布局设计问题

(Layout Design Problem)

东北大学 系统工程研究所

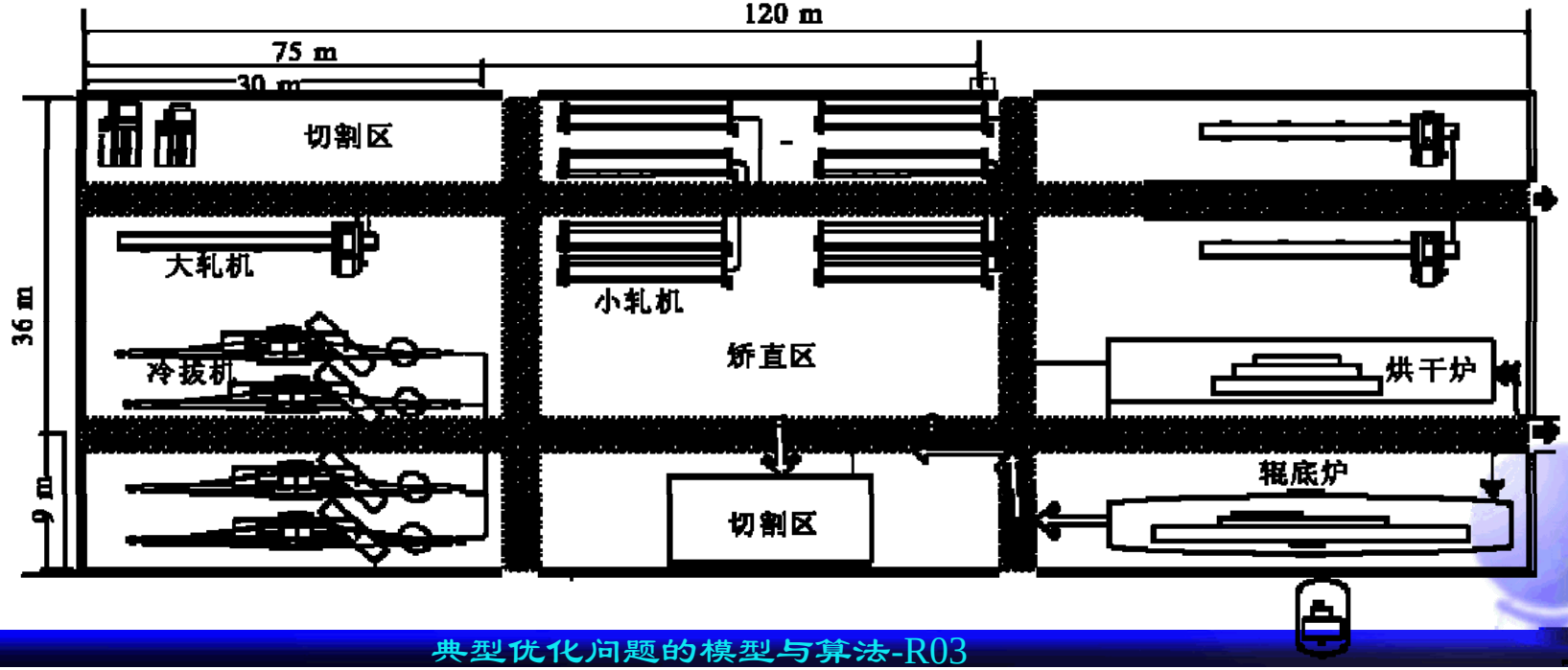
2011.09

设备布局设计问题

- 设备布局设计一直是跨学科的研究课题。
  - 设备布局的发展过程中既包含艺术因素又包含科学因素。
  - 最近这一课题得到运筹学和管理科学工作者的极大关注。
- 制造系统物理布局优化设计，是在系统设计初期必须解决的重要问题之一。
  - 在制造业中，总经营费用的20%～50%是物料搬用费用，而优良的设备布局可以使这个费用减少10%～30%；
  - 此外，在工厂车间的生产活动中，物料真正处于加工检验状态的时间只占生产周期的10%左右，而其余的大概90%的时间都处于停滞和搬用状态，而优良的设备布局能加快物料处理效率，减少在制品停留，提高企业生产率，减少产品生产周期。
  - 布局决策的好处可以在系统实现和系统运行过程中体现出来。
  - 好的布局方案为有效地利用系统提供了必要的基础。但是获得这样的解决方案，却是一项复杂的任务。
- 在制造环境中，布局设计是指：
  - 在确定的区域内最适宜地安排物理设备，如车间或机器。
  - 通常设计准则是最小化物料储运费用。
  - 此外，还要综合考虑布局的美观性，操作的舒适性以及生产的安全性等因素。

例如：钢管生产车间设施布局设计

- 一个钢管生产车间包括很多设备，例如：辊底炉，润滑烘干炉，高速冷轧管生产线，精密冷轧管机生产线，高精密链式冷拔拉管机组生产线，维修车床，精密矫直机组，切割机等。
- 钢管的生产是工艺流转性生产活动，合理的布局应在满足生产工艺的前提下，使物料 搬运距离最短（同时也要考虑安全事故低、搬运冲突少等问题）。
- 生产工艺决定了产品在不同设备之间的运输顺序，机器的布局影响的是运输距离，乘以单位运输成本，就构成运输费用。

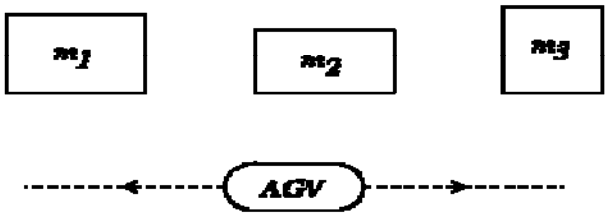


机器布局问题

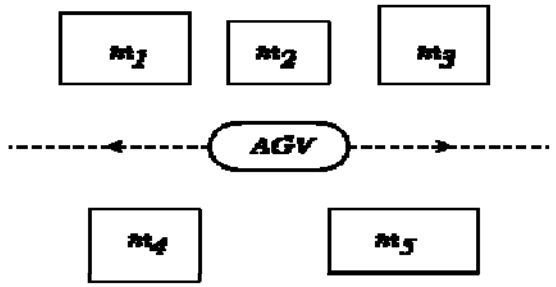
- 柔性机器系统设计
  - 柔性机器系统的设计包括机器和工作站的布局。
  - 柔性机器系统中机器的布局通常是由所采用的物料储运设备类型决定的。
- 物料储运设备：
  - 物料储运机器人；
  - 自动引导车 (AGV)；
  - 门式机器人。
- 机器布局的四种基本类型：
  - 线形单行；
  - 线形双行；
  - 环形单行；
  - 多行。

机器布局问题

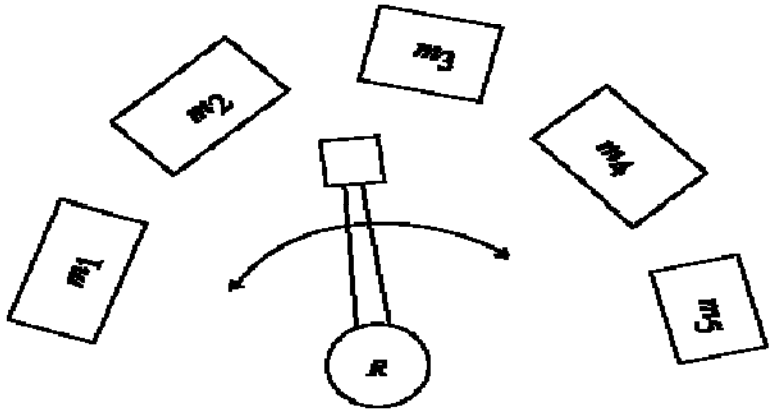
- 机器布局的四种基本类型：
  - 由一个 AGV 服务的机器趋于沿着直线安排 (a,b);
  - 而机器人的可达到范围约束使机器排成环形或簇状 (c,d)。



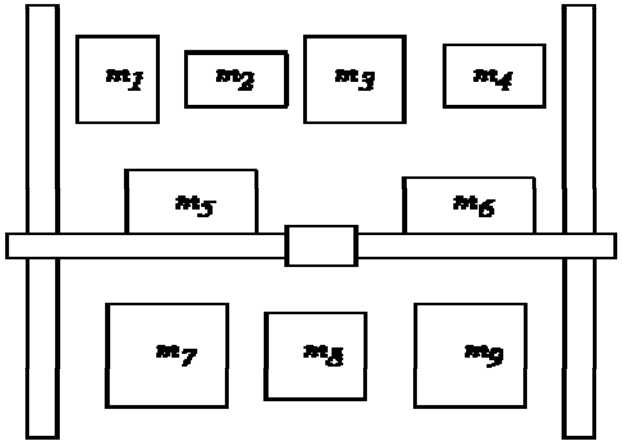
(a) linear single-row



(b) linear double-row



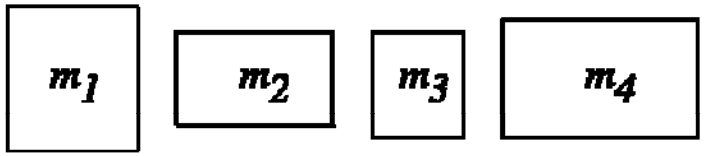
(c) circular single-row



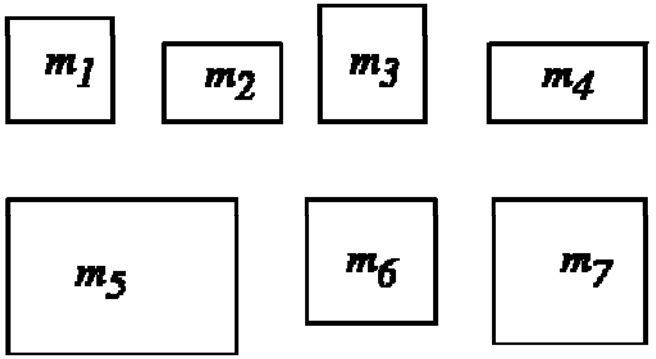
(d) multiple-row

机器布局问题

- 通常考虑两种问题的建模
  - 单行模式
  - 多行模式
- 因为四种布局类型中
  - 环形单行和线形单行可以看作单行布局模式;
  - 线形双行布局是多行布局的一个特例。
  - 在单行模式中，机器安排在一行内;
  - 而在多行模式中，机器成线形地安排在两行或多行内。



(a) single-row layout



(b) multiple-row layout

特点

- 具有组合的特性
- 通常作为二次指派问题建模
  - 需预先说明每个点的位置，然后为每个点指派设备以使全部的物料储运费用最小。
- 机器的大小通常不同
  - 如果两台机器间的距离由各自的尺寸和它们之间需要的间距决定，则位置之间的距离是不等的并且是不可预先确定的
- 布局问题在实践中应用广泛，但很难求解
  - 对于复杂工程系统的布局问题，一般会涉及到复杂几何形状，涉及到大量的非线性数值约束和大量的不可形式化的约束。
  - 同时制造系统布局设计也不是一个孤立的问题，它与系统其它设计（如工艺设计、产品设计、物流设计）紧密相关。
  - 另一方面，设备布局也会影响到物流系统的运行和控制。布局设计和物流系统的交叉设计能最大限度地减少、消除物流环节，使制造系统中的物流合理化，从而达到加快物料处理时间、降低物料运输成本和均衡设备负荷的目的。

布局设计问题的应用

- 制造系统的设备布局
- 建筑平面布局
- 集成电路/芯片布局
- 报纸/网页版面设计
- 造船业板材切割中拼装问题
- 机械行业冲压下料问题
- 玻璃排版优化问题
- 集装箱摆放布局
- 航天器的布局设计
- 卫星有效载荷布局设计
- 服装企业生产系统设备布局
- 林区储木加工厂生产工艺布局
- ……

“平面”、“规则物体”的布局优化问题

“平面”、“不规则物体”的布局优化问题，不仅有“布局设计问题”，同时还有“背包”、“TSP问题”

“立体”、“规则物体”的布局优化问题

“立体”、“不规则物体”的布局优化问题

求解方法

- 一般求解方法
  - 由于设备布局问题的组合特性，启发式技术是解决实际布局问题的最有效方法。

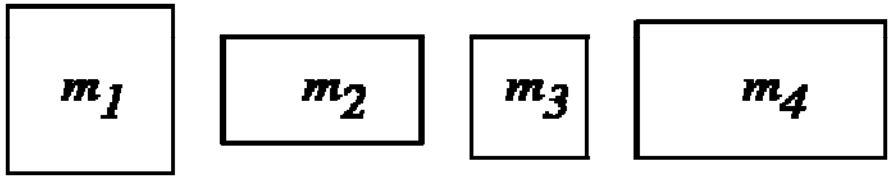
- GA求解
  - Tate和Smith将遗传算法用于有形状约束的不等区域设备布局问题。
  - Cohoon 等提出解决车间布局设计问题的分布式的遗传算法。
  - Tam介绍了将遗传算法用于设备布局问题的经验。
  - Cheng, Gen 的研究
    - Genetic search for facility layout design under interflows uncertainty
    - Genetic algorithms for multi-row machine layout problem

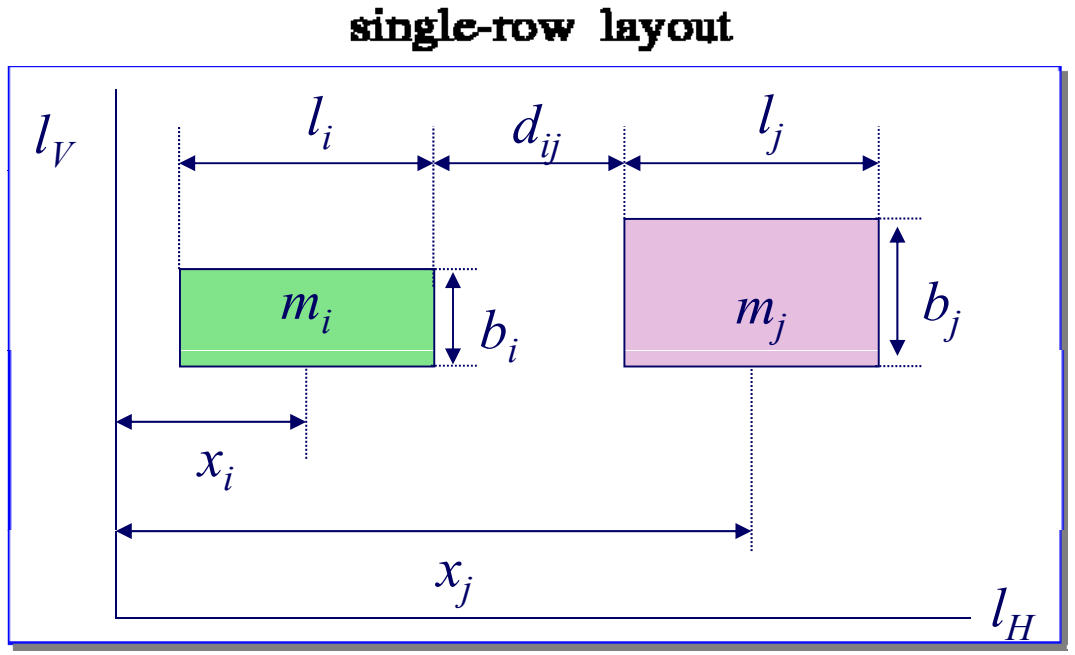
单行机器布局问题

(Single-row Machine Layout Problem)

单行机器布局问题

- 为了对s-MLP 建模，作如下假设
  - 机器是矩形的；
  - 机器方位已知；例如，所有的机器都是纵向放置的。





数学模型

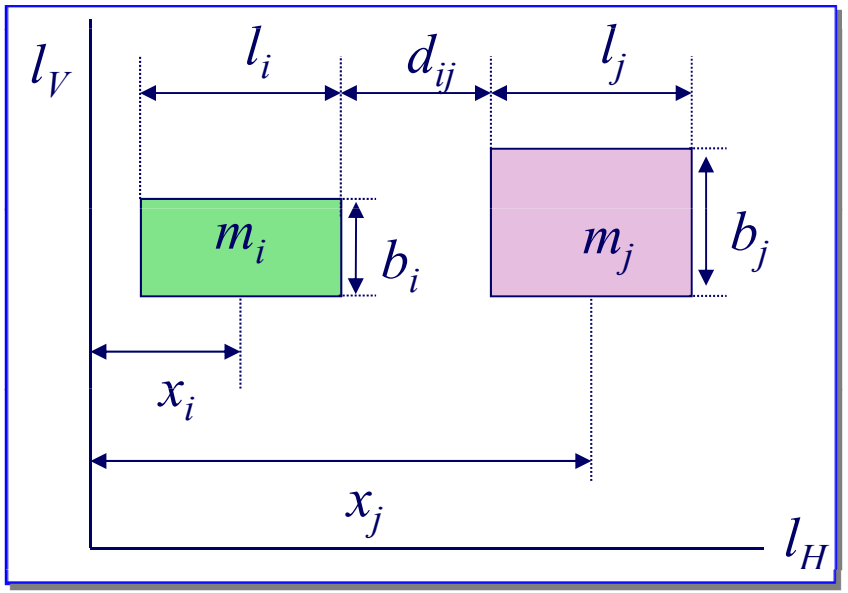
- **Heragu 和 Kusiak 提出了在目标函数和约束中含有绝对值的机器布局问题模型。**

$$\min \quad z(x)=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}c_{ij}f_{ij}\left|x_i-x_j\right|$$

$$\text{s. t.} \quad \left|x_i-x_j\right|\geq\frac{1}{2}(l_i+l_j)+d_{ij},$$

$$i=1,\ldots,n-1,\quad j=i+1,\ldots,n$$

$$x_i\geq 0,\qquad i=1,\ldots,n$$

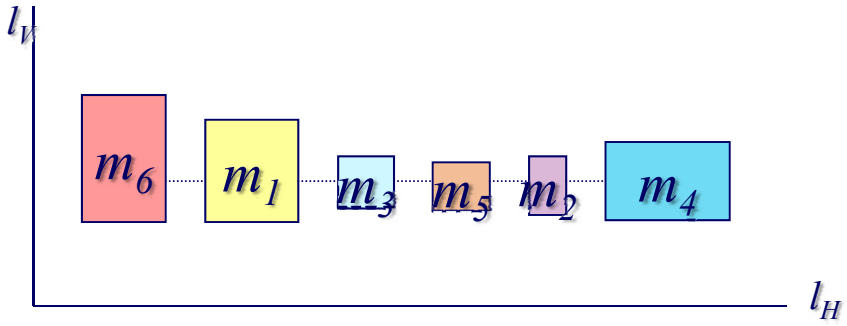


- 其中：
  - i 机器的索引号
  - f_{ij} 机器 i 和 j 之间的访问频率
  - c_{ij} 机器 i 和 j 之间的访问时每单位距离的储运费用
  - l_i 机器 i 的长度
  - d_{ij} 机器 i 和 j 之间的最小间距
  - b_i 机器 i 的宽度
  - x_i 机器 i 的中心线相对垂直参考线 l_V 的距离。

GA求解

- 单行机器布局是一种非常特殊的情况，这一问题可看作是机器的排序问题，因而它可以由两步完成:
  - 第一步：机器排序;
  - 第二步：根据机器的排序和几何需求建立实际布局。
- GA 对于处理这样的排列问题是非常有效的。
- 染色体表达
  - 对于单行情况，机器的换位表达是对机器布局进行编码以获得染色体的一种直接且有效的方法。
  - 通常，一个 ***n*** 台机器布局问题的染色体编码如下：

排列顺序	1	2	3	4	5	6
机器编号	6	1	3	5	2	4



- 遗传算子
  - 对于这样的编码，存在许多可用的遗传算子，如：部分映射交叉(PMX)、反转变异等。

GA求解

- 评价函数
  - 通过染色体的编码，我们可以得到机器的排列。
  - 根据机器的排序和几何尺寸需求，我们可以计算所有机器的 ***x*** 轴坐标(机器的位置或机器的实际布局):

$$[\ x_1^k \quad x_2^k \quad \ldots \quad x_n^k\ ]$$

  - 通过目标函数，可以获得染色体 v_k 的总费用

$$f_k = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} f_{ij} \left| x_i^k - x_j^k \right|$$

  - 由于布局设计问题是最小化问题，可以采用如下的评估函数转化为适值。

$$eval(v_k) = \frac{1}{f_k}$$

多行机器布局问题

(Multi-row Machine Layout Problem)

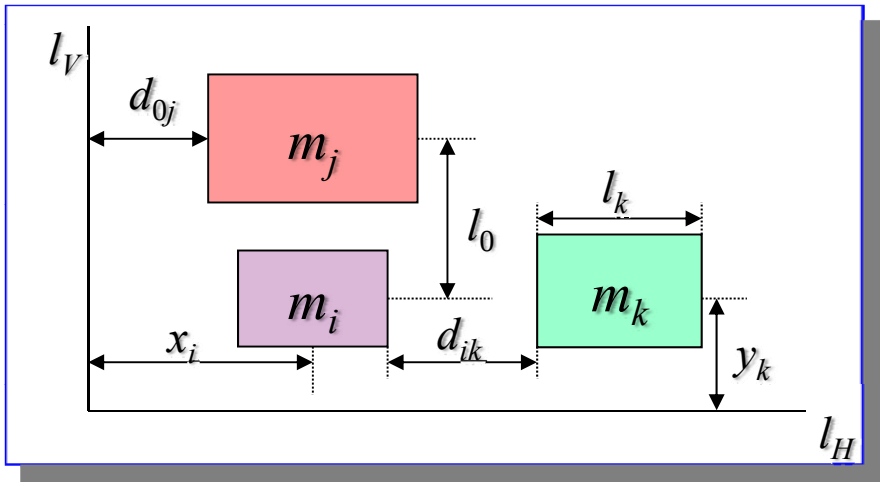
多行机器布局问题

- 解决设备布局问题的许多传统方法是将设备分配给事先确定的位置，这是离散优化方法。但在机器布局问题中，位置不能事先确定。
- 然而，在许多实际问题中，机器可以安排在事先定义好的行内，因为许多情况下，行的分隔可根据物料储运系统的特点事先确定，所以这一问题在一维上可以看作是离散的，在另一维上看作是连续的。

- **m-MLP** 问题由两个任务构成：
  - 分配机器到行 (确定 *y* 坐标)
  - 在每一行中寻找最佳的机器位置 (确定 *x* 坐标)
- 第一个任务是一个沿 *y* 轴的组合优化问题
- 第二个任务是一个沿 *x* 轴的连续优化问题，尽管是一个**s-MLP**问题，但由于行之间储运费用的存在，每一行的最优解对整个问题来讲未必是最好的
- 因此我们不能简单地把问题看作是几个 **s-MLP**。
- 问题的关键是：如何有效处理连续与离散优化的组合问题。

数学模型

- 索引号：
  - i 机器的索引号
- 参数：
  - f_{ij} 机器 i 和 j 之间的访问频率
  - c_{ij} 机器 i 和 j 之间的访问时每单位距离的储运费用
  - l_i 机器 i 的长度
  - l_0 两个相邻行之间的分隔距离
  - d_{ij} 机器 i 和 j 之间的最小间距
  - b_i 机器 i 的宽度



- 决策变量：
  - x_i 机器 i 的中心线相对垂直参考线 l_V 的距离
  - y_i 机器 i 的中心线相对水平参考线 l_H 的距离

$$z_{ik} = \begin{cases} 1, & \text{如果机器 } i \text{ 被分配到第 } k \text{ 行} \\ 0, & \text{其它} \end{cases}$$

数学模型

- 具有不等区域的多行机器布局问题可以描述为混和整数规划问题：

$$\min \quad \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij} f_{ij} \left(\left| x_i - x_j \right| + \left| y_i - y_j \right| \right)$$

$$\text{s. t.} \quad \left| x_i - x_j \right| z_{ik} z_{jk} \geq \left[\frac{1}{2}(l_i + l_j) + d_{ij} \right] z_{ik} z_{jk}, \; i, j = 1, \ldots, n$$

$$y_i = \sum_{k=1}^{m} l_0 (k-1) z_{ik}, \qquad i = 1, \ldots, n$$

$$\sum_{k=1}^{m} z_{ik} = 1, \qquad i = 1, \ldots, n$$

$$\sum_{i=1}^{n} z_{ik} < n, \qquad k = 1, \ldots, m$$

$$x_i, y_i \geq 0, \qquad i = 1, \ldots, n$$

$$z_{ik} = 0, 1, \qquad i = 1, \ldots, n, \; k = 1, \ldots, m$$

- 目标是极小化机器间作必要次数访问时的总费用，
- 约束(1)确保同一行上任何两台机器均不重叠，
- 约束(2)表明模型中的变量 y_i 是不必要的，因它可以由 z_{ik} 确定，但这一约束有助于理解模型。
- 约束(3、4)确保一台机器只放在一行上。

GA求解

- 染色体表达
  - 对于多行机器布局问题，**Cheng，Gen** 提出了一种表达模式。
  - 染色体由三个部分构成，包括：
    - 分隔符：　s　表示序列被分割成两部分的位置
    - 机器符号：　m_{i_j}　代表第 j 个位置的机器
    - 净间距：　Δ_{i_j}　表示机器 $m_{i_{j-1}}$ 和 m_{i_j} 之间的净间距
  - 对于 ***n*** 台机器和两行的情况，表达方式为：

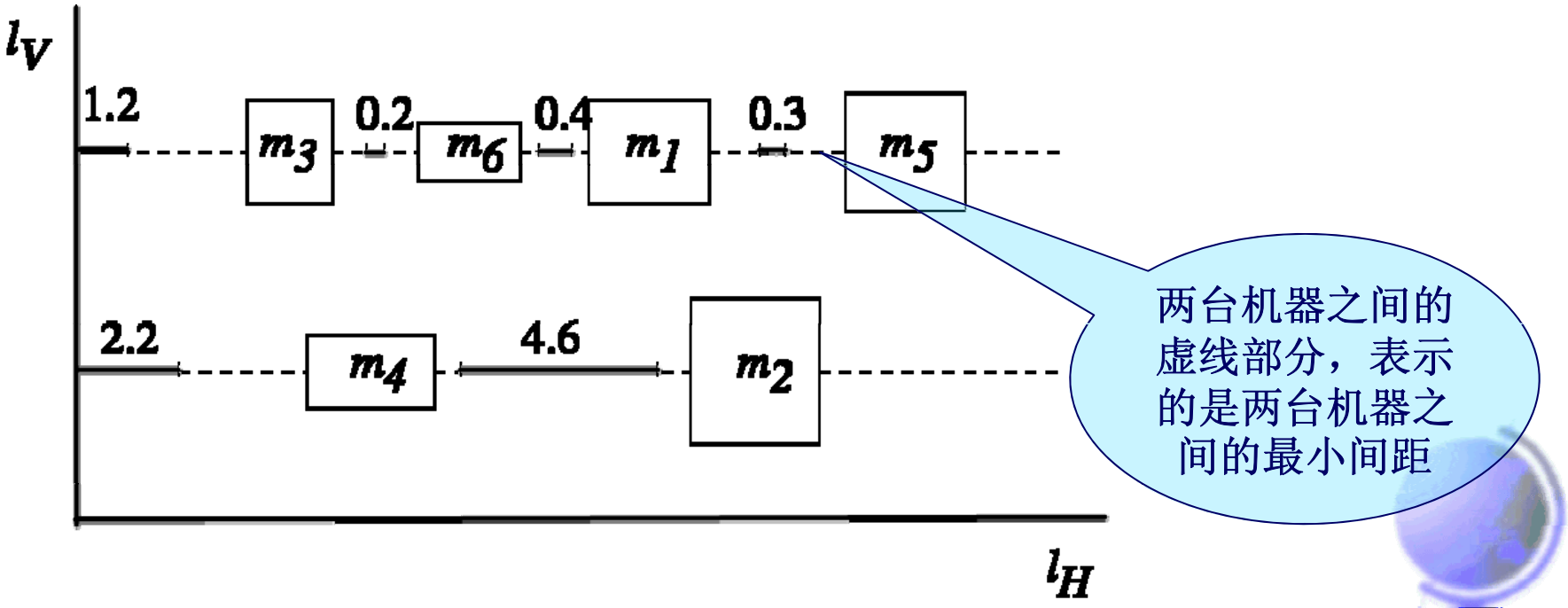
$$[s,\{m_{i_1},m_{i_2},\ldots,m_{i_n}\},\{\Delta_{i_1},\Delta_{i_2},\ldots,\Delta_{i_n}\}]$$

GA求解

- 编码与布局
  - 例如：6台机器，分2行的情况，假设染色体表达如下：

[4, {3, 6, 1, 5, 4, 2}, {1.2, 0.2, 0.4, 0.3, 2.2, 4.6}]

  - 根据这一编码，实际布局如图所示。



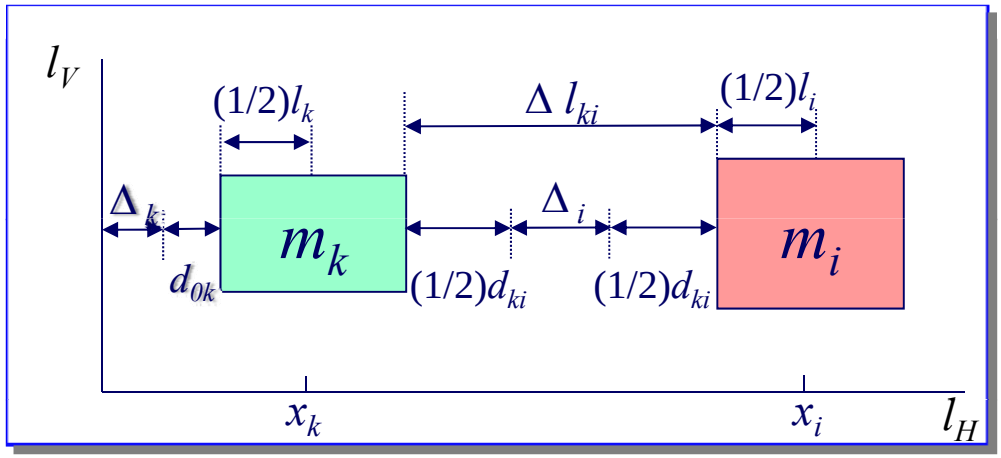
GA求解

- ***x*** 轴坐标的计算
  - 假设机器 m_k 和机器 m_i 排在同一行内，***x*** 轴坐标可根据净间距序列和机器排列序列唯一确定。

$$\Delta l_{ki} = \Delta_i + d_{ki}$$

$$x_i = x_k + d_{ki} + \Delta_i + \frac{1}{2}(l_i + l_k)$$

$$x_k = d_{0k} + \Delta_k + \frac{1}{2} l_k$$



Δ_i 机器 m_i 的净间距

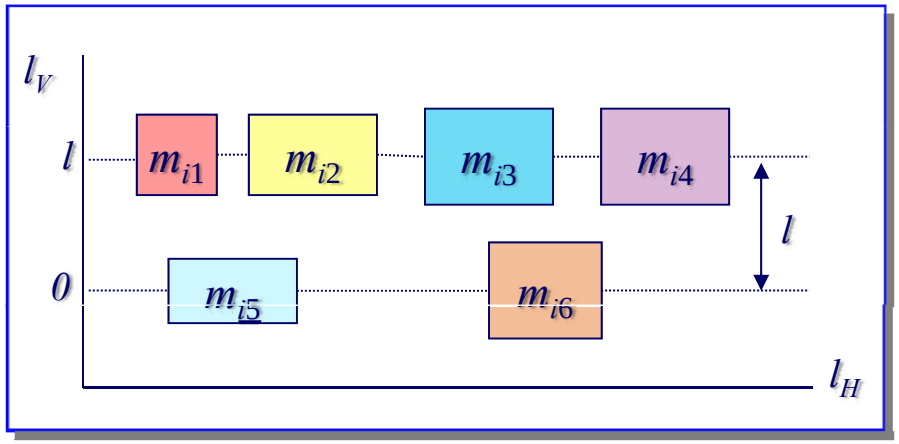
Δl_{ki} 两台机器间的间距

d_{ki} 机器 k and i 之间的最小间距

  - 由于净间距将会带来机器间访问总费用的增加，因此人们常认为，净间距为零是最好的。对于单行情况，这一结论是成立的，但是，对于多行情况，由于存在行间访问费用，这一结论是不成立的。

GA求解

- *y* 轴坐标的计算
  - *y* 轴坐标可基于行间距确定，而行间距可根据物料储运系统特点事先确定。
  - 考虑两行情况，令 *l* 代表两行间的距离。
    - 假设第一行的位置为 0, *y* 轴坐标计算如下：



$$y_i = \begin{cases} 0, & \text{if } m_i \text{ is in the first row} \\ l, & \text{if } m_i \text{ is in the second row} \end{cases}$$

GA求解

- 表达方法的优势（针对以*x*,*y*直接作为编码的表达方式）
  - 首先，这一编码技术可以显著减少进化过程中不合理后代的产生。
    - 由于 *x* 轴坐标由净间距间接表示，横向机器重叠将不会发生。
    - 用分隔符序列和机器排列序列，表达行间的机器分配，可以替代对 *y* 轴值的表达，纵向机器重叠将不会发生
  - 其次，采用这一编码机制，可以分三部分处理连续与离散优化的组合问题
    - 分配机器到行;
    - 在行内对机器排序;
    - 细微调整每台机器的位置(*x* 坐标)
  - 最后，因为采用序列表达机器排列，所以可以采用一些用于表达排序的传统的遗传算子来处理这一问题。
  - 可见，上述表达方法与问题的本质相吻合，并使遗传算法非常有效。

GA求解

- 初始化
  - 可以采用随机方法产生初始染色体，若不可行则抛弃，直至满足种群规模要求。
  - 产生过程分:分隔符、机器排列序列、净间距序列三部分进行。
- 初始化－分隔符
  - 对于 ***n*** 台机器、两行的情况，分隔符只是开区间 **(1, *n*)** 上的任意整数，可以随机产生。
- 初始化－机器换位
  - 采用随机的方式产生机器的顺序（产生方法有多种）
- 初始化－可行性检查
  - 对机器换位中由分隔符决定的任意一段，假设机器序列如下:

$$[\, m_1, m_2, \ldots, m_K \,]$$

  - 令 ***L*** 是工作区域的上限宽度，***S*** 表示所需要的必要空间，若机器序列满足下式，则是可行的，否则是不可行的。

$$S = \sum_{i=1}^{K} l_i + \sum_{i=1}^{K-1} d_{i,i+1} + d_{01} + d_{K0} \le L$$

GA求解

- 初始化—净间距
  - 产生净间距序列的过程实质上是一个随机方法。由于存在允许空间约束，所以净间距应该在允许区域内产生。
  - 对机器换位中由分隔符决定的任意一段，假设机器序列如下：

    $[\, m_1, m_2, \ldots, m_K \,]$
  - 令 L' 是可用空间，相应段的初始可用空间计算如下：

$$L' = L - \left(\sum_{i=1}^{K} l_i + \sum_{i=1}^{K-1} d_{i,i+1} + d_{01} + d_{K0}\right)$$

  - 令 Δ 表示相应段的净间距序列，初始化过程如下：

```
i ← 0;    Δ ← ϕ;
while (i < K) do
    在(0, L') 上，任取净间距 Δi;
    Δ ← Δ ∪ Δ i ; L' ← L' - Δ i ;
    i ← i + 1;
end
返回净间距序列 Δ ;
```

GA求解

- 交叉
  - 染色体由三部分构成，交叉过程也按三部分进行

- 交叉—新的分隔符：
  - 产生新的分隔符的过程很简单，主要由两步组成：
    - 确定分隔符的上下限；
    - 从上下限构成的区间内任选一个整数。
  - 例如，两个父代如下：
    - $p_1=[\{s^1\}, \{m_1^1, \ldots, m_n^1\}, \{\Delta_1^1, \ldots, \Delta_n^1\}]$
    - $p_2=[\{s^2\}, \{m_1^2, \ldots, m_n^2\}, \{\Delta_1^2, \ldots, \Delta_n^2\}]$
  - 上下限计算如下：
    - $s_U = \max\{s^1, s^2\}$，$s_L = \min\{s^1, s^2\}$
  - 所构成的闭区间为：
    - $[s_U, s_L]$
  - 新的分隔符是区间内随机产生的整数。

GA求解

- 交叉—新的机器换位：
  - PMX等各种针对换位表达的交叉算子

- 交叉—新的净间距序列：
  - 采用线性算术交叉来处理净间距序列。假设存在如下两个净间距序列：
    - $\{\Delta_1^1, \Delta_2^1, \cdots, \Delta_n^1\}$
    - $\{\Delta_1^2, \Delta_2^2, \cdots, \Delta_n^2\}$
  - 新的净间距计算如下：
    - $\Delta_i' = \alpha_1 \Delta_i^1 + \alpha_2 \Delta_i^2, \qquad i = 1, 2, \ldots, n,$
    - $\alpha_1, \alpha_2 \in (0, 1)$
    - 这里 α_1, α_2 是开区间 (0, 1) 内随机产生的实数
  - 特点：
    - 凸交叉需满足： $\alpha_1+\alpha_2 =1$，所提的线性交叉只要满足： $\alpha_1+\alpha_2 \leq 2$
    - 凸交叉产生的净间距将随着进化过程逐渐减小，搜索空间在很大程度上依赖于初始解空间。线性交叉的搜索空间可以大大增大，且独立于初始搜索空间。

GA求解

- 变异
  - 针对染色体的三个部分分别进行。
    - 分隔符部分随机产生
    - 机器换位部分，可采用基于换位表达的变异方法
    - 净间距部分，进行细微调整

- 变异—净间距细微调整
  - 变异采用邻域搜索技术对机器位置进行细微调整。
  - 假设给定染色体的净间距序列为：
    - $\{\Delta_1, \Delta_2, \cdots, \Delta_n\}$
  - 第 i 个非零基因 Δ_i 被选作变异
  - 令 r 是一个给定的整数，则可以得到 $2r$ 个净间距：

$$\Delta_{i'}^{1} = \frac{\Delta_i}{r}, \quad \Delta_{i'}^{j} = \Delta_{i'}^{j-1} + \frac{\Delta_i}{r}, \qquad j = 2,3,\ldots,2r$$

GA求解

- 变异—净间距细微调整
  - 这些净间距从 Δ_i/r 到 $2\Delta_i$ 变化，净间距序列如下：

$$\{\Delta_1^1, \Delta_2^1, \ldots, \Delta_{i'}^1, \ldots, \Delta_n^1\}$$

$$\{\Delta_1^2, \Delta_2^2, \ldots, \Delta_{i'}^2, \ldots, \Delta_n^2\}$$

$$\ldots$$

$$\{\Delta_1^n, \Delta_2^n, \ldots, \Delta_{i'}^k, \ldots, \Delta_n^n\}$$

  - 由上述净间距序列集合、给定染色体的分隔符及机器换位序列组成的染色体集合可看作为给定染色体的邻域。
  - 评估所有邻域，并将最好邻域作为后代。

GA求解

- 评估函数
  - 交叉、变异可能导致机器序列超出工作区域，可以采用惩罚策略，按以下方法确定：
    - 对于给定染色体 v_k，令L_k^1， L_k^2 分别是安排在第一行和第二行的机器需要的必要工作区域，令 $L_k^u = \max\{L_k^1, L_k^2\}$，则惩罚系数可计算如下：

$$\lambda_k = \begin{cases} 0, & \text{if } L_k^u - L \le 0 \\ L_k^u - L, & \text{otherwise} \end{cases}$$

  - 染色体 v_k 的适值函数可以表示为：

$$eval(v_k) = \frac{1}{f_k + \lambda_k P}, \quad k = 1, 2, \ldots, popSize$$

  - 其中，P 是正的大数惩罚值；f_k 是总费用，由下式确定。

$$f_k = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} f_{ij} \left(\left| x_i^k - x_j^k \right| + \left| y_i^k - y_j^k \right| \right)$$

  - x^k，y^k 由染色体解码得到。

End