

排序问题的网络分析方法

洪玉振

(河海大学国际工商学院, 江苏 南京 210098)

摘要 对应于一般单件车间排序问题, 构造了一种由节点、最短路径和相邻路径组成的隙网络. 通过网络分析, 探讨了求解这一最复杂的排序问题的局部最优解问题. 与启发式方法相比, 该方法为优化方法; 与分支定界法和整数规划法相比, 该方法是一种有效算法, 即随着问题规模的增大, 它具有多项式时间复杂性.

关键词 单件车间; 隙网络; 极小点; 多项式时间算法

中图分类号 :O224 **文献标识码** :A **文章编号** :1000-198X(2001)01-0034-06

单件车间的排序问题是最一般和最复杂的排序问题, 只有极个别的特例才找到了有效算法. 1959 年, Wangner^[1]提出用整数规划的方法解决这一问题, 但计算量大, 例如, 机器数为 5, 工件数为 6 的情况, 就有 105 个变量和 174 个约束条件. 1965 年, Brooks 和 White^[2]提出了分支定界法, 其效果比整数规划好, 但它毕竟属于部分列举法. 随着问题规模的增大, 其计算时间的上升幅度亦十分惊人. Ignall 和 Schrage^[3]有这样的统计结果: 用分支定界法求解 3 台机器上加工 n 个工件的流水车间排序问题的最优解所需要的计算时间为加工 $(n-1)$ 个工件需要的计算时间的 2 倍. 这仅是求解排列排序, 即加工顺序总数为 $n!$, 而不是 $(n!)^m$, 这里 m 为机器总数.

规模较大的单件车间排序问题或它的特殊情况——流水车间排序问题, 受计算时间或经济性的制约, 求其最优解是不现实的, 人们普遍采用启发式方法求“满意解”. 启发式方法的效果只能用统计数据比较衡量, 对于某一具体的排序问题, 无论在计算前或在计算后, 都不能确定所求得的结果的满意程度. 这是可以理解的: 启发式方法简单、方便, 但缺乏理论基础, 在找不到实用的优化方法之前, 大量的实际排序问题总得要解决, 只得使用这类方法了.

为此, 本文定义了一般单件车间排序问题的极小点或局部最优解: 任一工件的所有工序在一个排序中都获得最优安排. 与数学上函数极值点的定义相比, 这一排序极值点的定义更为严谨. 函数极值点只允许其自变量沿一个方向增减, 而遍历一个工件在排序中的加工安排方案本身就属于 NP 难题. 为此, 本文将单件车间排序问题转化成一种新型的网络问题, 通过对这种网络的分析和生成, 提出了求解这一排序问题极小点的多项式时间算法.

1 一般单件车间排序问题的描述

与排序问题通常使用的名词术语一致: n 个工件 $J_1, \dots, J_n$ 在 m 台机器 $M_1, \dots, M_m$ 上加工. 一个工件在一台机器上的加工称之一道“工序”. 工件 J_i 的工序数为 $\nu(i)$. 用“加工路线”表示工件加工工序先后顺序的技术约束, 用“加工顺序”表示各台机器上工件加工的先后顺序. 排序亦即确定加工顺序, 使给定的目标函数优化.

1.1 约束条件

所讨论的排序问题一般遵循约束^[4] (a) 任一工件不能同时在不同的机器上加工; (b) 任一工件的上一道工序完成后, 即可进入下一道工序所在的机器上加工; (c) 一旦工件的某道工序开始加工, 必须一直进行到该工序结束; (d) 允许工件在工序之间等待, 允许机器在工件未到达时闲置; (e) 每台机器同时只能加工一个工件.

1.2 目标函数

用 $\{J\}$ 表示以下排序：

$$P(\alpha(1), \beta(1)) \cdots P(\alpha(i), \beta(i))$$

$$P(\alpha(m), \beta(m)) \cdots P(\alpha(n), \beta(n))$$

$p(\alpha(r), \beta(r))$ 不仅表示工件 J_j 的第 j 道工序, 而且表示该工序的加工时间, 它为机器 M_r 上的第 s 个加工.

现用 $a_{r,s}$ 表示工序 $p(\alpha(r), \beta(r))$ 的最早开工时刻, 且若 $\beta(r,s) = 1, s = 1$, 则 $a_{r,s} = 0$. 定义机器 M_r 上开始加工 $p(\alpha(r), \beta(r))$ 前的空闲时间间隔为

$$x_{r,s} = a_{r,s} \quad (s = 1)$$

$$x_{r,s} = a_{r,s} - a_{r,s-1} - p(\alpha(r), \beta(r,s-1)) \quad (2 \leq s \leq \beta(r))$$

由此, 我们获得完成 $\{J\}$ 中全部工件加工所需的时间, 即总流程时间

$$D_{\max} = \max\{D_r \mid r \in [1, m]\}$$

其中
$$D_r = \sum_{s=1}^{\beta(r)} (p(\alpha(r), \beta(r,s)) + x_{r,s})$$

给定所有工件的加工时间和加工路线, 则 $D_{\max}$ 仅取决于如何排序. 本文亦取 $D_{\max}$ 为目标函数^[5].

与 $a_{r,s}$ 相应, 我们用 $b_{r,s}$ 表示 $p(\alpha(r), \beta(r,s))$ 在 $\{J\}$ 中的最迟开工时刻, 且若 $\beta(r,s) = \alpha(i), s = \beta(r)$, 则 $b_{r,s} = D_r - p(\alpha(r), \beta(r,s))$.

1.3 渐变量和关键工序

存在 $\delta > 0$, 对于每一个工序 $p(\alpha(r), \beta(r,s))$, 都有一个正整数 $N_{r,s}$, 使

$$p(\alpha(r), \beta(r,s)) = N_{r,s} \cdot \delta$$

称 δ 为该排序问题的渐变量. 若所有的 $N_{r,s}$ 之间不存在公约数, 则称 δ 为该排序问题的最大渐变量. 亦即, 任一工序都可表示为渐变量 δ 的整数倍. 显然, $D_{\max}$ 也是 δ 的整数倍.

设某一加工顺序 $\{J\}$ 的总流程时间为 $D_{\max}$, 渐变量为 δ , 我们对 $\{J\}$ 中的任一工序作如下分类: 若 $p(\alpha(r), \beta(r,s))$ 增加 δ 时, $D_{\max}$ 亦增加 δ , 则称 $p(\alpha(r), \beta(r,s))$ 为关键工序; 若 $p(\alpha(r), \beta(r,s))$ 减少 δ 时, $D_{\max}$ 亦减少 δ , 则称 $p(\alpha(r), \beta(r,s))$ 为单支关键工序, 简称单支工序; 若 $p(\alpha(r), \beta(r,s))$ 增加 δ 时, $D_{\max}$ 增加 δ , $p(\alpha(r), \beta(r,s))$ 减少 δ 时, $D_{\max}$ 保持不变, 则称 $p(\alpha(r), \beta(r,s))$ 为并支关键工序; 若 $p(\alpha(r), \beta(r,s))$ 增加 δ 时, $D_{\max}$ 保持不变, 则 $p(\alpha(r), \beta(r,s))$ 为非关键工序.

2 极小点和隙

作为最优排序, $\{J\}$ 必须具有这样的性质: 对于给定的目标函数, $\{J\}$ 中的任一工件的加工安排是最优的. 换言之, 从 $\{J\}$ 中取出任一工件的全部工序, 再任意地插回去, 都不可能获得比 $D_{\max}$ 更优的目标值.

定义具备以上性质的排序为极小点, 其总流程时间为极小值. 不具备这一性质的排序也称之点, 但不是极小点. 当然, 从某一点上抽出的工件愈多, 而任意插回去又能满足这一性质, 则该点愈接近于最小点.

定理 1 如果 $\{J\}$ 中任一具有单支工序的工件获得最优安排, 则 $\{J\}$ 为极小点, 反之亦然.

证明 根据单支工序的定义, 若 J_i 不含单支工序, 则从 $\{J\}$ 中抽出 J_i 后, $\{J\}$ 的目标函数值仍保持不变. 另一方面, 若 $\{J\}$ 为极小点, 则 $\{J\}$ 中的每一工件都获得了最优安排, 自然包括具有单支工序的工件在内. (证毕)

已知 $\{J\}$ 中至少包含一个单支工序的工件 J_i 的全部工序为

$$P(\alpha(k), \beta(k)) \cdots P(\alpha(i), \beta(i))$$

$p(\alpha(k), \beta(k))$ 为 J_i 的第 k 道工序, 亦即

$$\beta(\alpha(k), \beta(k)) = k$$

我们用 $\{J\}_i$ 表示抽出 J_i 的全部工序后的 $\{J\}$. 若 J_i 在 $\{J\}$ 中的安排不是最优的, 则存在另一个点, 其目标函数值小于 $D_{\max}$. 基于此, 我们取 $(D_{\max} - \delta)$ 为 $\{J\}_i$ 的总流程时间, 从而定义 $\{J\}_i$ 的隙 $y_{r,s}$ 为 $p(\alpha(r), \beta(r,s))$ 开工前或 $p(\alpha(r), \beta(r))$ 完工后机器 M_r 上的最长闲置时间间隔:

$$y_{r,s} = b_{r,s} \quad (s = 1) \quad (1)$$

$$y_{r,s} = b_{r,s} - a_{r,s-1} - p_{\alpha(r,s-1),\alpha(r,s-1)} \quad (2 \leq s \leq \mu(r) - 1) \quad (2)$$

$$y_{r,s} = D_{\max} - \delta - a_{r,s} - p_{\alpha(r,s),\alpha(r,s)} \quad (s = \mu(r)) \quad (3)$$

根据约束条件(c),处于加工状态的工序不能中断,所以求取 J_i 在 $\{J\}_i$ 中的最优安排时, J_i 的工序仅能插置在 $\{J\}_i$ 的隙中.

对于每一个隙 $y_{r,s}$,用 $A_{r,s}$ 和 $B_{r,s}$ 分别表示其开始时刻和结束时刻,即

$$y_{r,s} = B_{r,s} - A_{r,s}$$

运用式(1)~(3),并注意到

$$B_{r,s} = b_{r,s} \quad (1 \leq s \leq \mu(r) - 1)$$

$$B_{r,s} = D_{\max} - \delta \quad (s = \mu(r))$$

用 $(B_{r,s} - A_{r,s})$ 取代式(2)(3)中的 $y_{r,s}$,于是产生

$$A_{r,s} = 0 \quad (s = 1)$$

$$A_{r,s} = a_{r,s-1} + p_{\alpha(r,s-1),\alpha(r,s-1)} \quad (2 \leq s \leq \mu(r))$$

基于上述分析,使 $\{J\}_i$ 中的所有工序在最早开工时刻加工,我们计算得每个隙的开始时刻 $A_{r,s}$;同理,使 $\{J\}_i$ 中的工序在最迟开工时刻加工,计算得隙的结束时刻 $B_{r,s}$.因此,计算 $\{J\}_i$ 中所有的隙共需2次排序计算.

3 隙网络的生成和分析

3.1 隙

工件 J_i 的第 k 道工序 $p_{i,\alpha(k),\alpha(k)}$ 仅能插置在机器 $M_{\alpha(k)}$ 上的某个隙 $y_{\alpha(k),s}$ 中.根据隙的定义,如果

$$p_{i,\alpha(k),\alpha(k)} > y_{\alpha(k),s}$$

我们将 $p_{i,\alpha(k),\alpha(k)}$ 插置于 $y_{\alpha(k),s}$ 时,则 $\{J\}_i$ 的目标值 $D_{\max}^{(i)} \geq D_{\max}$,此时,无论选择什么方案将 J_i 的其它工序安排到 $\{J\}_i$,都不可能得到比 $D_{\max}$ 更优的目标值.因此,在机器 $M_{\alpha(k)}$ 上我们仅考虑大于或等于 $p_{i,\alpha(k),\alpha(k)}$ 的那些隙:

$$y_{\alpha(k),\alpha(k,1)} \cdots y_{\alpha(k),\alpha(k,\eta(k))}$$

其中 $y_{\alpha(k),\alpha(k,s)}$ 为机器 $M_{\alpha(k)}$ 上的第 $\alpha(k,s)$ 个隙.不失一般性,令

$$\alpha(k) = k$$

$$\alpha(k,s) = s$$

则

$$y_{k,s} = B_{k,s} - A_{k,s} \quad (1 \leq k \leq \nu(i), 1 \leq s \leq \eta(k))$$

我们取 $A_{k,s}$ 为图1中第 k 列上的第 s 个节点.

另外增设初始节点 $A_{0,0} = 0$ 和对应的初始工序 $p_{i,0} = 0$ 终止工序 $p_{i,\alpha(i)+1} = 0$ 和对应的第 $(\nu(i)+1)$ 列终止节点

$$A_{\alpha(i)+1,1} = D_1 \cdots A_{\alpha(i)+1,m} = D_m$$

其中 D_s 为机器 M_s 在 $\{J\}_i$ 中的最早收工时间.

3.2 最短路径

当工序 $p_{i,k}$ 插置于隙 $y_{k,w}$ 时,其后续列上的节点 $A_{r,s}$ 可能会延迟,亦即 $y_{r,s}$ 可能会减小.如果 $p_{i,k}$

之前的工序选择其相应列上的不同隙插置时, $A_{r,s}$ 的延迟时间可能又有所不同.然而,我们需要获得的是 $p_{i,k}$ 插置在 $y_{k,w}$ 时 $A_{r,s}$ 必须延迟的最短时间,我们用从节点 $A_{k,w}$ 到节点 $A_{r,s}$ 的最短路径的值来表示这一最短延迟时间.若 $p_{i,k}$ 插置在 $y_{k,w}$ 后,再将 $p_{i,r}$ 插置 $y_{r,s}$,其 $\{J\}_i$ 的目标值大于或等于 $D_{\max}$,则从 $A_{k,w}$ 到 $A_{r,s}$ 不存在最短路径,用该最短路径的值 $+\infty$ 表示.

首先定义可行隙.如果 J_i 的某道工序插置其相应的某个隙后, $\{J\}_i$ 的目标值小于 $D_{\max}$,则称该隙为可行隙,否则称之为不可行隙.

$A_{1,1}$	$A_{2,1}$	$\cdots$	$A_{v(\hat{i}),1}$	$A_{v(\hat{i})+1,1}$
$A_{1,2}$	$A_{2,2}$	$\cdots$	$A_{v(\hat{i}),2}$	$A_{v(\hat{i})+1,2}$
$A_{0,0}$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$A_{1,\eta(1)}$	$A_{2,\eta(2)}$		$A_{v(\hat{i}),\eta(v(\hat{i}))}$	$A_{v(\hat{i})+1,m}$

图1 $\{J\}_i$ 的隙网络上的所有节点

($A_{r,s}$ 既是第 s 列上第 r 个节点,又是该节点的取值)

Fig.1 All the nodes on the gap-network of $\{J\}_i$

($A_{r,s}$ is both the r th node on s th column and its value)

最短路径的定义和符号如下：

$\Sigma_{(r,s)\backslash k,w}$ ：从 $A_{k,w}$ 到 $A_{r,s}$ 的最短路径，其取值为 $p_{i,k}$ 插置 $y_{k,w}$ 后 $A_{r,s}$ 必须后延的最短时间，且 $A_{r,s}$ 后延了这一时间后， $y_{r,s}$ 仍为可行隙。

$\Sigma_{(r,s)\backslash k,w}^{(k-1,v)}$ ：当 $p_{i,(k-1)}$ 插置 $y_{k-1,v}$ 时，从 $A_{k,w}$ 到 $A_{r,s}$ 的最短路径。

$\Sigma_{(r,s)\backslash k,w}^{(k-1,v)\backslash k-2,u}$ ：当 $p_{i,k-1}$ 插置 $y_{k-1,v}$ 且 $p_{i,k-2}$ 插置 $y_{k-2,u}$ 时，从 $A_{k,w}$ 到 $A_{r,s}$ 的最短路径。

3.3 网络的生成

生成网络就是确定节点间的最短路径和相邻路径及其取值。为此我们需要符号 $\Omega_{(k,w),k-1}$ 第 k 列的节点子集，第 k 列上存在到节点 $A_{r,s}$ 的最短路径的节点都在该子集中。

定理 2 最短路径 $\Sigma_{(r,s)\backslash k,w}$ 存在的充分与必要条件为

$$\Sigma_{(r,s)\backslash k,w} < +\infty \quad (4)$$

$$\text{其中 } \Sigma_{(r,s)\backslash k,w} = \min \left\{ \Sigma_{(r,s)\backslash k,w}^{(k-1,v)} \mid A_{k-1,v} \in \Omega_{(k,w),k-1} \cap \Omega_{(r,s),k-1} \right\} \quad (5)$$

$$\Sigma_{(r,s)\backslash k,w}^{(k-1,v)} = \min \left\{ \Sigma_{(r,s)\backslash k,w}^{(k-1,v)\backslash k-2,u} \mid A_{k-2,u} \in \Omega_{(k-1,v),k-2} \cap \Omega_{(k,w),k-2} \cap \Omega_{(r,s),k-2} \right\} \quad (6)$$

$\Sigma_{(r,s)\backslash k,w}^{(k-1,v)\backslash k-2,u}$ 由以下计算过程确定：

$$\begin{aligned} A'_{k,w} &= A_{k,w} + \Sigma_{(k,w)\backslash k-1,v}^{(k-2,u)} \\ A'_{r,s} &= A_{r,s} + \Sigma_{(r,s)\backslash k-1,v}^{(k-2,u)} \end{aligned}$$

将 $p_{i,k}$ 插置 $y_{k,w}$ 并且取 $A'_{k,w}$ 为开工时间，我们计算 $A'_{r,s}$ 在 $\{J\}_i$ 中的后延时间 $\sigma_{(r,s)\backslash k,w}^{(k-1,v)\backslash k-2,u}$ 。

如果 $B_{r,s} - A'_{r,s} - \sigma_{(r,s)\backslash k,w}^{(k-1,v)\backslash k-2,u} < p_{i,r}$

且 $A'_{k,w} + p_{i,k} > B_{r,s} - p_{i,r}$

则 $\Sigma_{(r,s)\backslash k,w}^{(k-1,v)\backslash k-2,u} = +\infty$

否则 $\Sigma_{(r,s)\backslash k,w}^{(k-1,v)\backslash k-2,u} = \Sigma_{(r,s)\backslash k-1,v}^{(k-2,u)} + \sigma_{(r,s)\backslash k,w}^{(k-1,v)\backslash k-2,u} \quad (7)$

定义 $\sigma_{(r,s)\backslash k,w}^{(k-1,v)\backslash k-2,u}$ 为 $\Sigma_{(r,s)\backslash k,w}^{(k-1,v)\backslash k-2,u}$ 的相邻路径。

证明：

a. 必要性。假设 $\Sigma_{(r,s)\backslash k,w} = +\infty$ ，则根据式 (5) (6) 仅有 2 种可能。

可能性 1： $\Omega_{(k,w),k-1} \cap \Omega_{(r,s),k-1} = \emptyset$ ，亦即，当 $p_{i,k-1}$ 插置第 $(k-1)$ 列上任何一隙 $y_{k,w}$ 和 $y_{r,s}$ 中至少有一个成为不可行隙。

可能性 2： $\Omega_{(k,w),k-1} \cap \Omega_{(r,s),k-1} \neq \emptyset$ ，但对于任一 $A_{k-1,v} \in \Omega_{(k,w),k-1} \cap \Omega_{(r,s),k-1}$ ，当 $p_{i,k-1}$ 插置 $y_{k-1,v}$ 和 $p_{i,k}$ 插置 $y_{k,w}$ 后， $y_{r,s}$ 将转变成不可行隙。

无论以上哪种情况出现，都不存在从 $A_{k,w}$ 到 $A_{r,s}$ 的最短路径。

b. 充分性。我们运用数学归纳法证明充分性条件，同时给出隙网络的生成过程。

证 $k=2$ 时成立。

因为 $A_{0,0}=0$ ， $p_{i,0}=0$ ，所以从节点 $A_{0,0}$ 到它的后续列上的所有节点的最短路径存在且取值都为 0。对于每一 $v \in [1, \gamma(1)]$ 我们插置 $p_{i,1}$ 于 $y_{1,v}$ 并取 $p_{i,1}$ 的开工时间为 $A_{1,v}$ ，从而计算得 $A_{1,v}$ 到后续列上节点的最短路径。现在，我们讨论从 $A_{2,w}$ 到 $A_{r,s}$ 的最短路径。设

$$A_{1,v} \in \Omega_{(2,w),1} \cap \Omega_{(r,s),1} \quad (3 \leq r \leq \gamma(i)+1)$$

如图 2 所示，当 $p_{i,1}$ 插置 $y_{1,v}$ 时，节点 $A_{2,w}$ 和 $A_{r,s}$ 至少后延至

$$A'_{2,w} = A_{2,w} + \Sigma_{(2,w)\backslash 1,v}$$

$$A'_{r,s} = A_{r,s} + \Sigma_{(r,s)\backslash 1,v}$$

插置 $p_{i,2}$ 于 $y_{2,w}$ 且在时刻 $A'_{2,w}$ 开始加工 $p_{i,2}$ ，我们计算得 $A'_{r,s}$ 的后延时间 $\sigma_{(r,s)\backslash 2,w}^{(1,v)}$ ，因此隙 $y_{r,s}$ 减小为

$$y'_{r,s} = B_{r,s} - A'_{r,s} - \sigma_{(r,s)\backslash 2,w}^{(1,v)}$$

若 $y'_{r,s} < p_{i,r}$ ，则只要 $p_{i,1}$ 插置于 $y_{1,v}$ ，就不存在从 $A_{2,w}$ 到 $A_{r,s}$ 的最短路径，这种情况用 $\Sigma_{(r,s)\backslash 2,w}^{(1,v)} = +\infty$ 表示。另一方面，若 $A'_{2,w} + p_{i,2} > B_{r,s} - p'_{i,r}$ ，亦即 $p_{i,2}$ 在 $y_{2,w}$ 中的最早完工时刻超过 $p_{2,r}$ 在 $y_{r,s}$ 中的最迟开工时

刻,这种情况也表示为 $\Sigma_{(r,s)\{2,w\}}^{(1,v)} = +\infty$. 除这两种情况外,我们得到

$$\Sigma_{(r,s)\{2,w\}}^{(1,v)} = \Sigma_{(r,s)\{1,v\}} + \sigma_{(r,s)\{2,w\}}^{(1,v)}$$

因此,取遍每一个节点 $A_{1,v} \in \Omega_{(2,w),1} \cap \Omega_{(r,s),1}$, 最终求得

$$\Sigma_{(r,s)\{2,w\}} = \min\{\Sigma_{(r,s)\{2,w\}}^{(1,v)} | A_{1,v} \in \Omega_{(2,w),1} \cap \Omega_{(r,s),1}\}$$

假设 $(k-1)$ 列成立, 证 k 列成立.

设已知 $p_{i,k-2}$ 插置第 $(k-2)$ 列上不同的隙中时, 第 $(k-1)$ 列上节点到后续列上节点的全部最短路径, 我们来确定 $p_{i,k-1}$ 插置第 $(k-1)$ 列上不同的隙中时, 第 k 列上节点到后续列上节点的全部最短路径. 不失一般性, 设

$$A_{k-2,\alpha}; A_{k-2,\beta} \in \Omega_{(k-1,v),k-2} \cap \Omega_{(k,w),k-2} \cap \Omega_{(r,s),k-2}$$

且

$$\Sigma_{(k,w)\{k-1,v\}} = \Sigma_{(r,w)\{k-1,v\}}^{(k-2,\alpha)}$$

$$\Sigma_{(r,s)\{k-1,v\}} = \Sigma_{(r,s)\{k-1,v\}}^{(k-2,\beta)}$$

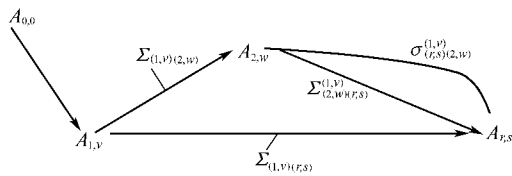


图2 $p_{i,1}$ 插置 $y_{1,v}$ 时确定的 $A_{2,w}$ 到 $A_{r,s}$ 的最短路径“→”和相邻路径“—”

Fig.2 Determining the shortest path “→” and the adjacent path “—” from $A_{2,w}$ to $A_{r,s}$ when $p_{i,1}$ has been inserted in $y_{1,v}$

如图3所示, 我们仅需要考虑2种情况.

情况1: $\alpha = \beta = u$.

即仅当 $p_{i,k-2}$ 插置 $y_{k-2,u}$ 时, $A_{k,w}$ 和 $A_{r,s}$ 同时获得由 $p_{i,k-1}$ 插置 $y_{k-1,v}$ 产生的最短后延时间. 因此, $p_{i,k}$ 在 $y_{k,w}$ 中和 $p_{i,r}$ 在 $y_{r,s}$ 中的最早开工时间分别为

$$A'_{k,w} = A_{k,w} + \Sigma_{(k,w)\{k-1,v\}}$$

$$A'_{r,s} = A_{r,s} + \Sigma_{(r,s)\{k-1,v\}}$$

同理可得相邻路径 $\sigma_{(r,s)\{k,w\}}^{(k-1,v)}$ 和

$$\Sigma_{(r,s)\{k,w\}}^{(k-1,v)} = \Sigma_{(r,s)\{k-1,v\}}^{(k-2,u)} + \sigma_{(r,s)\{k,w\}}^{(k-1,v)} \quad (8)$$

情况2: $\alpha \neq \beta$.

对于这种情况, 我们必须计算第 $(k-2)$ 列上每一个满足必要条件的点, 从而产生式(6). 显然, 式(8)仅是式(6)的特例.(证毕)

3.4 隙网络分析

定理3 对于第 r 列上任一节点 $A_{r,s}$, 若

$$\Omega_{(r,s),k} = \emptyset \quad (9)$$

则 J_i 在 $\{J\}_i$ 中不存在最优安排, 亦即 J_i 在 $\{J\}$ 中的加工安排最优.

证明: 如果 J_i 在 $\{J\}_i$ 中存在最优安排, 则 $p_{i,k}$ 必须插置在 k 列上的某个隙中, 而式(9)表明, 无论 $p_{i,k}$ 插置在 k 列哪个隙, 都会导致第 r 列上无可行隙.(证毕)

因此, 当这种情况出现时, 我们终止 $\{J\}_i$ 的隙网络生成, 转而检查其它具有单支工序的工件在 $\{J\}$ 中的安排是否最优.

网络分析的目的在于从网络上找出 J_i 在 $\{J\}_i$ 中的最优安排. 我们给出存在这一安排的充要条件: $\{J\}_i$ 的隙网络上第 $(i-1)$ 列到 i 列至少存在一条最短路径.

必要性是显而易见的, 证充分性则必须找出这一最优安排, 其途径有多条, 在此略述.

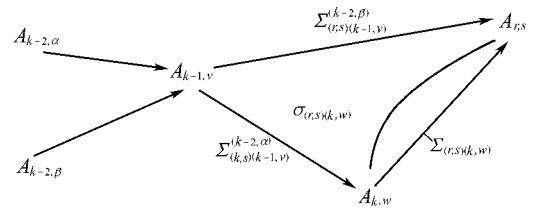


图3 $p_{i,k-1}$ 插置 $y_{1,v}$ 时确定的 $A_{k,w}$ 到 $A_{r,s}$ 的最短路径和相邻路径

Fig.3 Determining the shortest path and the adjacent path from $A_{k,w}$ to $A_{r,s}$ when $p_{i,k-1}$ has been inserted in $y_{k-1,v}$

4 结 论

假设 $D_{\max}$ 减少 $d \times \delta$ 方可达到某个极小点 ,且每个工件在每台机器上仅加工一次 ,即

$$\nu(1) = \dots = \nu(n) = m$$

在此我们取计算一次排序占用的时间为 1 个单位 ,现构造最坏情况^[6] (a)每个工件在优化的过程中始终存在单支工序 (b)任意两个不同列的节点始终存在最短路径 (c)遍历所有的 n 个工件后才使 $D_{\max}$ 减少 δ (d) J_i 的所有隙都为可行隙 .

计算 2 次排序可获得 $\{J_i\}$ 的所有隙和隙的开始时刻 .根据式 (5) (6) ,最多需 n^2 次排序来确定始于某个节点的全部最短路径 .找出极小点 ,需要的总计算时间为

$$f(R) = 2dmn^3$$

其中 R 表示问题的规模 .上式表明 $f(R)$ 具有多项式时间复杂性^[6] .因此 ,本文提出的极小点算法为有效算法 .

参考文献 :

[1] Wangner H M. An integer programming model for machine scheduling[J]. Nav Res Log Q ,1959 ,6 :131 ~ 140 .
[2] Brooks G H ,White C R. An algorithm for finding optimal or near optimal solutions to the production scheduling problem[J]. Journal of Industrial Engineering ,1965 ,16 :34 ~ 40 .
[3] Ignall E ,Schrage L E. Application of the branch and bound technique to some flow-shop problems[J]. Ops Res ,1965 ,13 :400 ~ 412 .
[4] 陈荣秋 . 排序的理论与方法[M] . 武汉 :华中理工大学出版社 ,1987 .4 ~ 5 .
[5] Panwalker S S ,Iskander W. Asurvey of scheduling rule[J]. Ops Res ,1997 ,25 :45 ~ 61 .
[6] Sotskov Y N. The complex of shop-scheduling problems with two or three jobs[J]. European Journal of Operational Research ,1991 ,53 :326 ~ 336 .

Network Analysis for Scheduling

HONG Yu-zheng

(College of International Industry and Commerce ,Hohai Univ . ,Nanjing 210098 ,China)

Abstract :In conformity to the general job shop scheduling problem ,this paper constructs a gap-network consisting of nodes ,shortest paths and adjacent paths .Through analysis of this network ,the local optimal solution of the most general and complex scheduling problem is discussed .Compared with the heuristics ,which have been widely used so far ,the present method is a local optimization method ,and compared with the current optimization methods ,it is of more value to practice .

Key words :job shop ;gap-network ;extremity ;polynomial time complexity algorithm