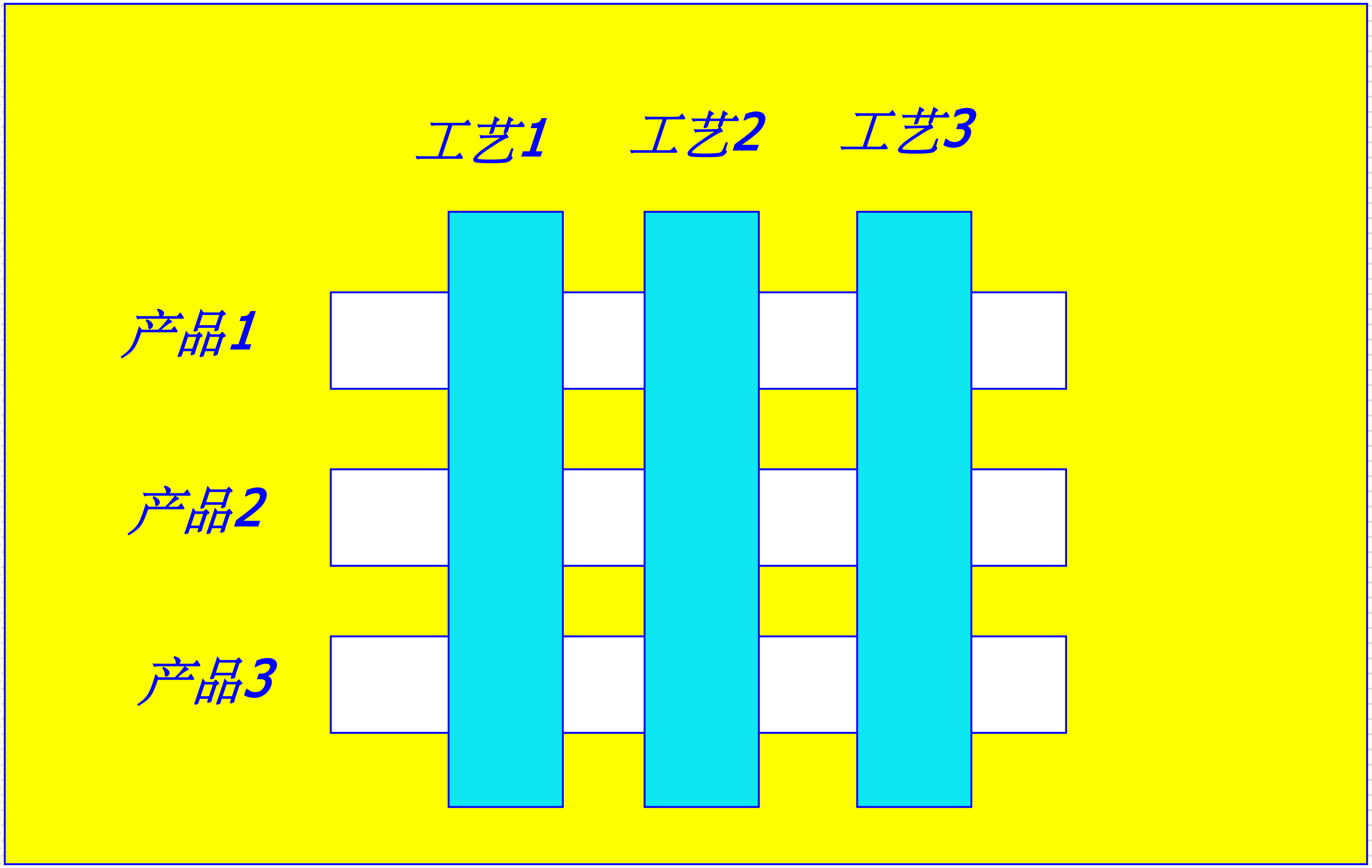
优点：

- ✓ 对产品的适应性较好、设备利用率高

缺点：

- ✓ 产品的物流复杂、流动无序、在制品库存量高、生产周期较长

■ 工艺对象专业化布置（示例）



流水线方式

■ 产品对象专业化布置

特点：

- ✓ 适用于产品品种较少、生产量较大；流水线方式
- ✓ 采用产品对象专业化的生产组织方式；
- ✓ 设备按照产品类的加工路线顺序排列成生产线

L型、 U型、 O型 、S型或混合形状；或多层厂房

优点：

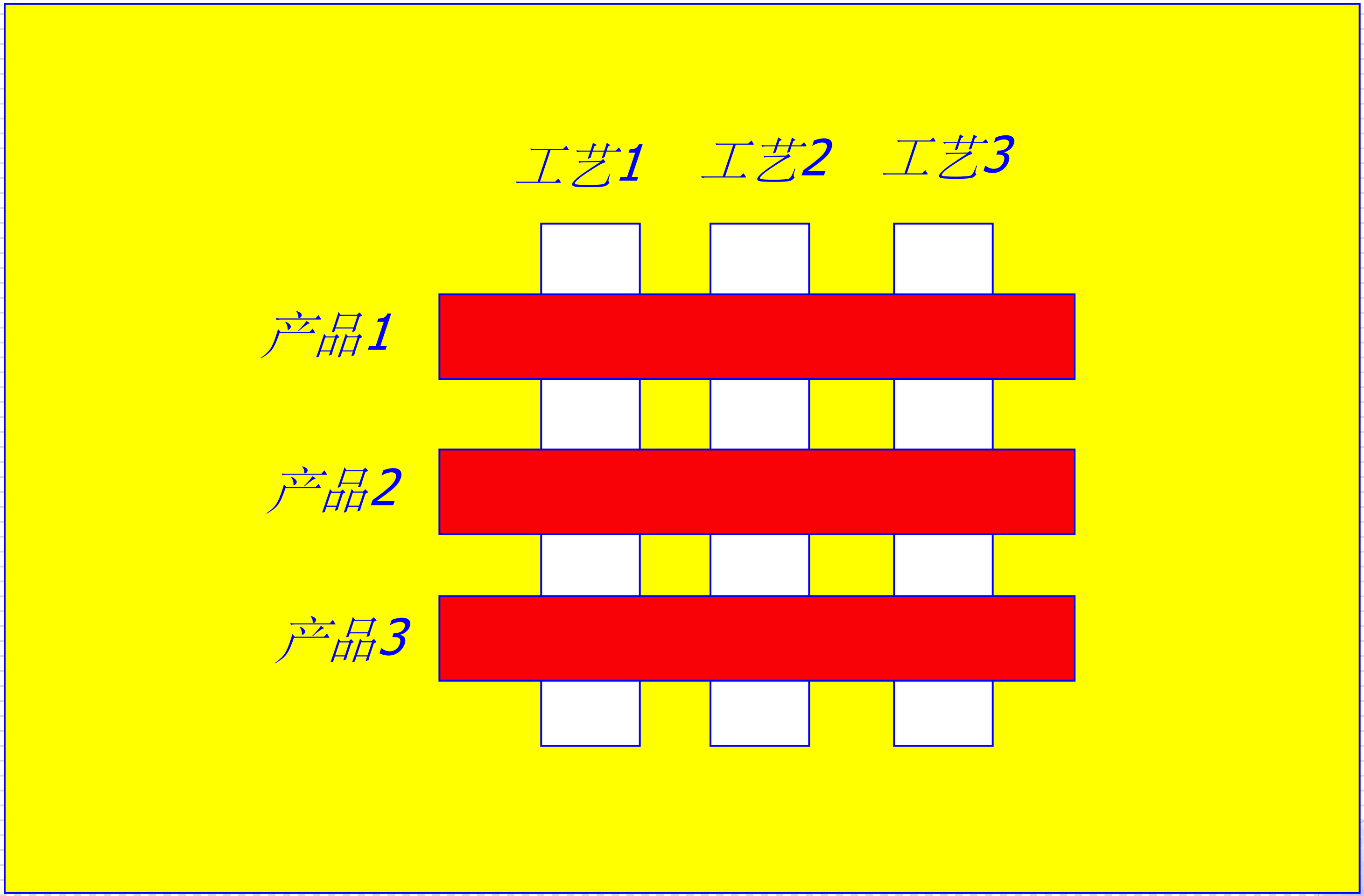
- ✓ 产品物流简单、流动有序、生产节拍和能力平衡便于计算

缺点：

- ✓ 生产周期和产出率取决于流水线的瓶颈能力

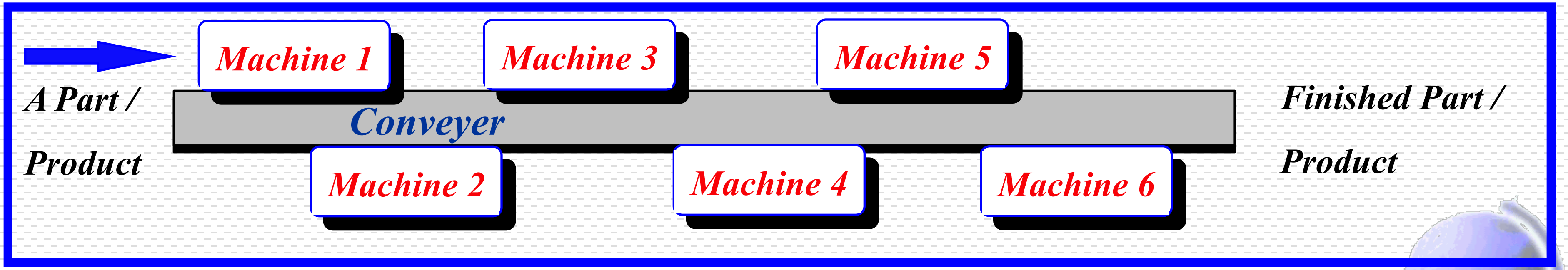
■ 产品对象专业化布置

流水线方式

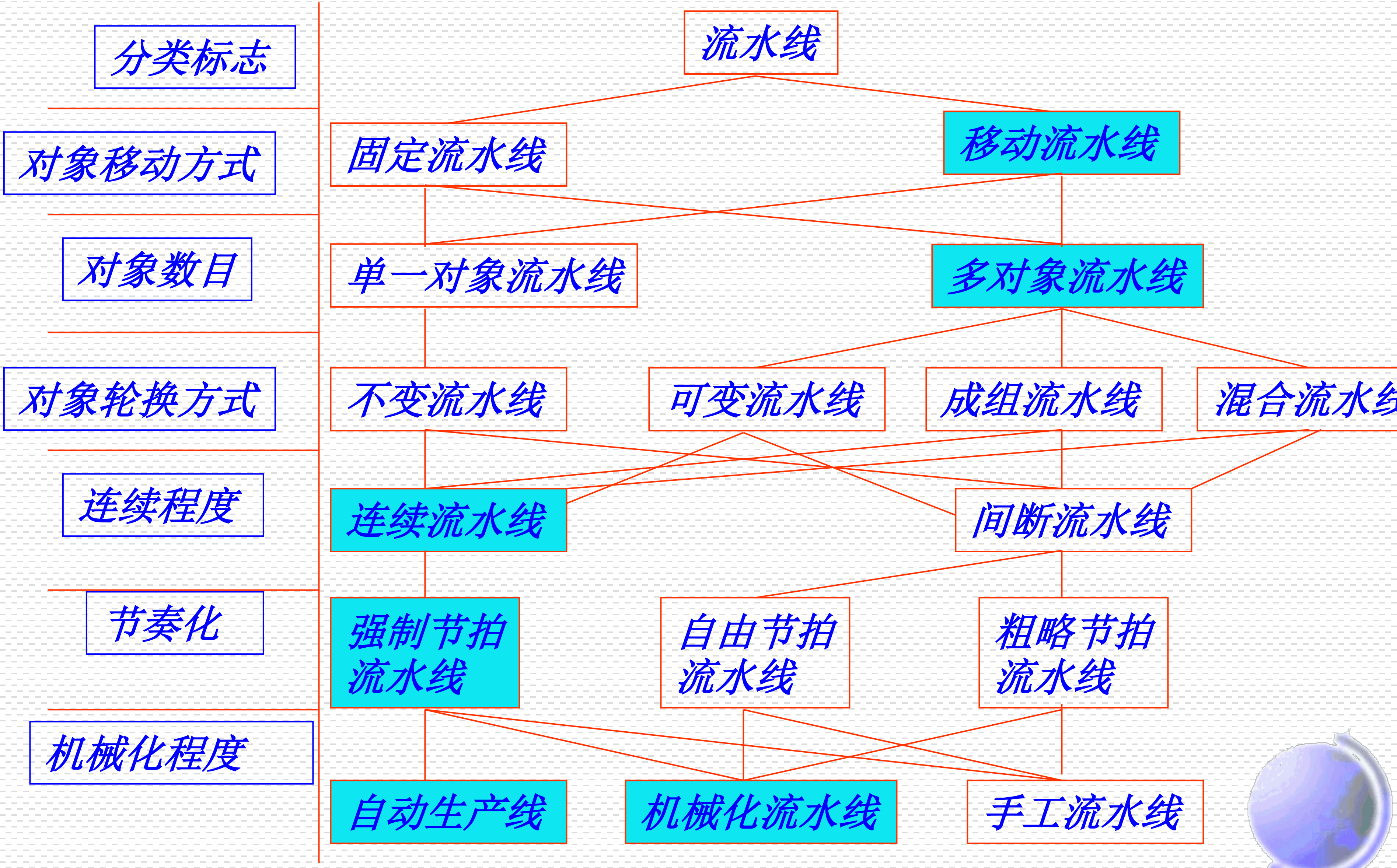


■ 流水生产线的概念与特征

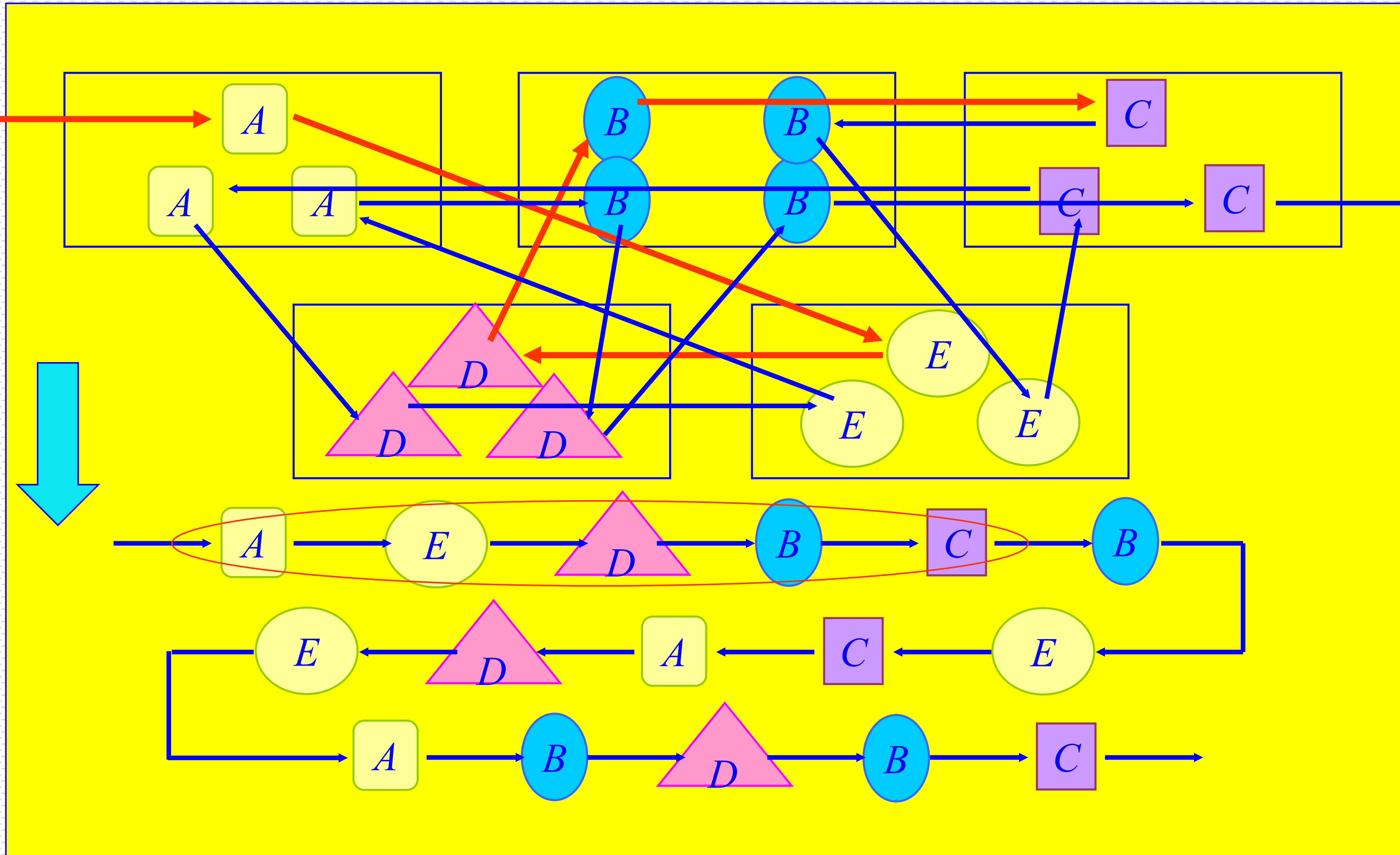
- 劳动对象按一定的工艺路线和统一的生产速度,连续不断地通过各个工作地,顺序地进行加工并出产产品或零件的一种生产组织形式.
- 流水生产的特征
  - 生产连续性高; 工作地专业化程度高
  - 按节拍进行生产;各工序生产能力平衡\成比例
  - 在制品在工序间作单向传送且及时



流水线的分类


分类标志
对象移动方式
对象数目
对象轮换方式
连续程度
节奏化
机械化程度
流水线
固定流水线
移动流水线
单一对象流水线
多对象流水线
不变流水线
可变流水线
成组流水线
混合流水
连续流水线
间断流水线
强制节拍流水线
自由节拍流水线
粗略节拍流水线
自动生产线
机械化流水线
手工流水线

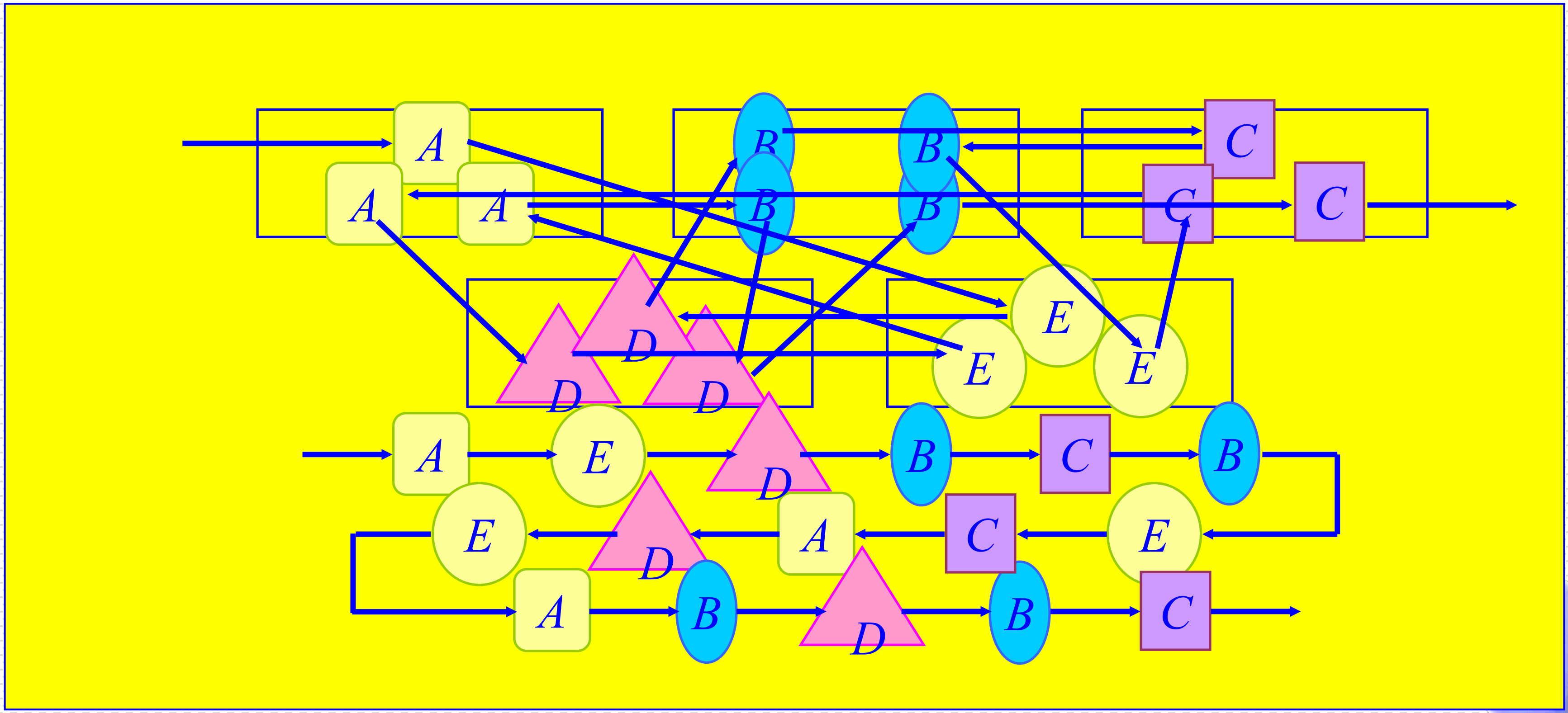
■产品对象专业化布置



■ 混合布局

混合布局: 零部件生产按工艺对象专业化形式，产品组装按产品对象专业化形式，如FMS（柔性制造系统）

固定布局：适用于如船舶制造业、大型装备、机床、发电机组等大型复杂产品



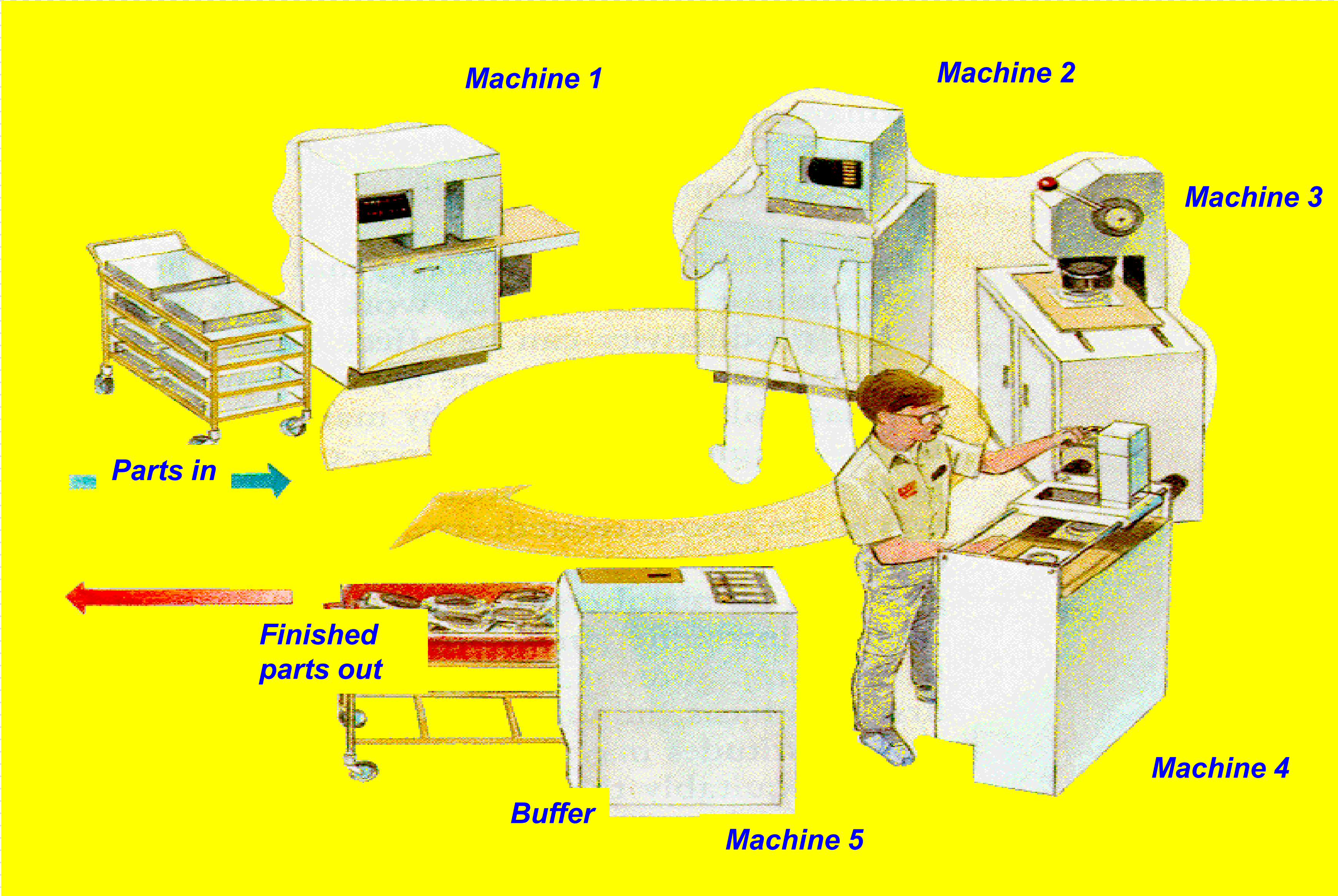
■ 成组单元布置

- ✓ **Job shop**型的单元布局
- ✓ 日本式单元制造型的布局
- ✓ 混合式单元制造布局

将不同的机器分成单元来生产具有相似形状和工艺要求的产品

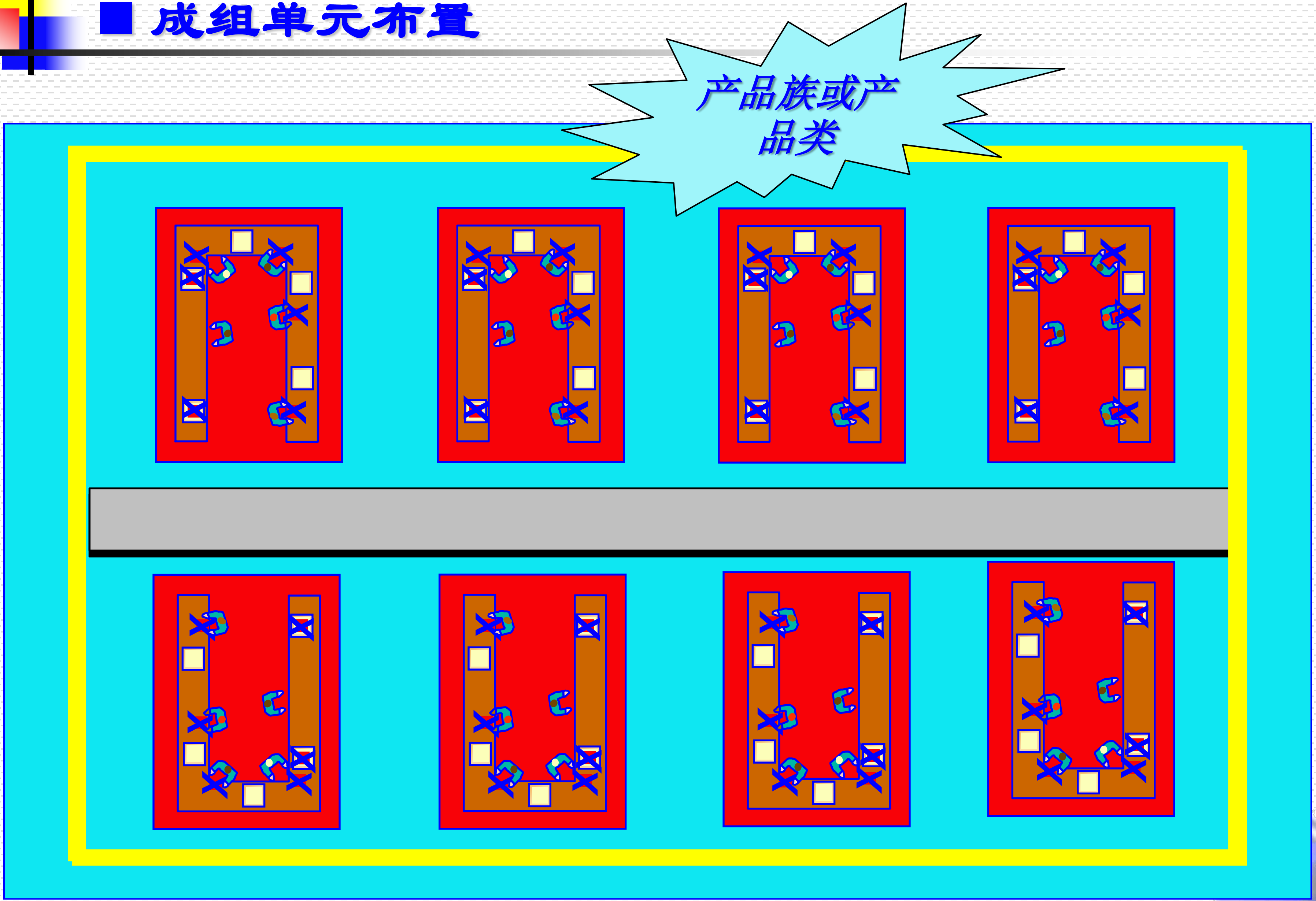
- *改善人际关系，增强参与意识*
- *提高操作技能*
- *减少在制品和物料搬运及生产过程中的存货*
- *缩短生产准备时间*
- *提高机器设备利用率*
- *减少机器设备投资*

■ 成组单元布置



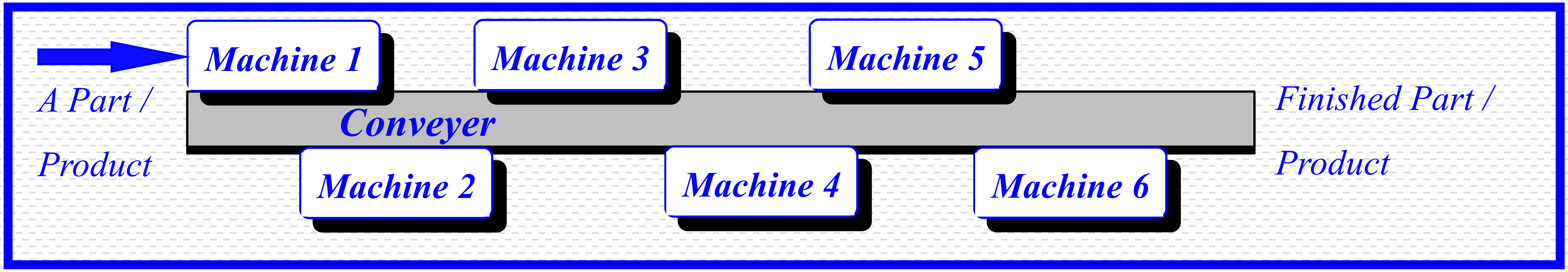
Machine 1
Machine 2
Machine 3
Parts in
Finished
parts out
Buffer
Machine 5
Machine 4

■ 成组单元布置

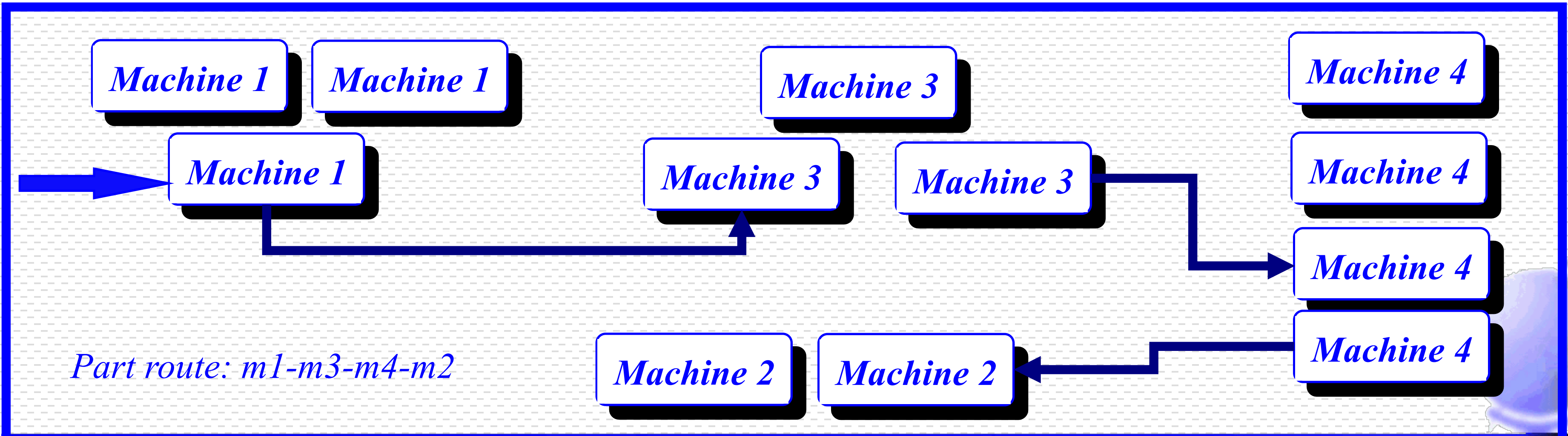


■ 成组单元布置

- **Mass production: Flow line (product line)**



- *Individual production: Job shop (functional layout)*



生产调度问题概述

调度问题概述--特征

- 一类典型的优化问题；调度问题考虑的是：
  - 随着时间的变化，如何调度有限的资源在执行任务的同时满足特定约束。
  - 资源可能在本质上是很不相同的：
    - 人力、金钱、机器、
    - 工具、材料、能源等等。
  - 任务也可以有不同的解释，从制造系统的机器划分到计算机系统的信息处理。一项任务通常可以用下面的因素来表示特征：
    - 完成时间、预期时间；
    - 相对紧急权重、处理时间；资源消耗。
  - 同时，一组反映任务之间先后约束的结构可以用不同的方式定义。
  - 还可以考虑度量调度性能好坏的不同判据.

调度问题概述--特征

- 调度问题几乎在现实环境(特别是工业工程领域)中无处不在。
- 许多制造工业提出的调度问题从本质上讲非常复杂，难以用传统优化方法求解。
- 通常这些难于求解的问题都表示为满足非常复杂约束的组合优化问题。
- 这些问题带有有限数量的可行解。
- 这些问题属于NP—难的问题。

调度问题概述--分类

- 流水车间调度问题
- 作业车间调度问题
- 机器调度问题

- 扩展调度问题：
  - 群体作业调度
  - 资源约束的项目调度
  - 多处理器调度
  - 车辆与路径调度
  - ……

- 按生产计划方式分类
  - 面向订单生产，
    - 强调准时高效，客户订单到达后才开始组织生产。
  - 面向库存生产，
    - 在客户订单到达之前就已开始生产，生产计划以库存为基础。
  - 混合生产模式，
    - 一方面根据预测，保留较大的库存，另一方面以一定的实时生产能力来满足高度客户化的订单需求。

生产调度问题概述--制造业生产模式

- 按生产过程的工艺流程特征分类
  - 离散式生产：
    - 产品是由离散的零部件装配而成，零部件以各自的工艺过程通过各个生产环节，物料运动呈离散状态。
    - 例如属于生产资料生产的机械、电子设备制造业，属于生活资料生产的机电整合消费产品制造业。
    - 离散型制造企业的生产方式多为单件、小批量、多品种
  - 流程式生产：
    - 在生产过程中，物料均匀、连续地按一定工艺顺序运动，生产流程具有连续性的特点和要求
    - 例如冶炼、化工、酿造等
  - 混合流程式生产：
    - 这是一种既具有流程式生产特征，又具有离散式生产特征的复杂生产方式

生产调度问题的几个特点

- **Flow Shop和Job Shop调度问题是最典型和最重要的两种车间调度问题。**
- 车间调度问题具有以下几个特点：
  - 复杂性：
    - 由于生产车间中工件、机器、缓存及搬运系统之间相互影响、相互作用、每个作业又要考虑它的加工时间、操作顺序、交货期等，因而相当复杂。
  - 动态随机性：
    - 在实际的生产调度系统中存在很多随机的和不确定的因素，比如作业到达时间的不确定性、设备的损坏/修复、作业交货期的改变、紧急定单等。
  - 多目标性：
    - 实际的计划调度往往是多目标的。生产调度的性能指标可以是成本最低、库存费最少、生产周期最短、生产切换最少、设备利用率最高、最短的延迟，最小的提前或者拖期惩罚等。这种多目标性导致调度的复杂性和计算量急剧增加。
  - 多约束性：
    - 生产车间中资源的数量、缓存的数量、工件的加工时间和加工顺序都是约束。此外还有一些人为的约束，如要求各机器上的负荷平衡等等。

生产调

表 1-1 生产系统分类

Table 1-1 Classification of Production Systems

分类依据	类型	特征	相关企业
按生产组织类型	产品导向(product focused)	以生产线、装配线为主，工艺过程封闭，投资大，加工批量大	汽车、家电、化工、食品、电子等
	工艺导向(process focused)	将同类工艺所需设备集中在一起，投资小，生产柔性好	机械制造
	模块式生产(module process)	是成组技术的推广，便于零件标准化，有一定批量，设备有重复	机械制造
按需求来源	按库存生产	按预测组织生产	家电等
	按订单生产	按订单组织生产，可细分为按订单装配(ATO)、按订单制造(MTO)、按订单设计(ETO)等	船舶、能源设备、飞机等
按生产批量	连续生产(Continuous production)	产品单一，常年无间歇生产	冶炼、水泥、酿造等
	大量生产(mass production)	产品较多，数量大，间歇性重复生产	汽车、家电、电子等
	批量生产(batch production)	品种多，每个品种较少，产品定型，设备通用，又可分为大批量、中批量、小批量	大多数机械制造行业
	单件生产(job shop production)	品种多，每个品种数量少，设备通用，生产技术准备周期长，生产周期长，成本高	某些重型机械制造行业
	大量客户化生产(mass customization production)	新近发展起来的一种生产方式，强调大量和客户化	计算机、服装等
	项目生产(project production)	产品体积庞大，投资大，周期长，作为项目进行管理	船舶、飞机、路桥、电站建设等

上述第二种分类较为流行，第三种分类最为详细。对于第三种分类来说，属于其中第一类的企业可称为流程型制造企业，属于后几类的企业一般可统称

流水车间调度问题

(Flow-shop Scheduling Problems)

问题描述

- 一般描述
  - n 个工件要在 m 台机器上加工，
  - 每个工件需要经过 m 道工序，每道工序要求不同的机器。
  - n 个工件在 m 台机器上的加工顺序相同。
  - 工件 i 在机器 j 上的加工时间是给定的，设为 t_{ij} $(i=1,\ldots,n;\ j=1,\ldots,m)$。
- 问题的目标
  - 求 n 个工件在每台机器上最优的加工顺序，使最大流程时间达到最小。

问题描述

- 对该问题常常作如下假设:
  - 每个工件在机器上的加工顺序是: **1,2,…,*m***;
  - 每台机器同时只能加工一个工件;
  - 一个工件不能同时在不同的机器上加工;
  - 工序不能预订;
  - 工序的准备时间与顺序无关，且包含在加工时间中;
  - 工件在每台机器上的加工顺序相同，且是确定的。

三类FSP问题

- 确定型流水车间问题
  - 假定工件的加工时间是已知的确定量
- 随机型流水车间问题
  - 加工时间按照一定的概率分布而变化
- 模糊型流水车间问题
  - 每个工件的模糊交货期表示为决策者对工件完成时间的满意度

实际调度问题

- 实际调度问题往往更加复杂，例如：
- 通常要满足一定的约束条件
  - 顾客的交货时间、
  - 资源约束、
  - 工艺约束等
- 性能指标通常可以分成多种类型
  - 最大能力指标
    - 最大生产率、最短生产周期等
  - 成本指标
    - 最大利润、最小化运行费用、最小投资、最大收益等
  - 客户满意度指标
    - 最短的延迟，最小提前或者拖后惩罚等

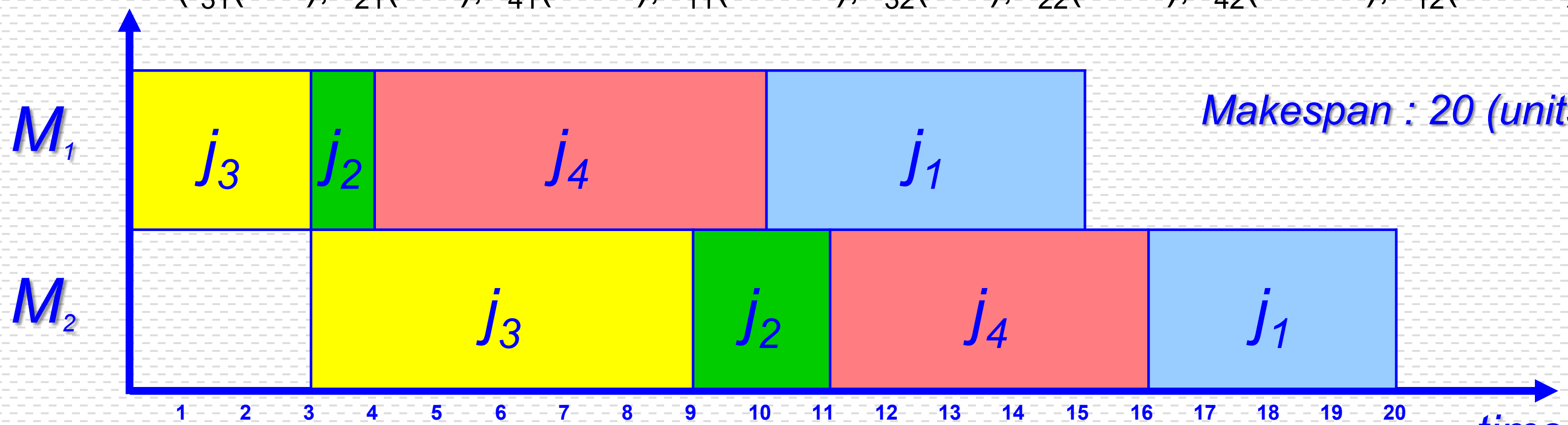
实例

- 确定型FSP的一个实例：4个工件、2台机器
  - 工件顺序：j_3, j_2, j_4, j_1.

Job i	1	2	3	4
$t_{i\,1}$	**5**	**1**	**3**	**6**
$t_{i\,2}$	**4**	**2**	**6**	**5**

  - 调度S如下：

S = (t_{31}(0-3), t_{21}(3-4), t_{41}(4-10), t_{11}(10-15), t_{32}(3-9), t_{22}(9-11), t_{42}(11-16), t_{12}(16-20))



4-job and 2-machine problem

应用

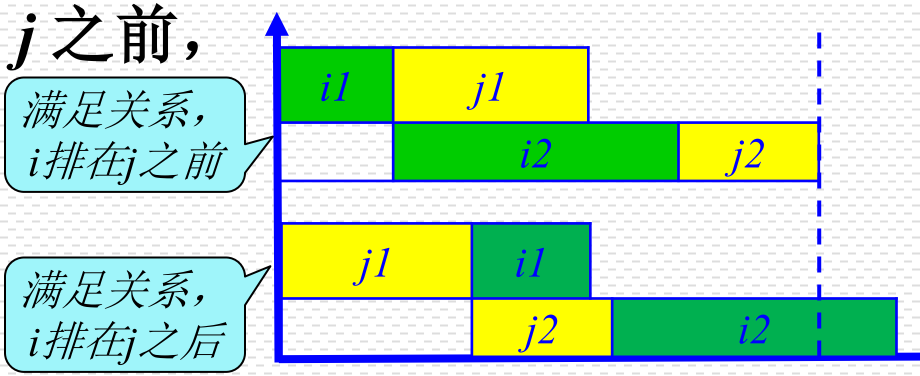
- 墙纸行业生产调度问题
- 装饰布企业车间生产调度
- 飞机维修调度问题
- 化工涂料生产的配料及调度
- 集装箱码头装卸作业的调度问题
- 汽车总装车间调度

- 柔性流水车间调度问题
  - 冷轧厂热处理车间冷卷热处理生产调度问题
  - 炼钢-连铸浇次组合与排序
  - 石油、化工等流程工业

两台机器FSP的最优调度

- **Johnson规则：**
  - 最优调度中工件 i 排在工件 j 之前，
  - 如果：

$$\min\{t_{i1}, t_{j2}\} \le \min\{t_{i2}, t_{j1}\}$$



- **最优顺序可以直接利用这个结果通过两个工序之间的检查来构造。**
  - 令 I 表示工件序列，S 表示顺序，**Johnson**算法可以描述为：

input: t_{i1}, t_{i2}, job list I,

output: best schedule S

step 1: Let $U = \{i \mid t_{i1} < t_{i2}\}$ and $V = \{i \mid t_{i1} \ge t_{i2}\}$

step 2: Sort jobs in U with non-decreasing order of t_{i1}

step 3: Sort jobs in V with non-increasing order of t_{i2}

step 4: An optimal sequence is the ordered set U followed by the ordered set V

8个工件问题的实例

Job i	1	2	3	4	5	6	7	8
t_{i1}	5	2	1	7	6	3	7	5
t_{i2}	2	6	2	5	6	7	2	1

- 最优解构造如下：

step 1: *Job sets U={2, 3, 6} and V={1, 4, 5, 7, 8}*

step 2: *Sort jobs in U as follows:*

Job i : 3 2 6

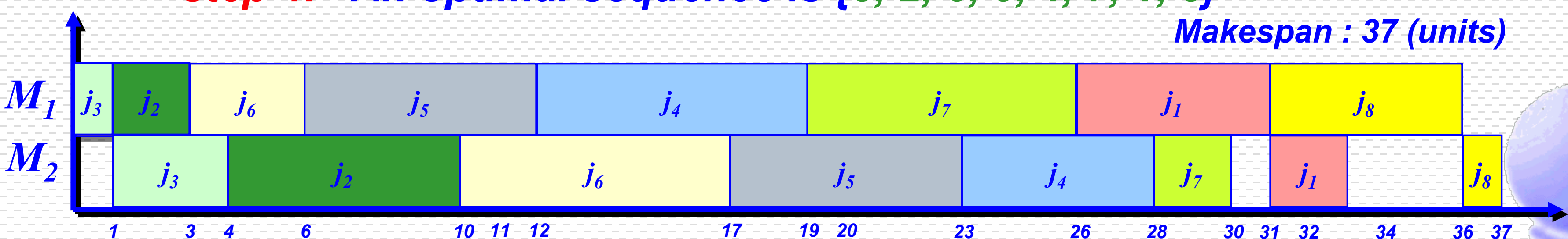
t_{i1} : 1 2 3

step 3: *Sort jobs in V as follows:*

Job i : 5 4 7 1 8

t_{i2} : 6 5 2 2 1

step 4: *An optimal sequence is {3, 2, 6, 5, 4, 7, 1, 8}*



m台机器的启发式算法

- Palmer启发式算法
- Gupta启发式算法
- CDS启发式算法
- RA启发式算法
- NEH启发式算法

作业车间调度问题

(Job-shop Scheduling Problems)

问题描述

- **古典作业车间问题：**
  - 有 m 台机器和 n 个工件；
  - 每个工件包含一个由多道工序组成的工序集合；
  - 工件的工序顺序是预先给定的；
  - 每道工序要在它所要求的机器上加工某一固定时间;
- **问题的目标：**
  - 确定机器上工序的加工顺序，使最大流程时间，即完成所有工件的时间最短.

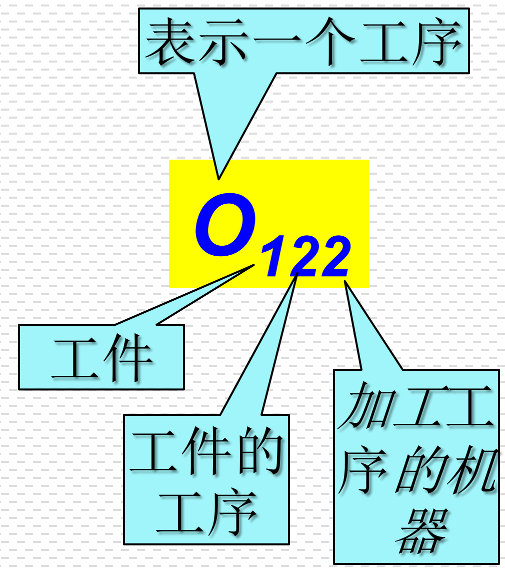
问题描述

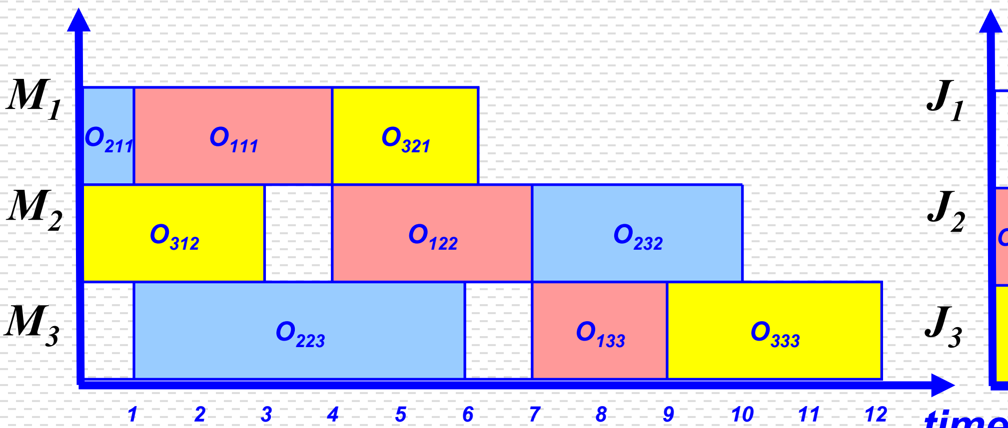
- 对工件和机器有如下约束：
  - 一个工件不能两次访问同一机器；
  - 不同工件的工序之间没有先后约束；
  - 工序一旦进行不能中断；
  - 每台机器一次只能加工一个工件；
  - 下达时间和交货期都不是给定的。

实例

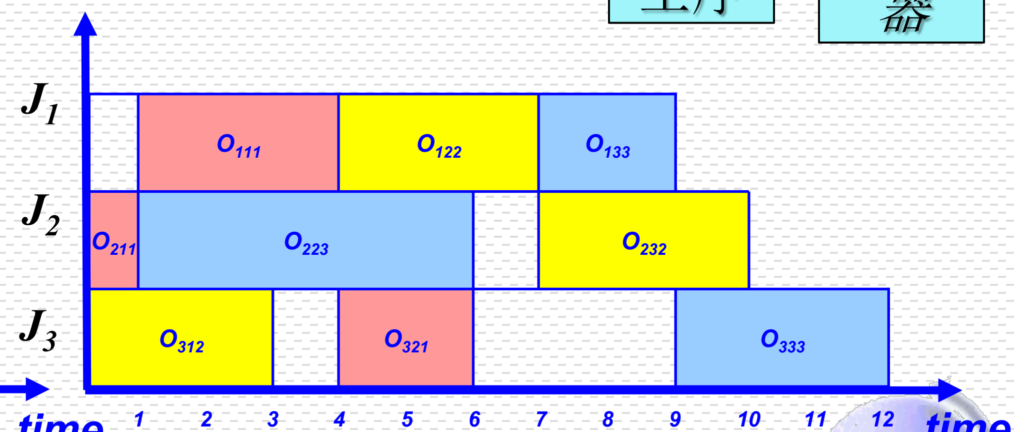
- 3个工件、3个机器的问题，可以用两类甘特图来表示

Job	Processing Time: Operations 1	2	3	Job	Machine Sequence: Operations 1	2	3
J_1	3	3	2	J_1	M_1	M_2	M_3
J_2	1	5	3	J_2	M_1	M_3	M_2
J_3	3	2	3	J_3	M_2	M_1	M_3





Machine Gantt chart



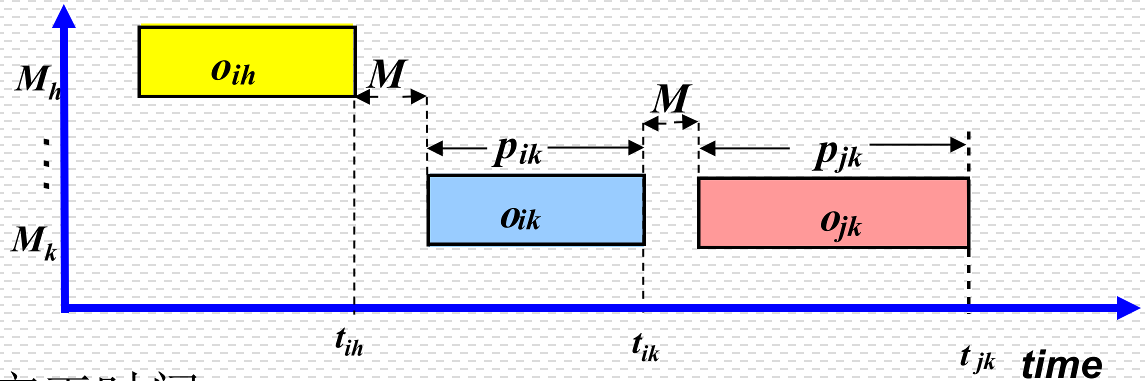
Job Gantt chart

应用

- 离散型企业的车间调度问题
  - 离散型制造业的产品往往由多个零件经过一系列并不连续的工序的加工最终装配而成。
  - 例如属于生产资料生产的机械、电子设备制造业，
  - 属于生活资料生产的机电整合消费产品制造业。
  - 作业车间调度问题是系统运行不可或缺的核心技术
- 模具企业生产调度问题
- 铁路列车运行调整问题
- 船用柴油机生产计划与调度
- 汽车制造业中，冲压车间生产计划和调度
- 数字媒体开发过程中的资源配置问题
- 家纺企业车间调度问题
- 钢铁企业中
  - 钢管生产调度问题
  - 冷轧生产线的机组作业调度问题

整数规划模型

- Baker讨论了JSP问题的整数规划模型
- 索引号
  - i, j : 工件的索引
  - h, k : 机器的索引
- 参数
  - n : 工件数
  - m : 机器数
  - t_{jk} : 工件j 在机器k上的完工时间
  - p_{jk}: 工件 j 在机器k上的加工时间
  - M : 很大的正数



Time chart for model

- 决策变量
  - $$a_{ihk} = \begin{cases} 1, & \text{工件}i\text{在机器}h\text{上的加工先于机器}k \\ 0, & \text{否则} \end{cases}$$
  - $$x_{ijk} = \begin{cases} 1, & \text{工件 } i \text{ 先于工件 } j \text{ 来到机器 } k \\ 0, & \text{否则} \end{cases}$$

整数规划模型

- 以最大流程时间最短为目标的JSP问题可描述如下：

$$\min\ t_M = \max_{1\le k\le m}\left\{\max_{1\le i\le n}\{t_{ik}\}\right\} \quad (1)$$

s.t.

$$t_{ik} - p_{ik} + M(1-a_{ihk}) \ge t_{ih,}\ i=1,2,\dots,n;h,k=1,2,\dots,m \quad (2)$$

$$t_{jk} - t_{ik} + M\left(1-x_{ijk}\right) \ge p_{jk,}\ i,j=1,2,\dots,n;k=1,2,\dots,m \quad (3)$$

$$t_{ik} \ge 0,\quad i=1,2,\dots,n;k=1,2,\dots,m \quad (4)$$

$$x_{ijk},\ \mathrm{a}_{ihk} = 0 \mathrm{or} 1,\quad i,j=1,2,\dots,n;h,k=1,2,\dots,m \quad (5)$$

考虑的是机器的工序非堵塞关系：多个工件不能在同一机器上同时加工。

考虑的是工序之间的前后关系：一个工件不能在不同的机器上同时加工。

线性规划模型

- Adams等人讨论了作业车间调度问题的线性规划(LP)模型
- 以最大流程时间最短为目标，模型描述如下：

保证每个工件的工序顺序满足预先的要求

$$
\begin{aligned}
\min \quad & t_n && && (1)\\
\text{s. t.} \quad & t_j - t_i \ge d_i, && (i,j)\in A && (2)\\
& t_j - t_i \ge d_i \ \text{ or } \ t_i - t_j \ge d_j && (i,j)\in E_k,\quad k\in M && (3)\\
& t_i \ge 0, && i = 1,2,\dots,n && (4)
\end{aligned}
$$

保证每台机器一次只能加工一个工件

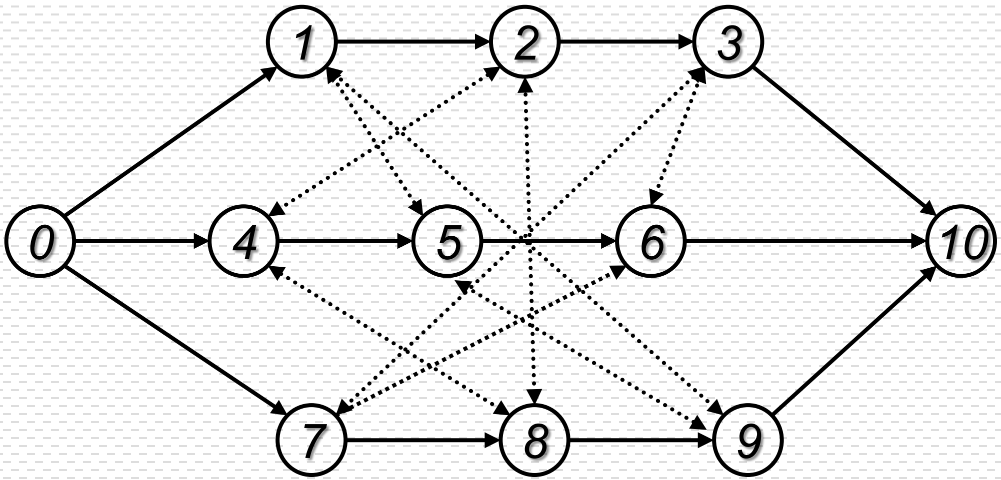
- $N=\{0,1,2,\ldots,n\}$ -- 表示工序的集合，**0**, ***n*** 表示虚设的起始和完成工序;
- $M=\{1,2,\ldots,m\}$ -- 是机器的集合;
- A -- 表示每个工件的工序前后关系约束的工序对集合;
- E_k --是机器 ***k*** 上加工的工序对集合(没有明确先后关系);
- d_i -- 工序 ***i*** 的加工时间，是一定的;
- t_i -- 工序的起始时间，优化过程中有待确定的变量。

图模型

- 作业车间调度问题可以用非连接图来表示
- 非连接图： $G = (N, A, E)$ 定义为：
  - N：包含代表所有工序的节点；
  - A：包含连接同一工件的邻接工序的边
  - E：包含连接同一机器上加工工序的非连接边
    - 所谓非连接边是可以有两个可能方向的边。
- 一个可行的调度：
  - 将固定所有非连接边的方向，以确定同一机器上的工序顺序；
  - 一旦对于某台机器的顺序确定下来，连接工序的非连接边将被通常的带优先箭头的连接边取代
  - 非连接边集 E 分解成子集系E_k;
    - $E=E_1 \cup E_2 \cup E_3 ... \cup E_m$，每台机器一个系

图模型——实例

- 3个工件、3台机器的非连接图，每个工件包含3道工序



- 节点N = {0,1.2,3,4,5,6,7,8,9,10}对应工序，其中0和10是虚设的起始工序和终止工序；
- 连接边 A={(1,2), (2,3), (4,5), (5,6), (7,8), (8,9)} 对应同一工件的工序前后约束；
- 非连接边(虚线表示)：
  - E_1={(1,5),(5,9),(9,1)}对应机器1上加工的工序；
  - E_2={(4,2),(2,8),(8,4)}对应机器2上加工的工序；
  - E_3={(7,3),(3,6),(6,7)}对应机器3上加工的工序。

传统的启发式算法

- 分类:
  - 一次通过启发式;
    - 一次通过启发式指通过基于优先分配规则一次固定一个工序，简单地构造一个完全解。有多种规则可用来从特定的子集中选择工序作为下一步的调度。
    - 这种启发式的特点是快，而且能找到不太坏的解。
  - 多次通过启发式;
    - 可以重复应用一次通过启发式来建立更为复杂的多次通过启发式，以花费额外的计算费用来获得更好的调度。
- 算法:
  - 优先分配启发式
  - 随机分配启发式
  - 瓶颈移动启发式
  - …

遗传算法求解

- 由于受工序的加工路线的约束，作业车间调度问题不像旅行商问题(TSP)那样容易确定一个自然表达。
- 到目前为止，还没有一个用系统不等式来表达先后约束的好方法。因此，不便于应用惩罚法来处理这些类型的约束。
- 大多数遗传算法和JSP问题的研究者喜欢使用修复策略来处理非可行性和非法性。
- 构造JSP问题遗传算法的重要主题：
  - 如何设计一个合适的表达解的方法和基于特定问题的遗传算子，使得不管在初期还是在进化过程中产生的所有染色体都将产生可行的调度，而这将是影响遗传算法的各个阶段的关键阶段。

作业车间的表达方法

- 直接方法
  - 一个调度被编码为一个染色体，应用遗传算法来进化染色体以确定一个好的调度。
    - 基于工序的表达法;
    - 基于工件的表达法;
    - 基于工件对关系的表达法;
    - 基于完成时间的表达法;
    - 随机键表达法都属于此类。
- 间接方法
  - 例如基于优先规则表达法，工件调度的分配规则序列被编码为一个染色体，应用遗传算法来进化染色体以确定一个好的分配规则序列。
    - 基于优先表的表达法;
    - 基于优先规则表达法;
    - 基于非连接图表达法;
    - 基于机器表达法都属于此类。

基于工序的表达

- 对于 *n* 工件 *m* 机器问题，一个染色体包括 *n*m* 个基因。
- 每个工件出现在染色体中 *m* 次，每个基因不表明一个工件的具体的工序，而是指有上下依赖关系的工序。
- 很容易看出染色体的任意排列总能产生可行调度。

- 例如：3个工件3台机器的调度问题
  - 这个表反映了，工件的各工序的加工时间、在哪台机器上加工、加工顺序是什么。

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

染色体编码的例子: v =

3	2	1	3	2	2	1	1	3

基于工序的表达

- 染色体解码说明：

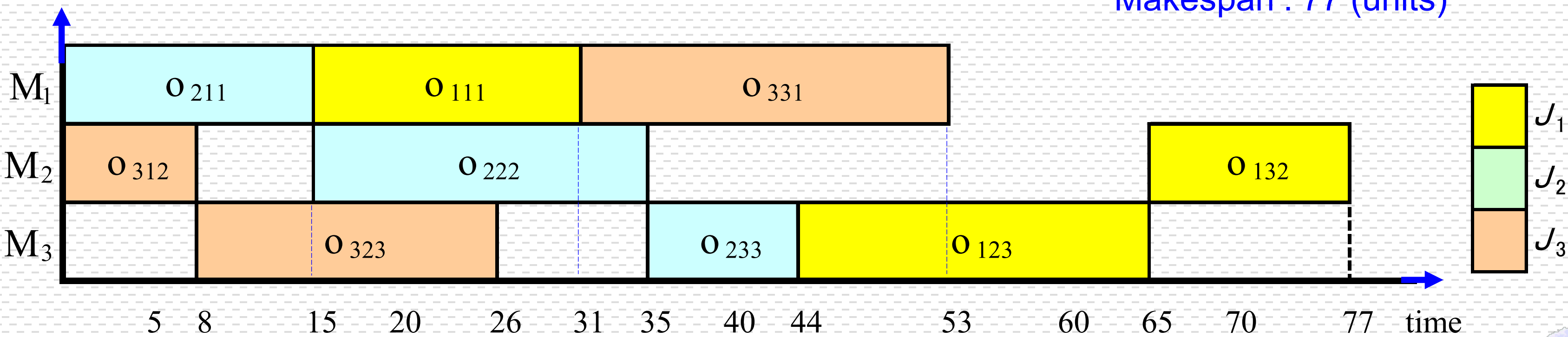
(1 : job 1, 2 : job 2, 3 : job 3)

染色体编码： v = 3 2 1 3 2 2 1 1 3

对应的加工机器： 2 1 1 3 2 3 3 2 1

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

Makespan : 77 (units)



Gantt chart of one feasible schedule

基于工件的表达

- 对于 n 工件 m 机器问题，一个染色体由包含 n 个工件的列表组成。
- 每个调度根据工件的顺序来构造。
- 对于一个给定的工件顺序，列表中第一个工件的所有工序将被首先调度，然后考虑列表中第二个工件的工序。
- 加工中的工件第一道工序要求的加工时间应为相应机器最可能得到的加工时间，然后考虑第二道工序，如此进行，直到工件的所有工序排完。
- 这个过程以列表中每个工件的适当顺序重复进行。
- 工件的任意序列对应一个可行的调度。

- 例如：3个工件3台机器的调度问题

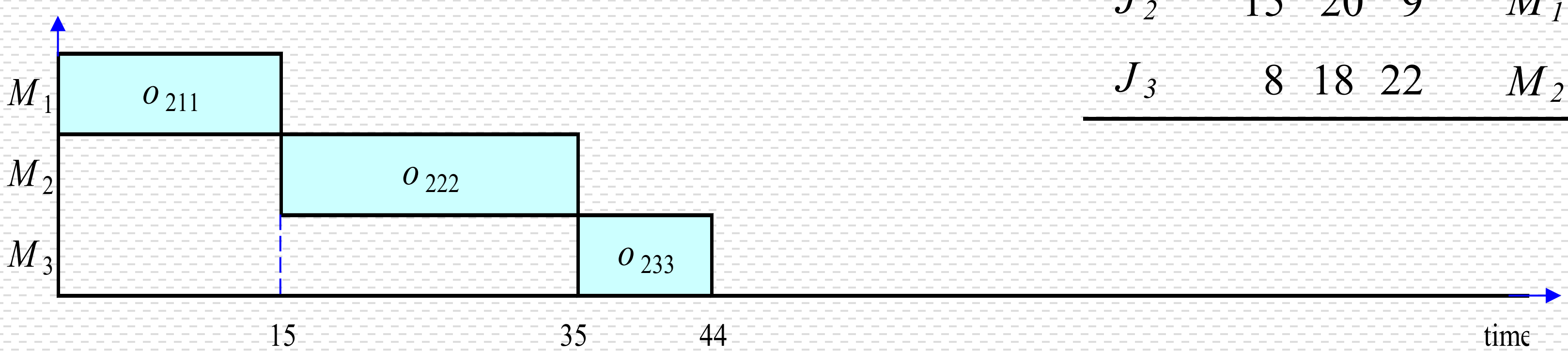
	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

染色体编码的例子: v = | 2 | 3 | 1 |

- 染色体解码说明：
  - 假设给定染色体为：v=[2 3 1]
  - 第一个要加工的工件是：J_2，
  - 工件J_2的工序前后约束是：[m_1, m_2, m_3]
  - 每台机器的相应的加工时间为：[15, 20, 9]
  - 首先，J_2被调度如下。

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

染色体编码：v = | 2 | 3 | 1 |



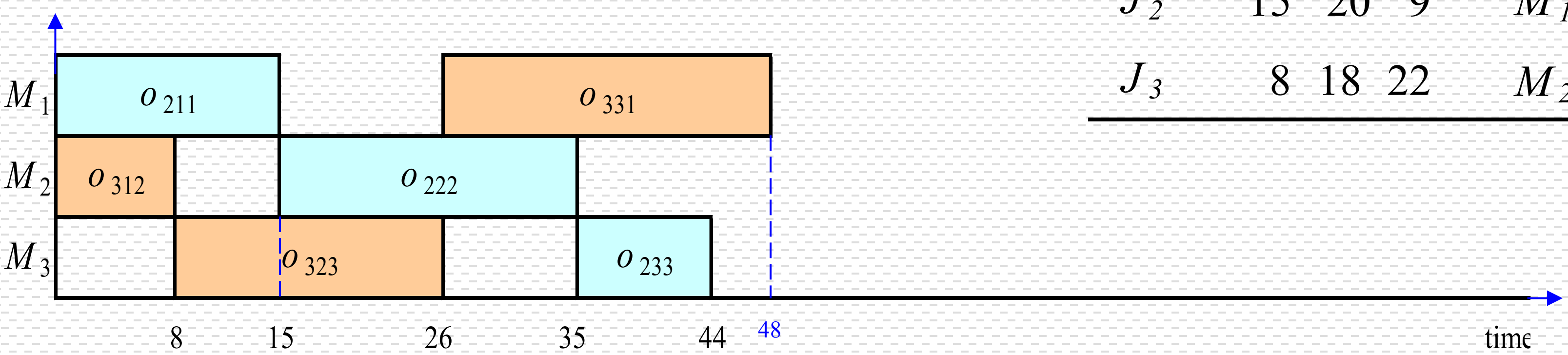
Gantt chart for the first job 2

基于工件的表达

- **染色体解码说明：**
  - 第二个要加工的工件是：J_3，
  - 工件J_3的工序前后约束是： $[m_2, m_3, m_1]$
  - 每台机器的相应的加工时间为：[8, 18, 22]
  - 其次， J_3被调度如下。

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

染色体编码： v = | 2 | 3 | 1 |



Gantt chart for the second job 3

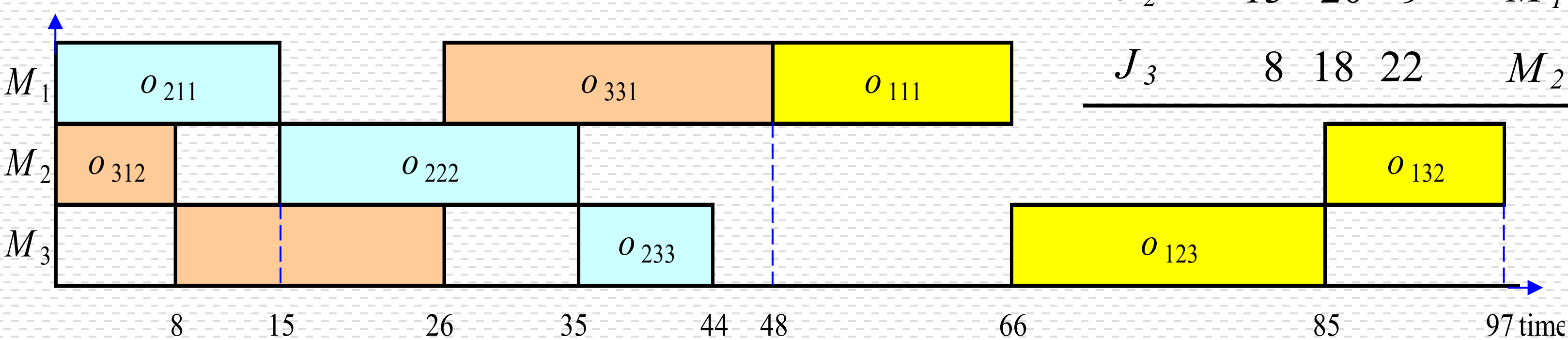
基于工件的表达

- 染色体解码说明：
  - 第三个要加工的工件是：J_1,
  - 工件J_1的工序前后约束是： $[m_1, m_3, m_2]$
  - 每台机器的相应的加工时间为：[16, 21, 12]
  - 最后， J_1被调度如下。

染色体编码: v = | 2 | 3 | 1 |

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

Makespan : 97 (units)



Gantt chart for the last job 1

基于优先表的表达

- 对于 n 工件 m 机器问题，一个染色体由 m 个子染色体组成，
- 每个子染色体对应一台机器，由一串长度为 n 的符号表示，每个符号代表一道在相关机器上处理的工序。
- 子染色体不能描述机器上工序的先后顺序，因为它们是优先列表，每台机器有其自身的优先列表。
- 实际的调度是由染色体通过模拟分析机器前等待排队的状态得到的，
- 如果有必要，用优先列表来确定调度，即选择首先出现在优先列表中的工序。
- 例如：3个工件3台机器的调度问题

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

染色体编码的例子：v =

2	3	1	1	3	2	2	1	3

基于优先表的表达

- **染色体解码说明：**
  - 根据优先列表可知第一道优先的工序是：机器M_1上加工工件J_2，M_2上加工J_1，M_3上加工J_2。
  - 根据给定的工序先后约束，只有机器M_1上加工工件J_2是可行的调度。因此首先对机器M_1调度。

染色体：v =

2	3	1	1	3	2	2	1	3

机器 M_1 的优先列表（2 3 1）；机器 M_2 的优先列表（1 3 2）；机器 M_3 的优先列表（2 1 3）

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

基于优先表的表达

- **染色体解码说明：**
  - 优先工序为：M_2上加工J_1，M_3上加工J_2。
  - 第二道优先工序为：机器M_1上加工工件J_3，M_2上加工J_3，M_3上加工J_1。
  - 只有机器M_2上加工工件J_3是可行的调度。因此对机器M_2调度。

染色体： v = | 2 | 3 | 1 | 1 | 3 | 2 | 2 | 1 | 3 |

机器 M_1 的优先列表：2 3 1

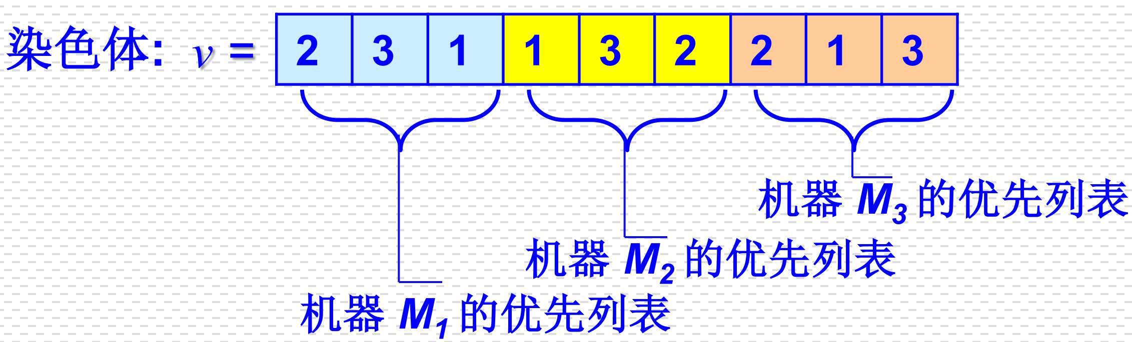
机器 M_2 的优先列表：1 3 2

机器 M_3 的优先列表：2 1 3

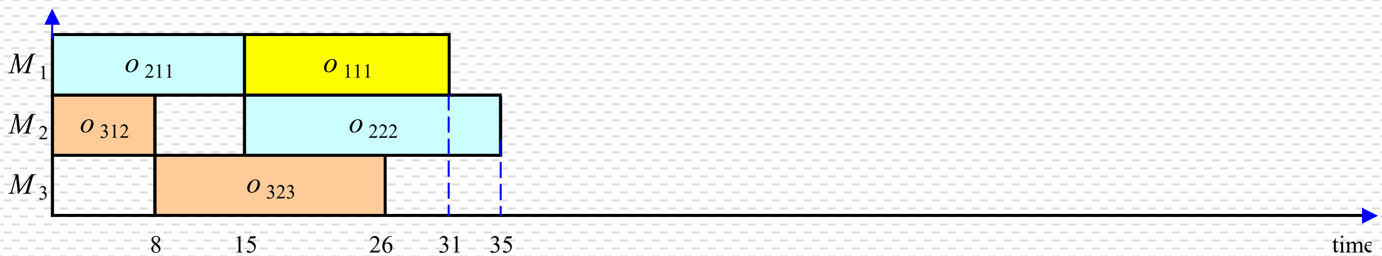
	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

基于优先表的表达

- 染色体解码说明：
  - 优先工序为：M_2上加工J_1，M_3上加工J_2，M_1上加工J_3，M_3上加工J_1。
  - 下一道优先工序为：机器M_1上加工工件J_1，M_2上加工J_2，M_3上加工J_3。
  - 可调度的工序是：M_1上加工J_1，M_2上加工J_2，M_3上加工J_3。

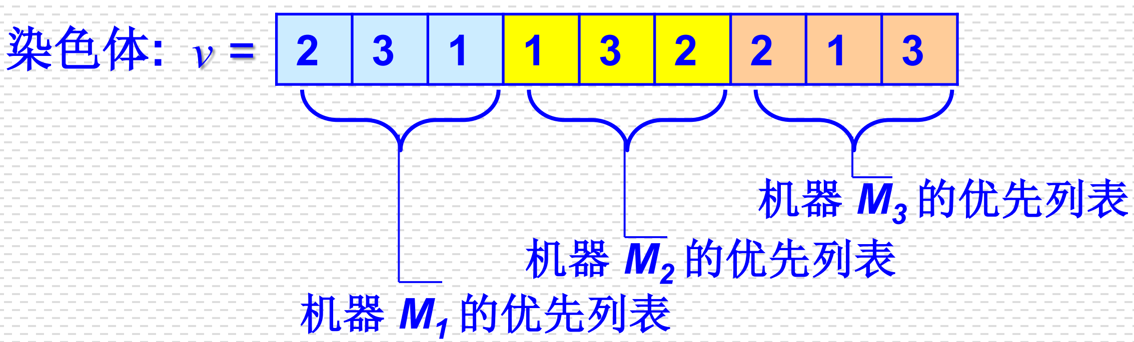


	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

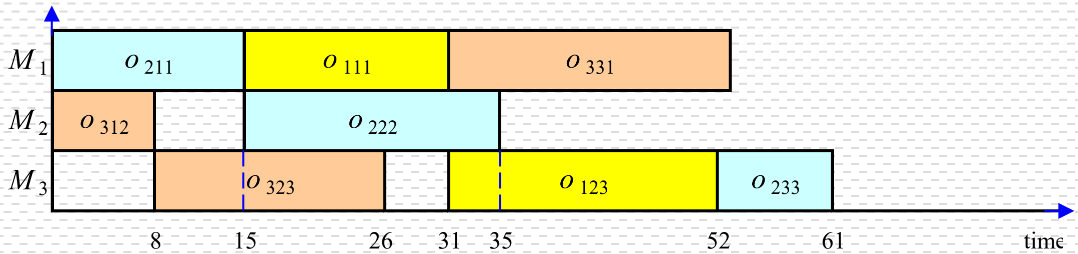


基于优先表的表达

- **染色体解码说明：**
  - 优先工序为：M_2上加工J_1，M_3上加工J_2，M_1上加工J_3，M_3上加工J_1。
  - 可调度的工序是：M_3上加工J_1，M_3上加工J_2，M_1上加工J_3。



	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1



基于优先表的表达

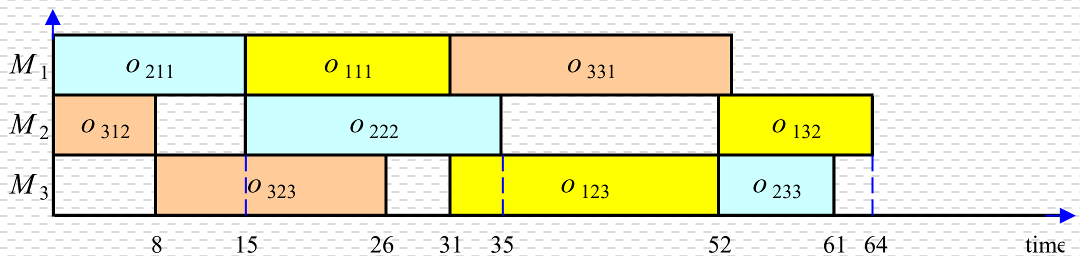
- **染色体解码说明：**
  - 优先工序为：M_2上加工J_1。
  - 可调度的工序是：M_2上加工J_1。

染色体：v =

2	3	1	1	3	2	2	1	3

机器 M_1 的优先列表 (2 3 1)；机器 M_2 的优先列表 (1 3 2)；机器 M_3 的优先列表 (2 1 3)

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1



基于工件对关系的表达

- 采用二进制矩阵编码一个调度。矩阵根据相应机器上工件对的前后关系来确定。
- 例如：3个工件3台机器的调度问题。
  - 工件的工序前后约束和问题的一个可行调度如表所示。

工序先后约束				可行调度			
Job	机器顺序			Machine	工件调度		
J_1	M_1	M_2	M_3	M_1	J_2	J_1	J_3
J_2	M_1	M_3	M_2	M_2	J_3	J_1	J_2
J_3	M_2	M_1	M_3	M_3	J_2	J_1	J_3

  - 用一个二进制变量来表示工件对的先后关系，定义如下：

$$x_{ijm} = \begin{cases} 1, & \text{如果工件 } i \text{ 优先于工件 } j \text{ 在机器 } m \text{ 上加工} \\ 0, & \text{其它} \end{cases}$$

  - 对于给定的可行调度，可以得到一个二进制矩阵表达法：

$$\begin{array}{l} (J_1,J_2)\text{ on }(M_1,M_2,M_3): \\ (J_1,J_3)\text{ on }(M_1,M_2,M_3): \\ (J_2,J_3)\text{ on }(M_1,M_3,M_2): \end{array} \begin{pmatrix} x_{121} & x_{122} & x_{123} \\ x_{131} & x_{132} & x_{133} \\ x_{231} & x_{233} & x_{232} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

基于优先规则的表达

- Dorndorf和Pesch提出了一种基于优先规则的遗传算法，
- 其中染色体编码为一个工件分配规则的序列。
- 基于分配规则序列，用优先分配启发式构造调度。

- 对于 n 工件 m 机器问题，染色体是一个 $n*m$ 个项的串（$p_1, p_2 \ldots, p_{nm}$），p_i 代表预先设定的优先分配规则集合的一个规则。处于第i个位置的元素表达在算法的第i步迭代中的冲突应该由优先规则 p_i 来重新处理。更精确地说，用规则p_i 从冲突集合中选取工序，采用随机选择来断开链。

- 例如：选择的优先规则如下表

序号	规则	描述
1	SPT	选择具有最短加工时间的工序
2	LPT	选择具有最长加工时间的工序
3	MWR	选择剩余加工时间最长的工序
4	LWR	选择剩余加工时间最短的工序

染色体编码的例子：v =

1	2	2	1	4	4	2	1	3

基于优先规则的表达

- **算法参数说明：**

$PS_t =$ 包含 t 步已调度工序的部分调度;

$S_t =$ 对应于 PS_t 在迭代 t 时可调度工序的集合;

$\sigma_i =$ 可调度工序集中,工序能够开始的最早时间 $i \in S_t$;

$\phi_i =$ 可调度工序集中,工序能够完成的最早时间 $i \in S_t$;

$C_t =$ 迭代 t 时冲突工序的集合.

- **基于优先规则的调度过程：**

step 1. 让 t=1， PS_t 为空， S_t 包含所有无紧前工序的工序。

step 2. 确定 $\phi_t^*=\min_{i\in S_t}\{\phi_i\}$ 和能实现 ϕ_t^* 的机器 m^* ；
如果存在多个这样的机器，则通过随机选择来断开链。

step 3. 形成一个冲突集合 C_t，它包含需要 m^* 进行加工的工序且满足 $i\in S_t$, $\sigma_i<\phi_t^*$。
从 C_t 中根据优先规则 p_t 选择一道工序尽可能早地添加到 PS_t 中，
产生一个新的部分调度 PS_{t+1}，如果存在多道工序可选，则随机选择一个。

step 4. 从 S_t 中移出被选择的工序，将直接后续工序添加到 S_t，
更新 PS_{t+1}， t=t+1

step 5. 返回 step 2 直到产生一个完整调度

基于优先规则的表达

- 染色体解码说明：

v =

1	2	2	1	4	4	2	1	3

第1步, $t=1$:

$PS_t = \{\}$;

$S_t = \{o_{111}, o_{211}, o_{312}\}$;

$\sigma_i = \{0,0,0\}$;

$\phi_i^* = \min\{16,15,8\} = 8$;

$m^* = 2$

$C_t = \{o_{312}\}$

$PS_{t+1} = \{o_{312}\}$

$S_{t+1} = \{o_{111}, o_{211}, o_{323}\}$;

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

No	规则	描述
1	SPT	选择具有最短加工时间的工序
2	LPT	选择具有最长短加工时间的工序
3	MWR	选择剩余加工时间最长的工序
4	LWR	选择剩余加工时间最短的工序

基于优先规则的表达

- 染色体解码说明：

v = | 1 | 2 | 2 | 1 | 4 | 4 | 2 | 1 | 3 |

第2步, $t=2$:

$$PS_t = \{o_{312}\};$$
$$S_t = \{o_{111}, o_{211}, o_{323}\};$$
$$\sigma_i = \{0,0,8\};$$
$$\phi_i^* = \min\{16,15,26\} = 15;$$
$$m^* = 1$$
$$C_t = \{o_{111}, o_{211}\}$$
$$PS_{t+1} = \{o_{312}, o_{111}\}$$
$$S_{t+1} = \{o_{123}, o_{211}, o_{323}\};$$

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

No	规则	描述
1	SPT	选择具有最短加工时间的工序
2	LPT	选择具有最长加工时间的工序
3	MWR	选择剩余加工时间最长的工序
4	LWR	选择剩余加工时间最短的工序

基于优先规则的表达

- **染色体解码说明：**

$v =$

1	2	2	1	4	4	2	1	3

第3步, $t = 3$:

$$PS_t = \{o_{312}, o_{111}\};$$

$$S_t = \{o_{123}, o_{211}, o_{323}\};$$

$$\sigma_i = \{16,16,8\};$$

$$\phi_i^* = \min\{37,31,26\} = 26;$$

$$m^* = 3$$

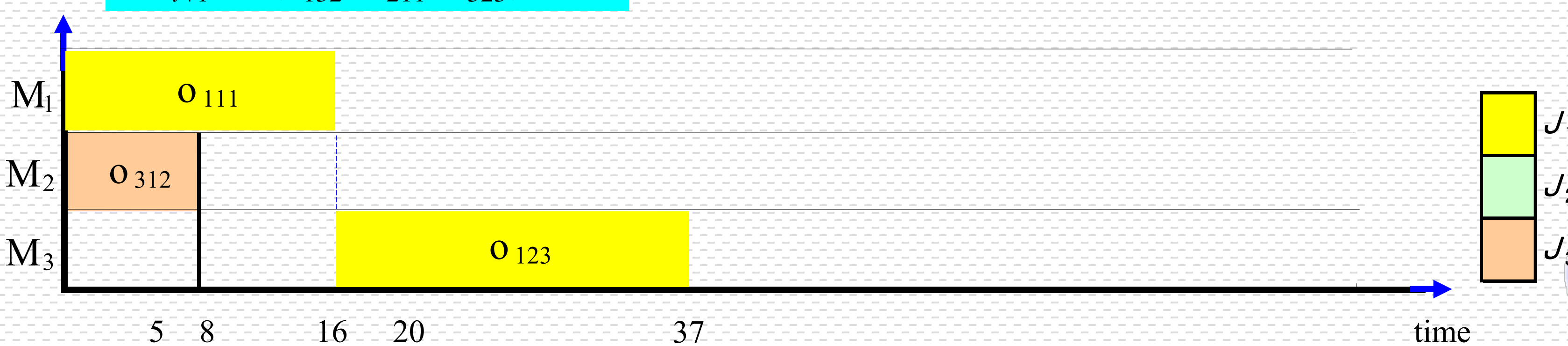
$$C_t = \{o_{123}, o_{323}\}$$

$$PS_{t+1} = \{o_{312}, o_{111}, o_{123}\}$$

$$S_{t+1} = \{o_{132}, o_{211}, o_{323}\};$$

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

No	规则	描述
1	SPT	选择具有最短加工时间的工序
2	LPT	选择具有最长加工时间的工序
3	MWR	选择剩余加工时间最长的工序
4	LWR	选择剩余加工时间最短的工序



基于优先规则的表达

- 染色体解码说明：

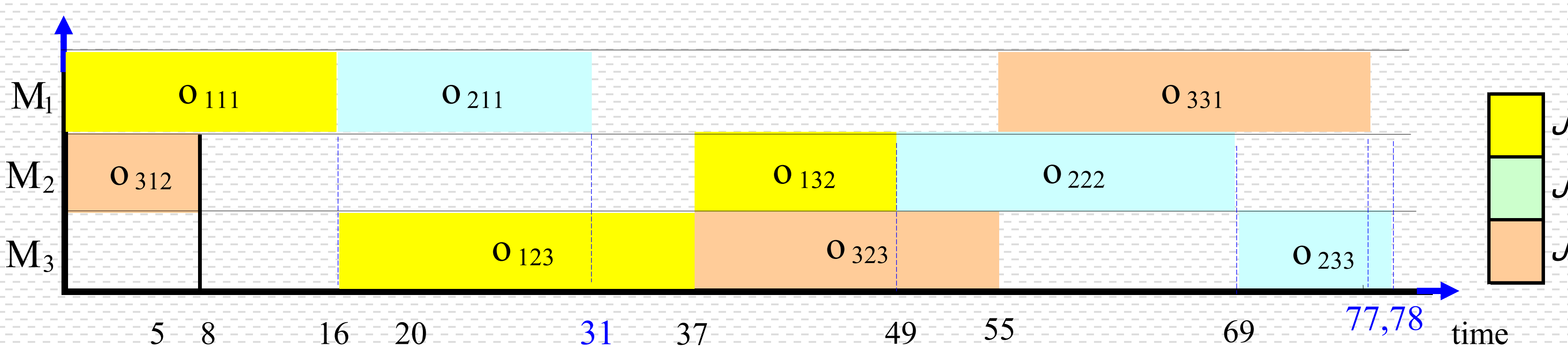
v =								
1	2	2	1	4	4	2	1	3

	p_{ikj}			M_j		
Operations	1	2	3	1	2	3
J_1	16	21	12	M_1	M_3	M_2
J_2	15	20	9	M_1	M_2	M_3
J_3	8	18	22	M_2	M_3	M_1

- 以此类推，
- 最后得到一个完全调度

No	规则	描述
1	SPT	选择具有最短加工时间的工序
2	LPT	选择具有最长加工时间的工序
3	MWR	选择剩余加工时间最长的工序
4	LWR	选择剩余加工时间最短的工序

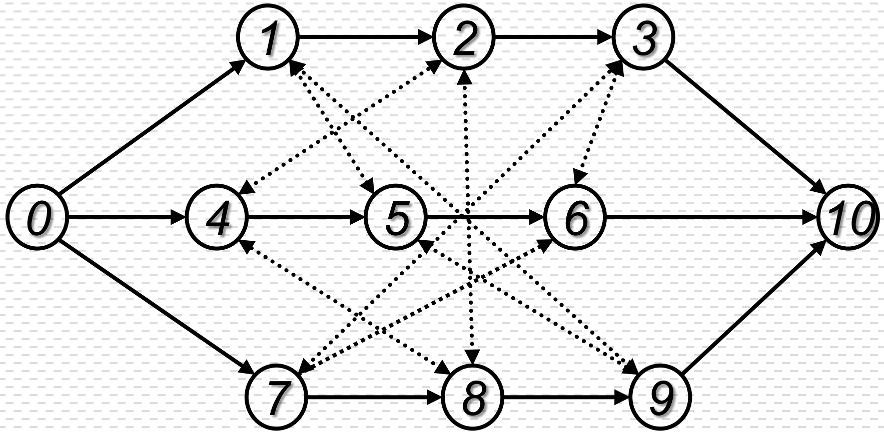
Makespan : 78 (units)



基于非连接图表达

- Tamaki和Nishikawa提出了一种基于非连接图的表达法，
- 也可视为一种基于工件对关系的表达法。
- 一个染色体包含了一个对应 E 中非连接边有序列表的二进制串，其中 e_{ij} 代表节点 i 与 j 之间的非连接边，其定义为:

$$e_{ij}=\begin{cases}1, & \text{标定非连接边的方向是从节点 } j \text{ 到节点 } i \\ 0, & \text{标定非连接边的方向是从节点 } i \text{ 到节点 } j\end{cases}$$



染色体编码的例子: v =

e_{15}	e_{19}	e_{59}	e_{24}	e_{28}	e_{48}	e_{36}	e_{37}	e_{67}
0	0	1	1	0	0	0	1	1

机器调度问题

(Machine Scheduling Problems)

问题概述

- 机器调度问题既有丰富的研究内容，同时又是一个在机械制造、逻辑、计算机结构等方面有广泛应用前景的研究领域。
- 机器调度问题包括的主要方面有：
  - 机器配置
    - 单机问题
    - 多机问题
  - 工件特征（不限于）
    - 工件之间的先后关系
    - 工件下达时间
    - 工件交货期
    - 工件优先权
  - 目标函数
    - 单目标问题
    - 多目标问题
- 机器调度问题属于组合优化问题，其复杂性常常通过转换为已知复杂性的其他问题来确定

单机调度问题

- 基本单机调度问题的假设可描述为：
  - 一组由 n 个独立的单道工序工件在时间 0 等待加工；
  - 工件的装设时间与工件的顺序无关，可以包括在加工时间内；
  - 工件是可知的；
  - 机器连续可用，在工件等待加工时机器不能闲置；
  - 不允许工件预定机器；
- 问题的目标：
  - 确定工件的加工顺序，使得一些性能的度量最优。

基本定义

- $(j_1, j_1, \ldots j_n)$ 表示工件；
- p_i表示工件 j_i 的加工时间；
- d_i表示工件 j_i 的交货期；
- w_i表示权重；
- c_i表示完工时间；
- r_i表示准备时间；
- 流程时间（flowtime） $F_i = c_i - r_i$，即工件在系统中的总逗留时间。用于度量系统对各个服务需求的反应，表示工件在到达和离开系统的时间长度。
- 延迟时间（lateness） $L_i = c_i - d_i$，即工件的完工时间超过其交货期的总时间。用于度量一个调度对于给定交货期的一致性。负值表示在交货期前完工；正值通常伴有惩罚和费用。
- 拖期时间（tardiness） $T_i = \max(0, L_i)$，表示工件j_i不满足交货期的延迟时间，如果满足交货期，其值为0。
- 令∏表示工件之间无闲置时间的可行调度的集合。对于一个给定的调度σ∈∏，令 $f(\sigma)$ 表示相应的目标函数值，且$r_i = 0$。

各种问题

① 平均流程时间问题（mean flowtime problem）

$$\min_{\sigma\in\Pi} \quad f(\sigma) = \frac{1}{n}\sum_{i=1}^{n} F_i = \frac{1}{n}\sum_{i=1}^{n} c_i$$

- 可以用工件加工时间的非减顺序来调度工件，即：$p_{i1} \le p_{i2} \ldots \le p_{in}$。这就是著名的最短加工时间调度规则(SPT规则)。

② 加权流程时间问题（weighted flowtime problem）

$$\min_{\sigma\in\Pi} \quad f(\sigma) = \sum_{i=1}^{n} w_i F_i = \sum_{i=1}^{n} w_i c_i$$

- 可以用 p_i/w_i 的非减顺序来调度工件，即: $p_{i1}/w_{i1} \le p_{i2}/w_{i2} \ \ldots \le p_{in}/w_{in}$。这就是加权最短加工时间规则(WSPT)。

③ 平均拖期时间问题（mean tardiness problem）

$$\min_{\sigma\in\Pi} \quad f(\sigma) = \frac{1}{n}\sum_{i=1}^{n} T_i$$

各种问题

④ **最大流程时间问题（maximum flowtime problem），也称为制造周期(makespan)问题**

$$\min_{\sigma\in\Pi} f(\sigma) = \max_{1\le i\le n}\{F_i\} = \max_{1\le i\le n}\{c_i\}$$

⑤ **最大延迟时间问题（maximum lateness problem）**

$$\min_{\sigma\in\Pi} f(\sigma) = \max_{1\le i\le n}\{L_i\}$$

- 可以用最早交货期规则(**EDD**规则)来调度工件，即: $d_{i1} \le d_{i2} \ldots \le d_{in}$ 。

⑥ **最大拖期时间问题（maximum tardiness problem）**

$$\min_{\sigma\in\Pi} f(\sigma) = \max_{1\le i\le n}\{T_i\}$$

- 可以用**EDD**规则求解。

各种问题

⑦ 最少拖期工件数问题（minimum tardy job problem）

$$\min_{\sigma \in \Pi} f(\sigma) = \sum_{i=1}^{n} \delta(T_i) \ , \quad \delta(T_i) = \begin{cases} 1 & \text{if } T_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

- 以上讨论的所有指标都是工件完工时间的函数，所以有如下一般形式：

$$\min_{\sigma \in \Pi} f(\sigma) = f(c_1, c_2, \ldots, c_n)$$

- 它们都属于一类重要的性能度量—规则度量。
  - 如果满足如下条件，则一个性能度量是规则的：
    - 调度的目标是极小化 $f(\sigma)$；
    - $f(\sigma)$ 增加，当且仅当至少有一个完工时间增加

各种问题

⑧ 完工时间方差问题（completion time variance problem - CTV）

$$\min_{\sigma\in\Pi} f(\sigma) = \frac{1}{n}\sum_{j=1}^{n}(c_j-\bar{c})^2, \quad \bar{c} = \frac{1}{n}\sum_{j=1}^{n}c_j$$

- 在最优的CTV调度中，最长时间的工件首先加工
- Cheng和Cai已经证明CTV问题是**NP难题**，不存在求解CTV问题最优解的多项式或伪多项式算法。

⑨ 等待时间方差问题（waiting time variance problem - WTV）

$$\min_{\sigma\in\Pi} f(\sigma) = \frac{1}{n}\sum_{j=1}^{n}(w_j-\bar{w})^2, \quad \bar{w} = \frac{1}{n}\sum_{j=1}^{n}w_j$$

- 其中w_i表示工件j_i的等待时间：$w_i = c_i - p_i$
- 最优WTV调度是V型调度，也就是说最短加工时间工件的紧前和后继工件分别按LPT和SPT顺序调度。

各种问题

⑩ 完工时间绝对偏差和问题（total absolute differences in completion time- TADC）

$$\min_{\sigma\in\Pi} f(\sigma) = \sum_{i=1}^{n}\sum_{j=i}^{n} |c_j - c_i| = \sum_{i=1}^{n}(i-1)(n-i+1)p_{[i]}$$

- 其中[*i*]表示第 *i* 个加工的工件。
- 因为CTV问题的复杂性．Kanet提出了TADC作为方差的度量，并且提出了一个求解该问题的*O*(n*Log*n)算法。
- TADC和CTV问题的关键区别是前者为绝对偏差和后者为方差和。
- 通过匹配位置权重(*i*-1)(*n*-*i*+1)的非增顺序与加工时间 $p_{[i]}$ 的非减顺序来求解。

⑪ 等待时间绝对偏差和问题（total absolute differences in waiting time- TADW）

$$\min_{\sigma\in\Pi} f(\sigma) = \sum_{i=1}^{n}\sum_{j=i}^{n} |w_j - w_i| = \sum_{i=1}^{n} i(n-i)p_{[i]}$$

- TADW和WTV之间的关系类似于TADC和CTV之间的关系。
- 可通过匹配位置权重 *i*(*n*-*i*) 的非增顺序与加工时间 $p_{[i]}$的非减顺序来求解。

- 以上给出的这些度量都是非规则度量。
  - 非规则性能度量可以随工件完成时间的减少而增加。

提前/拖期(E/T)调度问题

问题的提出

- 调度模型中**Earliness and Tardiness**惩罚的研究是最近兴起的一个研究领域。
- 多年来，调度研究主要侧重于对工件完工时间的非减规则度量
  - 如平均流程时间，
  - 平均延迟时间，
  - 工件拖期率，
  - 以及及平均拖期时间这样的度量。
- 特别地，平均拖期时间准则尽管忽略了工件可能提前完工的结果，但它仍是度量交货期一致性的标准方式。
- 然而，人们对准时生产(JIT)的兴趣改变了这种状态。
  - 准时生产（**Just-In-Time**）源于提前和拖期都不应该鼓励的思想；
  - 提前和拖期惩罚概念的引进，已经成为调度领域中一个新的迅速发展的研究分支。
- 由于提前和拖期惩罚的使用，引出了一个非规则性能度量，导致了新的求解方法的研究。

E/T模型描述

- 令E_i和 T_i分别代表工件j_i 的提前和拖期时间，并定义为：

$$
\begin{aligned}
E_i &= \max\{0, d_i - c_i\} = (d_i - c_i)^+ \\
T_i &= \max\{0, c_i - d_i\} = (c_i - d_i)^+
\end{aligned}
$$

- 令$\alpha_i>0$, $\beta_i>0$ 是与每个工件相对应的单位提前和单位拖期惩罚。
- 对于一个给定的调度$\sigma\in\prod$，基本E/T调度问题可以写成：

$$
\begin{aligned}
\min_{\sigma\in\Pi} \quad f(\sigma) &= \sum_{i=1}^{n}[\alpha_i(d_i - c_i)^+ + \beta_i(c_i - d_i)^+] \\
&= \sum_{i=1}^{n}(\alpha_i E_i + \beta_i T_i)
\end{aligned}
$$

  - 惩罚可以采用不同的度量方法，可以对工件给出相同或不同的惩罚
  - 交货期可以是给定的，或者与工件顺序同时优化；
  - 最简单的模型是考虑所有工件有共同的交货期，更一般的模型是允许有不同的交货期。
  - 即使最简单模型的优化问题也是一个NP难题。

目标函数的几种形式

① 绝对偏差问题（absolute deviation problem）：

- E/T问题中的一个重要特例是工件完工时间与公共交货期的绝对偏差和的最小化问题，目标函数可以写成：

$$\min_{\sigma \in \Pi} \quad f(\sigma) = \sum_{i=1}^{n} | c_i - d |$$

- 其中$d_i=d$。模型中所有工件有相同的E/T惩罚系数
- 希望构造一个调度使得交货期 d 处于中间工件。
- 如果交货期太紧，则不可能在交货期前放置足够的工件，因此对于结定的问题，公共交货期太紧，就会成为限制问题，否则就是非限制问题。
- 非限制问题存在最优解，且最优解满足如下性质。
  - 调度中没有可插入的空闲时间；
  - 最优调度是V型调度；
  - 一个工件恰好在交货期完工；
  - 在最优调度中，第k个工件在时间d完工。其中k是大于或等于$n/2$的最小整数

目标函数的几种形式

② 加权绝对偏差问题（weighted absolute deviation problem）：

- **Hall等研究的加权E/T问题定义为：**

$$\min_{\sigma\in\Pi} \quad f(\sigma) = \sum_{i=1}^{n} w_i \mid c_i - d \mid$$

- 其中 w_i 是对应第 i 个工件的整数权重
- 模型的基本思想是不同的工件应该得到系数不同的惩罚。
- 从一般意义上来说，加权绝对偏差问题是NP完全问题。
- 令提前工件集 $E=\{j_i | c_i \le d\}$，拖期工件集 $T=\{j_i | c_i > d\}$，最优性条件如下
  - 存在一个最优解，其中某些工件恰好在交货期 d 完工；
  - 最优调度是V型调度，即：E中的工件按 w_i/p_i 非减排列，T中的工件按w_i/p_i非增排列；
  - 给定一个最优调度 σ^*，有：

$$\sum_{i\in E} w_i \ge \sum_{i\in T} w_i \quad , \quad \sum_{i\in E} p_i \ge \sum_{i\in T} p_i$$

目标函数的几种形式

③ 平方偏差问题（**squared deviation problem**）：

- 在某些情况下，不希望出现交货期的太大偏差，这时用与公共交货期的平方偏差作为性能度量可能更为合适：

$$\min_{\sigma\in\Pi} f(\sigma) = \sum_{i=1}^{n}(c_i - d)^2 = \sum_{i=1}^{n}(E_i^2 + T_i^2)$$

- 这是总绝对偏差的二次形式。
- **Elion**和**Chowdhury**曾经证明其最优解是**V**型调度
- **Bagchi, Chang and Sullivan**还研究了不同**E/T**惩罚的情况，即：

$$\min_{\sigma\in\Pi} f(\sigma) = \sum_{i=1}^{n}(\alpha E_i^2 + \beta T_i^2)$$

并行机调度问题

- 多机调度问题（Multiple machine scheduling）
  - 是研究机器对于一组工件的加工顺序，以确保所有工件在合理的总加工时间内加工完成。
  - 主要涉及以下三个问题：
    - 哪台机器分配给哪些工件?
    - 如何确定工件适当的加工顺序?
    - 如何评价调度的合理性?
  - 换句话说，多机调度理论主要关心的是如何提供一个机器与工件的完美匹配，或者是近似完美的匹配，然后确定每台机器上工件的加工顺序以达到预先规定的一些目标。
- 并行机器调度问题（Parallel machine scheduling）
  - 可以视为一类由多机调度问题松弛而得到的问题。
  - 在并行机器系统中，所有机器都相同并且每个工件可以在任何一个空闲机器上加工；
  - 每个加工完的工件释放一台机器并且离开系统。

工件描述及最优性准则

- 一个工件可从以下方面来描述：
  - 工件加工时间：单位加工时间或者各种加工时间；
  - 交货期要求：无交货期，公共交货期或者一般交货期；
  - 优先的工件：是否允许优先；
  - 先后约束：工件是独立的、非独立的或者有树形先后约束；
  - 工件准备时间：所有工件有相同准备时间或者任意准备时间。
- 最优性准则可以分为三个不同组：
  - 基于完成时间的度量；
  - 基于交货期的度量；
  - 基于闲置惩罚的度量。

假设条件

- 并行机器调度问题一般作如下假设：
  - 每个工件只有一道工序；
  - 一个工件同时只能在一台机器上加工；
  - 任何机器可以加工任意工件；
  - 一台机器同时只能加工一个工件；
  - 机器在调度期间内不会中断，且始终可用；
  - 机器加工时间与调度无关；
  - 机器装设时间可忽略；
  - 机器之间的运输时间可忽略；
  - 允许加工中的库存存在，其费用可以忽略；
  - 工件数可预先指定；
  - 机器数可预先指定；
  - 对于所有的 i 和 k，机器 m_k 上加工工件 j_i 的时间是预先给定的；
  - 对所有的 i，工件 j_i 的准备时间已知。

目标函数的几种形式

① 最小流程时间问题（minimum makespan problem）

- 考虑同时可用且相互独立的非优先工件的集合（n个工件）在完全一致、互不关联的并行机器集合(m台机器)($m<n$)上加工的调度问题。
- 最小流程时间问题定义为:

$$\min_{\sigma\in\Pi} f(\sigma) = \max\{c_j | j = 1, \ldots, n\}$$

- Garey和Johnson已经证明该问题在机器数不定的情况下从严格意义上来说是NP难题。
- 当机器数预先给出时可以在伪多项式时间内求解，因此只在一般的意义上是NP难题。

目标函数的几种形式

② 最小化加权流程时间问题（minimum weighted flowtime problem）

- 令w_i，r_i分别表示工件j_i的权重和准备时间。
- 问题是寻找一个最优的无预先指定的工件调度，使得加权流程时间最小：

$$\min_{\sigma \in \Pi} \quad f(\sigma) = \sum_{j=1}^{n} w_j (c_j - r_j)$$

- **Bruno**等已经证明，即使是双机系统，在不同工件具有不同权重的情况下，也是一个**NP**难题。

目标函数的几种形式

③ 最小化最大加权绝对延迟时间问题

（minmax weighted absolute lateness problem）

- 给定一个非限制的公共交货期 ***d***，

$$d \geq \sum_{j=1}^{n} p_j$$

- 问题是寻找一个最优的无优先工件的调度，使得最大加权绝对延迟时间最小，即：

$$\min_{\sigma \in \Pi} \quad f(\sigma) \quad = \quad \max\{w_j|c_j - d|; j = 1, \ldots, n\}$$

- **Li**和**Cheng**的研究表明，即使对于一个单机系统，该问题也是一个**NP**难题，并且提出了一个启发式方法来求解这一问题。

目标函数的几种形式

④ 最小化加权绝对延迟时间和问题

（ **minsum weighted absolute lateness problem** ）

- 该问题是寻找一个最优的工件调度，使得加权绝对延迟时间和最小，即：

$$\min_{\sigma\in\Pi} \quad f(\sigma) \;=\; \sum_{j=1}^{n} w_j |c_j - d|$$

- 最小和问题试图极小化工件完工时间关于交货期的加权绝对偏差和，
- 而最小最大问题是最小化工件完工时间关于交货期的最大加权绝对偏差。

End