

汽车装配车间生产计划与调度的 同时优化方法¹⁾

严洪森 夏琦峰 朱 如 刘霞玲

(东南大学自动化研究所 南京 210096)

(E-mail: hsyang@seu.edu.cn)

摘 要 文中提出三种新方法来解决汽车装配车间生产计划与调度的同时优化问题. 首先将汽车装配线简化为一个 Flow shop 问题, 并建立其混合整数规划模型, 以求得使各装配工位的准备成本和空闲时间尽可能少并尽可能满足产品需求的粗生产计划. 然后在粗生产计划的基础上考虑装配线的细节, 用 Tabu 搜索法与快速调度仿真相结合的三种不同启发式算法使生产计划与调度同时得到优化, 并给出了三种算法的复杂性. 大量算例的比较研究表明了这些算法的有效性和适用性.

关键词 装配车间, 生产计划与调度, 混合整数规划, Tabu 搜索

中图分类号 TB491; F213.1

APPROACHES TO SIMULTANEOUS PRODUCTION PLANNING AND SCHEDULING IN AUTOMOBILE ASSEMBLY WORKSHOPS

YAN Hong-Sen XIA Qi-Feng ZHU Min-Ru LIU Xia-Ling

(Research Institute of Automation, Southeast University, Nanjing 210096)

(E-mail: hsyang@seu.edu.cn)

Abstract Three new approaches is presented to the simultaneous production planning and scheduling problem in automobile assembly workshops. First of all, an automobile assembly line is simplified into a flow shop, its mixed integer programming model is formulated to obtain a rough production plan by minimizing the overproduction, underproduction, set-up and leisure time. On the basis of the obtained rough production plan, three different heuristic algorithms combining Tabu search with quick schedule simulation are used to optimize the production plans and schedules simultaneously, with more details of the assembly line being considered. The computational complexity of each algorithm is also given. Comparison between many computational examples of these algorithms is carried out, the result confirms their effectiveness and adaptability.

Key words Assembly workshops, production planning and scheduling, mixed integer

1) 国家“863”CIMS 主题项目(863-511-708-008 和 863-511-943-005)和教育部高等学校骨干教师资助计划资助
收稿日期 2000-08-17 收修改稿日期 2001-09-20

programming, Tabu search

1 引言

在现行的生产计划、调度与优化文献中,往往先单独优化生产计划,然后在被优化的计划基础上优化生产调度,其结果是计划与调度不能达到整体优化,甚至出现生产计划导致调度不可行或性能很差的情况^[1~6].为此,本文将提出三种新的方法来解决汽车装配车间的生产计划与调度的同时优化问题,以使综合性能指标最小化.本文方法的基本思想是,先将汽车装配线简化为一个 Flow shop 问题,并建立其混合整数规划模型,以求得使各装配工位的准备成本和空闲时间尽可能少并尽可能满足产品需求的粗生产计划,然后在粗生产计划的基础上考虑装配线的细节.用 Tabu 搜索法^[7,8]与快速调度仿真相结合的三种不同启发式算法使生产计划与调度同时得到优化.

2 粗生产计划

设一条装配线共有 m 个装配工位,计划区间内依据装配订单共需装 n 种汽车且第 i 种汽车需 d_i 辆.再将装配线简化为一个 Flow shop 问题,且最优计划的目标是最大限度地满足产品需求并使各装配工位的准备成本和空闲时间尽可能少,则可建立求解最优粗生产计划的混合整数规划模型如下:

$$\min J = \sum_{i=1}^n [a_i^+ (x_i - d_i)^+ + a_i^- (d_i - x_i)^+] + \sum_{j=1}^m \sum_{i=1}^n b_{ij} \text{sgn}(x_i) + \sum_{j=1}^m c_j \tau_j \tag{1}$$

$$\text{s. t.} \quad \sum_{i=1}^n t_{ij} x_i + \sum_{i=1}^n \Delta_{ij} \text{sgn}(x_i) + \tau_j = \beta_j \tag{2}$$

式中 n 为计划区间内需装配的汽车种类数; m 为汽车装配线上的装配工位数; x_i 为计划区间内装配第 i 种汽车的产量, $x_i \geq 0$ 且为整数; $\text{sgn}(x_i)$ 为符号函数,当 $x_i > 0$ 时, $\text{sgn}(x_i)$ 取 1, 否则取 0; d_i 为计划区间内对第 i 种汽车的需求,是整数; τ_j 为计划区间内第 j 个装配工位的空闲时间, $\tau_j \geq 0, j = 1, 2, \dots, m$; β_j 为计划区间内第 j 个装配工位的可用时间,是从计划区间内的生产总时间中扣除设备故障维修时间等所剩的时间; a_i^+ 为超产一辆第 i 种汽车的存储及占用流动资金的成本; a_i^- 为欠产一辆第 i 种汽车而违约受罚的成本; b_{ij} 为第 i 种汽车在第 j 个装配工位上的准备成本; c_j 为与第 j 个装配工位资源闲置有关的成本系数; t_{ij} 为第 j 个装配工位装配第 i 种汽车所需要的时间; Δ_{ij} 为第 i 种汽车在第 j 个装配工位上的准备时间; $(c)^+ \text{ 为 } \max(0, c)$.

式(1)和(2)的混合整数非线性规划模型可转化为混合整数线性规划模型如下:

$$\min J = \sum_{i=1}^n (a_i^+ \Delta^+ x_i + a_i^- \Delta^- x_i) + \sum_{j=1}^m \sum_{i=1}^n b_{ij} y_i + \sum_{j=1}^m c_j \tau_j \tag{3}$$

$$\text{s. t.} \quad \sum_{i=1}^n t_{ij} x_i + \sum_{i=1}^n \Delta_{ij} y_i + \tau_j = \beta_j \tag{4}$$

$$x_i - \Delta^+ x_i + \Delta^- x_i = d_i \tag{5}$$

$$B y_i \leq x_i \tag{6}$$

式中 $x_i \geq 0$ 且为整数; y_i 是布尔变量, $y_i \in (0, 1)$, 当 $x_i > 0$ 时为 1, 否则为 0; $\Delta^+ x_i \geq 0, \Delta^- x_i \leq 0$; B 是一个大的正数.

到此, 就可用分枝定界或割平面算法求解式(3) ~ (6) 的混合整数线性规划模型, 以获取使各装配工位的准备成本和空闲时间尽可能少并尽可能满足产品需求的粗生产计划.

3 生产计划与调度的同时优化

为加快问题求解速度, 在式(3) ~ (6) 的模型中忽略了装配线的细节, 并由此求出粗生产计划作为后续生产计划与调度同时优化问题迭代求解的初始计划. 另一方面, 考虑细节的同步装配线调度往往是一个非结构化问题, 很难用解析的方法来求解, 较为可行的办法是使用基于可变时间流的快速调度仿真. 此处快速的含义是不要图形显示, 仅仅通过快速仿真计算给定调度的性能指标. 至于最优计划与调度的迭代选择问题则由基于 Tabu 搜索的三种不同启发式算法来解决.

3.1 嵌入式 Tabu 搜索算法

为节省汽车装配准备时间, 同种汽车集中装配, 不分割成多个装配任务, 并在调度时仅占用一个排序位置. 参照式(1)和(2), 则求解汽车装配车间生产计划与调度同时优化问题的数学模型可描述如下:

min_{x,S} G(x,S) = min_{x,S} \left\{ \sum_{i=1}^n [a_{\mu(S,i)}^+(x_{\mu(S,i)} - d_{\mu(S,i)})^+ + a_{\mu(S,i)}^-(d_{\mu(S,i)} - x_{\mu(S,i)})^+] + \right. \\ \left. \sum_{j=1}^m b_{\mu(S,i)j} \text{sgn}(x_{\mu(S,i)}) + \sum_{j=1}^m c_j \tau + rf(x,S) + \right. \\ \left. q \sum_{j=1}^m \left[(\beta_j - \tau) - \frac{1}{m} \sum_{j=1}^m (\beta_j - \tau) \right]^2 \right\} \tag{7}

s. t. \quad \sum_{i=1}^n t_{\mu(S,i)j} x_{\mu(S,i)} + \sum_{i=1}^n \Delta t_{\mu(S,i)j} \text{sgn}(x_{\mu(S,i)}) + \tau = \beta_j \tag{8}

h(\mu(S,i)) \leq \psi_{\mu(S,i)} \tag{9}

x_{\mu(S,i)} \geq 0 \text{ 且为整数}, \tau \geq 0, j = 1, 2, \dots, m \tag{10}

式中 $\mu(S,i)$ 为 n 种汽车或其中部分通过装配线的顺序(调度) S 中的第 i 个位置所对应的汽车种类, 如果对于 $k \leq n$ 有 $\mu(S,i) = 0$, 则表示调度 S 中最多只包含 $k-1$ 种汽车; $b_{\mu(S,i)j}$ 为第 $\mu(S,i)$ 种汽车在第 j 个装配工位上的准备成本; $\Delta t_{\mu(S,i)j}$ 为第 $\mu(S,i)$ 种汽车在第 j 个装配工位上的准备时间; $h(\mu(S,i))$ 为第 $\mu(S,i)$ 种汽车的下线时间; $\psi_{\mu(S,i)}$ 为第 $\mu(S,i)$ 种汽车的交付期; $x = (x_{\mu(S,1)}, x_{\mu(S,2)}, \dots, x_{\mu(S,n)})^T$ 为生产计划向量; $f(x,S)$ 为汽车装配任务全部完成的时间, 由相应计划和调度共同决定; r 为任务完成时间权系数; q 为各装配工位负荷均衡权系数.

Tabu 搜索是由 Glover^[7,8] 提出的用于获取组合最优化难题近似解的一种高级启发式方法. 在应用 Tabu 搜索法求解生产计划与调度同时优化问题(7) ~ (10) 之前, 首先介绍两个概念, 即相邻计划和相邻调度. 设计划 $p^{**} = (x_1, x_2, \dots, x_n)^T$ 且对某些 i 有 $\tau_i > t_{ij} > 0$, 对 $j = 1, 2, \dots, m$, 则定义 $p = (x_1, x_2, \dots, x_{i-1}, x_i + 1, x_{i+1}, \dots, x_n)^T$, $p = (x_1, x_2, \dots, x_i \pm 1, \dots, x_{k-1}, \dots, x_n)^T$ 和 $p = (x_1, x_2, \dots, x_{i-1}, x_i - 1, x_{i+1}, \dots, x_n)^T$ 为 p^{**} 的相邻计划; 否则定义 $p = (x_1, x_2, \dots, x_i \pm 1, \dots, x_{k-1}, \dots, x_n)^T$ 和 $p = (x_1, x_2, \dots, x_{i-1}, x_i - 1, x_{i+1}, \dots, x_n)^T$ 为 p^{**} 的相邻计划. 可见 p^{**} 的相邻计划最多可能有 $n^2 + n$ 个.

对于调度 S , 我们定义仅交换 S 中的两个元素而形成的调度为相邻于 S 的调度.

不失一般性, 设初始调度 S_0 中共包含三种汽车, 即 $S_0 = \{a, b, c\}$, 则依定义, 相邻于 S_0 的调度共有三种: $S_1 = \{b, a, c\}$, $S_2 = \{c, b, a\}$, $S_3 = (a, c, b)$. 注意相邻调度仅交换初始调度 S_0 中的两个元素而得. 如 S_1 是仅交换 S_0 中的 a, b 两个元素而得到的, 是相邻于 S_0 的调度; 但 $\{c, a, b\}$ 是交换 S_3 中的 a, c 两个元素而得到的, 不能直接交换 S_0 中的两个元素而得, 所以 $\{c, a, b\}$ 不是相邻于 S_0 的调度. 一般地, 对于包含 n 种汽车的调度 S^{**} , 则相邻调度共有 $n(n-1)/2$ 种.

目前, Tabu 搜索法主要用于 Flow shop 调度、Job shop 调度和制造单元形成等方面^[9~10], 而在汽车装配线生产计划、调度方面的应用则非常少见. 本文要解决的是计划与调度的同时优化问题, 所以需要提出一种新的基于 Tabu 搜索的高级启发式算法, 即嵌入式 Tabu 搜索算法来解决. 其基本思想是以粗生产计划作为初始解, 在计划层用 Tabu 搜索寻找最好的计划, 并对计划层生成的每个相邻计划用另一个 Tabu 搜索寻找经过快速仿真计算具有最好性能指标的调度, 直至使生产计划与调度同时达到优化. 因一个 Tabu 搜索嵌套着另一 Tabu 搜索, 故取名为嵌入式 Tabu 搜索算法.

算法 1. 生产计划与调度同时优化问题 (7) ~ (10) 的嵌入式 Tabu 搜索算法 (ETS)

Step1. 初始化

- 1) 读取式 (7) ~ (9) 中的各种数据和参数;
- 2) 读取算法参数, 包括生产计划 Tabu 表长度 $PT\text{-size}$ 、给定的计划移动次数 $PM\text{-max}$ 、调度 Tabu 表长度 $ST\text{-size}$ 和给定的调度移动次数 $SM\text{-max}$;
- 3) 设置 $G\text{-best} = M\text{-big}$, $PM\text{-ctr} = 0$, $PT\text{-list} = \{\phi\}$, $p\text{-best} = \phi$, $S\text{-best} = \{\phi\}$.

Step2. 搜索初始可行计划与调度

- 1) 设置初始生产计划 $p_0 = x$;
- 2) 设置当前计划 $p = p_0$, 并调用算法 2 以搜索给定计划 p 的可行调度;
- 3) 如果 $SC\text{-flag} = 1$ 且 $G(p, S^{**}) < G\text{-best}$, 则设置 $p^{**} = p$, $p\text{-best} = p$, $S\text{-best} = S^{**}$, $G\text{-best} = G(p, S^{**})$, 然后转 Step3;
- 4) 生成 p_0 的一个相邻计划 p , 设置 $p_0 = p$. 然后转 Step2 的 2)。

Step3. 搜索最好计划与调度

- 1) 生成一个相邻于 p^{**} 的所有可能计划集, 并置 $G(p^{**}, S^{**}) = M\text{-big}$;
- 2) 对于该集合中的一个计划 p , 如果 p 不在 $PT\text{-list}$ 中, 则调用算法 2 以搜索对于给定计划 p 的最好调度, 并更新当前相邻集中的最好计划 $p^{**} = p$, $G(p^{**}, S^{**}) = G(p, S^{**})$, 如果 $SC\text{-flag} = 1$ 且 $G(p, S^{**}) < G(p^{**}, S^{**})$; 否则丢弃 p ; 如果计划 Tabu 表中 $PT\text{-list}$ 中的计划个数少于 $PT\text{-size}$, 将 p 加到 $PT\text{-list}$ 的顶部; 如此重复, 直至做完 p^{**} 的所有相邻计划;

3) 作一次计划移动, 设置 $p^{**} = p^{**}, G(p^{**}, S^{**}) = G(p^{**}, S^{**})$ 且如果 p^{**} 不在 $PT\text{-list}$ 中则将 p^{**} 加到 $PT\text{-list}$ 的顶部; 如果 $PT\text{-list}$ 中的计划个数 $> PT\text{-size}$, 则从 $PT\text{-list}$ 的底部删除一个最老的计划;

4) 如果计划有改善, 即如果 $G(p^{**}, S^{**}) < G\text{-best}$, 则更新最好解 $p\text{-best} = p^{**}$, $S\text{-best} = S^{**}$, $G\text{-best} = G(p^{**}, S^{**})$;

5) 更新到目前为止所作的计划移动次数 $PM\text{-ctr} = PM\text{-ctr} + 1$; 如果 $PM\text{-ctr} > PM\text{-max}$, 则停止迭代并转 Step4, 否则转 Step3 的 1)。

Step4. 输出结果

输出 $p\text{-best}$, $S\text{-best}$ 和 $G\text{-best}$.

这里 $M - big$ 为一个很大的数; $SC - flag$ 为对于给定计划 p 是否存在满足约束(8) ~ (10) 的调度的标志, $SC - flag = 1$ 表示存在, $SC - flag = 0$ 表示不存在; p^* 为在当前相邻计划集中的最好计划; p^{**} 为在紧前相邻计划集中的最好计划; $p - best$ 为到目前为止找到的最好生产计划; S^{***} 为相应计划的最好调度; $S - best$ 为针对到目前为止找到的最好生产计划, 所找到的最好调度; $G(p, S)$ 为与生产计划 p 和调度 S 相对应的性能指标; $G - best$ 为到目前为止所达到的最好指标.

算法 2. 基于 Tabu 搜索的调度优化

Step1. 初始化

设置 $SM - ctr = 0$, 调度 Tabu 表 $ST - list = \{\phi\}$.

Step2. 寻找初始调度

1) 对给定的当前生产计划 p , 依次按最早交付期和最少批量等优先规则, 确定初始调度 S_0 , 并置 $S^* = S_0$, $S^{**} = S_0$, $SC - flag = 0$;

2) 通过快速调度仿真, 计算相应于 p 和 S_0 的性能指标 $G(p, S_0)$, 并置 $G(p, S^{**}) = G(p, S_0)$, 如果 S_0 可行(即满足约束(8) ~ (10)), 置 $SC - flag = 1$.

Step3. 调度搜索

1) 生成一个相邻于紧前相邻调度集中的最好调度 S^* 的所有可能调度集, 并置 $G(p, S^*) = M - big$;

2) 对于该集合中的一个调度 S , 如果 S 不在 $ST - list$ 中, 则通过快速调度仿真, 计算调度性能指标 $G(p, S)$, 并更新当前相邻集中的最好调度 $S^* \leftarrow S, G(p, S^*) \leftarrow G(p, S)$, $SC - flag = 1$ 如果 S 可行且 $G(p, S) < G(p, S^*)$; $S^* \leftarrow S, G(p, S^*) \leftarrow G(p, S)$ 如果 S 和 S^* 都不可行且 $G(p, S) < G(p, S^*)$; 否则丢弃 S ; 如此重复, 直至做完 S^* 的所有相邻调度;

3) 作一次调度移动, 设置 $S^{**} \leftarrow S^*, G(p, S^{**}) \leftarrow G(p, S^*)$, 并将 S^* 加到 $ST - list$ 的顶部, 如果 $ST - list$ 中的调度个数 $> ST - size$, 则从 $ST - list$ 的底部删除一个最老的调度.

4) 如果调度有改善, 即 S^{**} 可行且 $G(p, S^{**}) < G(p, S^{***})$ 或 S^{**} 和 S^{***} 都不可行且 $G(p, S^{**}) < G(p, S^{***})$, 则更新最好解 $S^{***} \leftarrow S^{**}, G(p, S^{***}) \leftarrow G(p, S^{**})$;

5) 更新对当前计划所作的调度移动次数 $SM - ctr \leftarrow SM - ctr + 1$, 如果 $SM - ctr > SM - max$, 则停止迭代并转 Step4; 否则转 Step3 的 1) .

Step4. 返回.

算法 1 Step2 中的 x 是由求解式(3) ~ (6) 所获得的粗生产计划并被当作算法的初始解以加快问题的求解速度, 因为粗生产计划往往比随机选择的初始解更接近式(7) ~ (10) 的真实解. 尽管如此, 由于式(3) ~ (6) 忽略了装配线调度等方面的细节, 由此获得的粗生产计划对式(7) ~ (10) 来讲往往还是不可行. 此外, 从粗生产计划演变成可行计划的有效方法无疑是减少装配线的负荷, 所以算法 1 Step2 中的相邻计划可只定义为 $p = (x^1, x^2, \dots, x^{i-1}, x^{i+1}, \dots, x^n)^T$, 以加快获得初始可行解的速度.

如果汽车装配线具有随机特性(如设备故障、装配时间发生变化等), 可通过算法 1 的蒙特卡罗运行来求解其同时优化问题. 具体做法是: 1) 在每次蒙特卡罗运行之前, 先按照各随机变量的分布函数计算各自的样本值并用这些样本值代替式(7) ~ (9) 中的相应值, 然后调用算法 1; 2) 在第一次蒙特卡罗运行时, 算法 1 Step2 中的 x 为粗生产计划, 但第二次及以后的蒙特卡罗运行时, x 则为前次蒙特卡罗运行后所获得的最好生产计划以加快其求解速度, 因为尽管当前蒙特卡罗运行的样本值很可能同前次蒙特卡罗运行的不一样, 但前次最好生产计划往往还是比粗生产计划更接近本次最优解; 3) 所有蒙特卡罗运行结束后, 需计算除最

好调度以外的所有结果平均值,并用圆整后的最好计划平均值及随机参数平均值作为输入再一次调用算法 2 以获取相应的最好调度,最后将其和圆整的最好计划平均值及其它结果的平均值一起作为具有随机特性的汽车装配线生产计划与调度同时优化问题的解。

算法 1 的计算复杂性是 $o(0.5n^2(n^2-1) \times SM - \max(PM - \max))$ 次仿真。如果 n 比较大,用算法 1 求解式 (7) ~ (10) 会需要太多的时间以致无法在可接受的时间内获得最优解。为此我们将提出另一种求解式 (7) ~ (10) 的算法,即交替 Tabu 搜索法。

3.2 交替 Tabu 搜索算法

交替 Tabu 搜索算法的基本思想: 1) 从初始生产计划开始寻找一个可行计划与调度; 2) 给定调度,用 Tabu 搜索寻找最好的计划; 3) 反过来给定计划,又用另一 Tabu 搜索寻找最好的调度; 4) 交替使用 2), 3) 两步直至找到最好的计划与调度。由于分别对计划与调度交替使用两个 Tabu 搜索,故称为交替 Tabu 搜索算法。

算法 3. 生产计划与调度同时优化问题的交替 Tabu 搜索算法(ATS)

Step1. 初始化

同算法 1 的 Step1.

Step2. 搜索初始可行计划与调度

同算法 1 的 Step2.

Step3. 搜索最好计划与调度

1) 同算法 1 的 Step3 的 1);

2) 同算法 1 Step3 的 2), 除了以这里的“通过快速调度仿真计算相应 p 和 S^{**} 的目标值 $G(p, S^{**})$ ”替换那里的“调用算法 2 以搜索对于给定计划 p 的最好调度”;

3) 调用算法 2 以搜索对于给定 p^* 的最好调度 S^{**} ;

4) 同算法 1 的 Step3 的 3);

5) 同算法 1 的 Step3 的 4);

6) 同算法 1 的 Step3 的 5)。

Step4. 输出结果同算法 1 的 Step4.

算法 3 的计算复杂性是 $o((n^2 + n + 0.5n(n-1)SM - \max)PM - \max)$ 次仿真。如果 n 足够大,则算法 1 的复杂性是 $o(n^4)$,而算法 3 是 $o(n^2)$ 。所以算法 3 比算法 1 快得多,但后者往往比前者获得更好解。如果 n 非常大,即使算法 3 也无法在可接受的时间内获得最优或次优解。为此,我们将提出串行 Tabu 搜索法来求解式 (7) ~ (10)。

3.3 串行 Tabu 搜索算法

无疑,加速求解过程的一种有效方法是减少算法的计算复杂性,也就是减少为获取最好调度所做的仿真总次数。为此,串行 Tabu 搜索算法的基本思想是 1) 从初始计划开始寻找一个可行计划; 2) 从可行计划开始,使用 Tabu 搜索寻找最好的计划; 3) 对于最好的计划,使用另一 Tabu 搜索寻找最好的调度。由于对计划和调度依次使用 Tabu 搜索,故称为串行 Tabu 搜索算法。

算法 4. 生产计划与调度同时优化问题的串行 Tabu 搜索算法(STS)

Step1. 初始化

同算法 1 的 Step1.

Step2. 搜索可行计划

同算法 1 的 Step2,除了以这里的“调用算法 5 以确定相应于 p 的调度”替换那里的“调用算法 2 以搜索给定计划 p 的可行调度”和以这里的 S 替换那里的 S^* 。

Step3. 搜索最好计划

同算法 1 的 Step3, 除了以这里的“调用算法 5 以确定相应于 p 的调度”替换那里的“调用算法 2 以搜索对于给定计划 p 的最好调度”, 以这里的‘ S ’替换那里的‘ S^{***} ’和以这里的“转 Step4”替换那里的“停止迭代并转 Step4”。

Step4. 搜索最好调度

- 1) 设置 $p = p\text{-best}$, 并调用算法 2 以搜索相应于 p 的最好调度;
- 2) 如果调度有改善, 即 $G(p, S^{***}) < G\text{-best}$, 则更新最好解 $S\text{-best} = S^{***}$, $G\text{-best} = G(p, S^{***})$.

Step5. 输出结果

同算法 1 的 Step4.

这里 S 是由算法 5 确定的一个调度.

算法 5. 相应于给定计划的调度确定算法

Step1. 确定调度

- 1) 对于给定计划 p , 依次按照最早交付期、最少批量等优先规则, 确定调度 S ;
- 2) 通过快速调度仿真, 计算相应于 p 和 S 的 $G(p, S)$;
- 3) 如果 S 不可行, 则置 $SC\text{-flag} = 0$, 否则置 $SC\text{-flag} = 1$;

Step2. 返回.

算法 4 的计算复杂性是 $o((n^2 + n)PM\text{-max} + 0.5n(n-1)SM\text{-max})$ 次仿真. 如果 n 足够大, 算法 4 的复杂性同算法 3 一样都是 $o(n^2)$. 但是, 使用算法 4 从初始可行计划开始搜索到最好计划与调度要比使用算法 3 少做 $[1 + 0.5n(n-1)SM\text{-max}](PM\text{-max} - 1)$ 次仿真. 因此, 算法 4 要比算法 3 快, 但后者往往比前者获得更好解.

在生产计划与调度的同时优化中, 计划的职能是确定每种汽车的装配批量, 调度的职能是确定每种汽车按什么样的顺序上线装配.

4 举例

求解汽车装配线生产计划与调度同时优化问题的嵌入式 Tabu 搜索算法(ETS), 交替 Tabu 搜索算法(ATS)和串行 Tabu 搜索算法(STS), 以及求解式(3)~(6)的分枝定界算法已用 VC++ 5.0 编成软件. 借助于这些算法软件, 我们研究了汽车装配线生产计划与调度的同时优化问题. 在下面的例子中, 所有这四种算法软件都在内存 128M 的 Pentium 450 PC 机上 Win 98 环境中运行, 并只考虑两种汽车装配线: 一种汽车装配工位无缓冲区, 另一种有缓冲区. 每种装配线同南京某汽车总装厂的装配线一样都有 33 个工位, 并且有缓冲区装配线的每个工位的虚拟缓冲区都有 3 个存放位置(一个存放位置只能放一辆汽车).

例 1. 考虑有准备有缓冲区随机情况. 假定某班需要装配 5 种汽车, 其产量分别为 4, 5, 14, 20, 5 辆. 计划区间为一班(480 分). 式(7)~(9)中的其它参数已经给定, 其中装配时间 t_{ij} 是一个服从正态分布的随机变量, 方差为其均值的平方乘以 0.0025. 每个工位前后两个故障之间的间隔和设备维修时间都服从指数分布. 第 j ($j = 1, 2, \dots, 33$) 个工位的故障率 λ 为 $1/480$, 其维修率 θ 为 $1/2$. 算法参数 $PT\text{-size} = 80$, $ST\text{-size} = 10$, $PM\text{-max} = 100$, $SM\text{-max} = 3$.

在 PC 机上运行分枝定界算法软件 3.0 秒后, 我们获得粗生产计划为 $x = (5, 6, 35, 20, 5)^T$, 其中第 i 个分量为第 i 种汽车的计划产量. 然后, 以 x 为初始解, 在 PC 机上分别使用 ETS,

ATS 和 STS 软件各做了 6 次蒙特卡罗运行, 获得其结果的平均值如表 1 所示. 表中使用 ETS 软件获得的最好计划是(4, 7, 16, 19, 7), 其中第 i 个分量为第 i 种汽车的计划产量; 使用 ETS 软件获得的最好调度为{4, 1, 5, 3, 2}, 其中第 i 个元素为调度中第 i 个位置的汽车类型. 仿真时间为每种算法的 6 次蒙特卡罗运行的平均时间, 且不包括读参数和计算样本值的时间(23.38 秒). ETS 和 ATS 的仿真时间分别比 STS 长 22.20 倍和 1.47 倍. 但 ETS 和 ATS 的最好性能指标且分别比 STS 少 14.56% 和 13.46%.

表 1 有准备有缓冲区随机情况的蒙特卡罗运行结果平均值

算法	p_best	S_best	G_best	仿真时间(分)
ETS	(4, 7, 16, 19, 7)	{4, 1, 5, 3, 2}	145825	85.62
ATS	(5, 8, 14, 19, 7)	{5, 1, 2, 4, 3}	147709	9.10
STS	(5, 3, 18, 21, 7)	{5, 1, 2, 4, 3}	170676	3.69

例 2. 考虑无准备随机情况, 其它条件同例 1.

在 PC 机运行分枝定界算法软件 4.0 秒后, 我们获得粗生产计划 $x = (4, 5, 36, 20, 5)^T$. 以 x 为初始解. 在 PC 机上针对有无缓冲区两种汽车装配线分别使用 ETS, ATS 和 STS 软件各做了 6 次蒙特卡罗运行后, 获得了各自结果平均值. 其中仿真时间为 0.24 ~ 35.24 分. ETS 和 ATS 的仿真时间平均比 STS 长 9.06 倍和 1.13 倍. 但 ETS 和 ATS 的最好性能指标且平均比 STS 少 7.41% 和 6.21%. 有缓冲区汽车装配线的最好性能指标平均比无缓冲区汽车装配线少 5.29%.

例 3. 考虑无准备确定情况, 假定某班需要装配 10 种汽车.

在 PC 机上运行分枝定界算法软件 11.2 秒后, 获得了粗生产计划. 以该粗生产计划为初始解, 在 PC 机上针对有无缓冲区两种汽车装配线分别运行 ETS, ATS 和 STS 软件后, 获得了各自结果. 其中仿真时间为 0.52 ~ 991.10 分. ETS 和 ATS 的仿真时间平均比 STS 长 89.60 倍和 64.96%. 但 ETS 和 ATS 的最好性能指标且平均比 STS 少 3.79% 和 1.73%. 有缓冲区汽车装配线的最好性能指标平均比无缓冲区汽车装配线少 3.17%.

例 4. 考虑无准备确定情况, 假定某班需要装配 30 种汽车.

在 PC 机上运行分枝定界算法软件 102.2 秒后, 获得了粗生产计划. 以该粗生产计划为初始解, 在 PC 机上针对有无缓冲区两种汽车装配线分别运行 ATS 和 STS 软件后, 获得了各自结果(ETS 无结果, 因无法在可接受的时间内获得最好解). 其中仿真时间为 5.32 ~ 482.85 分. ATS 的仿真时间平均比 STS 长 2.64 倍. 但 ATS 的最好性能指标且平均比 STS 少 4.33%. 有缓冲区汽车装配线的最好性能指标平均比无缓冲区汽车装配线少 8.52%.

5 结论

本文研究了汽车装配车间生产计划与调度的同时优化问题, 着重讨论了粗生产计划的生成方法、生产计划与调度同时优化问题的三种高级启发式求解方法(ETS, ATS 和 STS), 并给出了它们的计算复杂性. 开发了相应软件. 借助这些软件, 用大量算例进行了比较研究, 结果表明:

1) 与传统方法相比, 本文方法的优点是将解析方法、Tabu 搜索和快速调度仿真有机地结合在一起, 有效地解决了汽车装配车间生产计划与调度的同时优化问题, 并保证至少有一个可行解;

2) ETS 的问题求解速度最慢, 但获得的性能指标往往最好; STS 的问题求解速度和获得的性能指标正好与 ETS 相反; 而 ATS 则介于 ETS 和 STS 之间;

3) ETS 适合求解小问题, ATS 适合求解中规模问题, STS 适合求解大规模问题; 如果问题非常大, 则建议使用算法 4 的 Step 1 ~ 2 求出可行计划与调度作为问题的解;

4) 具有随机特性的汽车装配线生产计划与调度的同时优化问题可用 ETS, ATS 和 STS 的蒙特卡罗运行求解;

5) 如果可行, 应采用具有虚拟缓冲区的汽车装配线, 因为有虚拟缓冲区的性能指标往往比无缓冲区好。

此外, 本文的 ETS, ATS 和 STS 也可用于求解其目标函数和约束不同于式(7) ~ (10) 的模型, 以及其它行业(如家电等) 装配线的生产计划与调度的同时优化问题. 这些算法已在汽车装配车间生产计划与调度的集成优化系统中加以实现, 并应用于南京某汽车总装厂, 产生了显著经济效益。

参 考 文 献

1 Lasserre J B. An integrated model for job-shop planning and scheduling. *Management Science*, 1992, **38**(8): 1201 ~ 1211

2 Yan H S. An interaction/ prediction approach to hierarchical production planning and control with delay interaction equations. *Computer Integrated Manufacturing Systems*, 1997, **10**(4): 309 ~ 320

3 Yan H S, Jiang Z J. An interaction/ prediction approach to solving problems of hierarchical production planning in flexible automation workshops. *International Journal of Computer Integrated Manufacturing*, 1998, **11**(6): 513 ~ 523

4 Yan H S. Hierarchical stochastic production planning with delay interaction. *Journal of Optimization Theory and Applications*, 2000, **104**(3): 659 ~ 689

5 严洪森, 张晋格, 王 炎, 黎贞渭. 基于 EHLEP-N 模型的 FMS 实时调度和控制. *自动化学报*, 1992, **18**(6): 679 ~ 685

6 Yan H S, Wang N S, Zhang J G, Cui X Y. Modelling, scheduling and simulation of flexible manufacturing systems using extended stochastic high-level evaluation Petri nets. *Robotics and Computer-Integrated Manufacturing*, 1998, **14**(2): 121 ~ 140

7 Glover F. Tabu search- part I. *ORSA Journal on Computing*, 1989, **1**(3): 190 ~ 206

8 Glover F. Tabu search- part II. *ORSA Journal on Computing*, 1990, **2**(1): 4 ~ 32

9 Nowicki E. The permutation flow shop with buffers: A tabu search approach. *European Journal of Operational Research*, 1999, **116**(1): 205 ~ 219

10 Armentano V A, Ronconi D P. Tabu search for total tardiness minimization in flow shop scheduling problems. *Computers and Operations Research*, 1999, **26**(3): 219 ~ 235

严洪森 1982 年毕业于哈尔滨船舶工程学院自动控制系, 1989 年和 1992 年分别在哈尔滨工业大学电气工程系和控制工程系获工学硕士和博士学位, 1992 年至 1994 年在南京航空航天大学机械工程学科从事博士后研究工作, 现任东南大学自动化研究所教授, 博士生导师, 美国工业工程师学会(IIE) 高级会员. 主要研究方向为 CIMS 及 FMS 建模、生产计划、调度、控制、仿真、并行工程、敏捷制造和知识化制造等。

夏琦峰 1977 年和 2000 年分别在东南大学自动控制系和自动化研究所获工学学士和硕士学位, 现任华为技术有限公司上海研究所工程师. 主要研究方向为生产计划、调度与控制。

朱 如 1992 年和 1999 年分别在东南大学电气工程系和自动化研究所获工学学士和硕士学位, 现任南京邮电学院信息网络技术研究所工程师. 主要研究方向为管理信息系统和工业自动化。

刘霞玲 1998 年在武汉化工学院工业自动化系获工学学士学位, 2001 年在东南大学自动化研究所获工学硕士学位, 现任华为技术有限公司上海研究所助理工程师. 主要研究方向为生产计划与调度。