

文章编号: 1001-0920(2002)03-0353-04

一种监控与调度集成的滚动式算法

刘斌, 李忠卫, 曾建潮

(太原重型机械学院 系统仿真与计算机应用研究所, 山西 太原 030024)

摘要: 给出一种通过滚动调度将动态调度与监控集成的方法, 分析了在实际调度中改变调度策略的必要性和可行性, 对滚动调度中的滚动窗口进行了描述。仿真实例表明该算法是有效的。

关键词: 批处理生产过程; 动态调度; 监控

中图分类号: TP 278 **文献标识码:** A

Rolling algorithm of the integration of scheduling and supervision for batch processes

LIU Bin, LI Zhong-wei, ZENG Jian-chao

(Division of System Simulation and Computer Application,

Taiyuan Heavy Machinery Institute, Taiyuan 030024, China)

Abstract: Based on the analysis of the importance and feasibility for changing the scheduling strategy in actual industrial processes, the rolling zone is defined, and a rolling algorithm is presented for the integrated scheduling and supervision of batch processes. Simulation results show the validity of the algorithm.

Key words: batch processes; dynamic scheduling; supervision

1 引言

批处理生产过程中的动态调度问题一直没有得到很好的解决。这是因为在实际过程中, 被控对象一般都是一个复杂的混合系统, 除了会受到外界的干扰外, 系统的实际模型总会与调度中用到的状态方程有所出入。在这种情况下, 只采用单一不变的调度策略已无法达到最优调度的目的。所以, 有必要在静态调度的基础上引入监控, 从而不断地跟踪被控对象实际状态的变化, 进而调整调度规则, 使被控对象朝着所希望的方向运行。

2 调度监控集成与滚动式算法

在这方面, 采用滚动式方法将动态调度与监控集成的方法是一种较好的选择。这是因为: 首先, 滚动式调度方法的各个滚动窗口的求解规模和所需时间一般都比较小, 适合于在线计算; 其次, 滚动式方法能跟踪系统的变化, 在其滚动窗口的每一个开始时刻可以根据该时刻系统的实际情况调整其初始状态和初始条件^[1,2]。

这里使用文献[3]中的模型。 $t_{j,i}$, τ_{i-1} 分别表示在 τ_{i-1} 时刻激发第 i 个变迁, 且该变迁为 t_j ; $p v(\tau_{i-1})$,

收稿日期: 2001-01-05; 修回日期: 2001-06-06

基金项目: 山西省青年科学基金项目(981601)

作者简介: 刘斌(1975—), 男, 湖北咸宁人, 硕士生, 从事混合动态系统理论等研究; 曾建潮(1963—), 男, 陕西大荔人, 教授, 博士, 从事智能计算、复杂系统建模、控制与仿真等研究。

$t_{j,i}$ 表示在 τ_{-1} 时刻激发 t_j 变迁会产生利润; $mv(\tau_{-1}, t_{j,i})$ 表示在 τ_{-1} 时刻激发 t_j 变迁会消耗的材料花费; $sv(\tau_{-1}, \tau, t_{j,i}, t_{j1,i+1})$ 表示相邻两个变迁激发时刻之间的存储费用。

集成的结构如图 1 所示。监控层对运行的被控对象进行监测,不断地采样和计算,得到在各个采样时刻的某些对对象的输出影响较大的变量值。若发现某时刻该值的变化已超出其正常范围,则监控层向调度层输入调度策略的变更信号和新的变量值,调度层将不再按原来的调度依据(如某过程的完成时间)进行调度。给出在新的调度依据下的调度方案后,调度层的输出通过监控层成为被控对象新的输入信号。

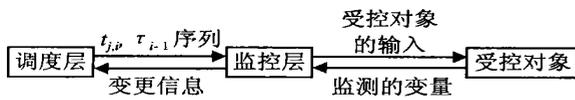


图 1 调度与监控的集成结构

在滚动式调度与监控的集成中,首先要解决的问题是,应该监测被控对象中的哪些变量,或者说,对象中的哪个变量(如反应时间 h) 对目标函数 J 的导数 $(\partial J / \partial h)$ 的绝对值比较大,其值大的就应该是监测中要测量的变量;其次要解决的问题是,在调度中应该定义什么样的滚动窗口;再次解决的问题是,有一些变量是无法测量的,那么对它们应该怎么办。

对于第 1 个问题,由文献[3]可知, $t_{j,i}$ 和 τ_{-1} 实际上都是某系统变量(如反应时间 h) 的函数,而 h 又是时变的,即它是 t 的函数,则有

$$t_{j,i} = t_{j,i}(h(t)), \quad \tau_{-1} = \tau_{-1}(h(t))$$

代入 pv, mv, sv 中,有

$$pv = pv(\tau_{-1}(h(t)), t_{j,i}(h(t))) = \xi(\tau_{-1}(h(t)), t_{j,i}(h(t))) C_p$$

$$mv = mv(\tau_{-1}(h(t)), t_{j,i}(h(t))) = M(\tau_{-1}(h(t)), t_{j,i}(h(t))) C_m$$

$$sv = sv(\tau_{-1}(h(t)), \tau(h(t)), t_{j,i}(h(t)), t_{j1,i+1}(h(t))) =$$

$$S(\tau_{-1}(h(t)))(\tau(h(t)) - \tau_{-1}(h(t))) C_s$$

故有

$$\frac{\partial J}{\partial h} = \frac{\partial pv}{\partial h(t)} - \frac{\partial mv}{\partial h(t)} - \frac{\partial sv}{\partial h(t)}$$

所以,求 $\partial J / \partial h$ 就是求 $\sum \partial pv / \partial h(t)$, $\sum \partial mv / \partial h(t)$ 和 $\sum \partial sv / \partial h(t)$ 。但在实际上, pv, mv, sv 的表达式很难给出。所以对于第 1 个问题,一般是由经验给出对调度影响大的变量。或者在静态调度仿真中适

当变化某 h , 观察调度的结果是否变化较大。

对于第 2 个问题,滚动方法的滚动窗口的选择对调度结果有很大影响。从前述可以看出,监控的目的是当某变量的变化超过一个范围时调整调度策略。于是可以每一次变量的超出范围为一个滚动窗口的开始,以下一次变量的超出范围为本滚动窗口的结束。其间采用文献[3]中提出的调度算法。相邻的第 i 和 $i+1$ 窗口, i 窗口的末状态作为 $i+1$ 窗口的初状态。

设各滚动窗口分别有 $n_1, n_2 - n_1, n_3 - n_2, \dots, n_n - n_{n-1}$ 个变迁激发,且 $n_1 < n_2 < \dots < n_n, n_i$ 表示第 i 个窗口结束时,从第 1 个到第 i 个窗口共激发了 n_i 个变迁,则有

$$n_i \quad \{k | k \in Z^+, C_r(h_1(\tau_{-1} - \epsilon)) = 1 \text{ 或}$$

$$C_r(h_2(\tau_{-1} - \epsilon)) = 1, \dots, \text{且 } m_p(\tau_{-1}) = 0\}$$

其中, $m_p(t)$ 为过程库所 p 在 t 时刻的标识, $h_i(t)$ 为对目标函数影响大的变量, $C_r(h_i(t))$ 为监测判断函数,当 $C_r(h_i(t)) = 1$ 时,说明变量的变化已超出范围,将对调度结果产生大的影响。这时就表明了下一个滚动窗口的即将出现和本窗口的即将结束。

调度模型为

$$J = \max_{\substack{n \\ k=2 \\ i=n_{k-1} \\ t_j, t_{j1}}}^{nk} [pv(\tau_{-1}, t_{j,i}) - mv(\tau_{-1}, t_{j,i}) - sv(\tau_{-1}, \tau, t_{j,i}, t_{j1,i+1})] \quad (1)$$

$S(\tau_i + \epsilon) = S(\tau_i), S(t)$ 为 t 时刻受控对象的状态, ϵ 为一个很小的正数。在 (τ_i, τ_{i+1}) 之间的约束与文献[3]中类似。

对于第 3 个问题,某些变量无法直接得到,例如在文献[3]中对调度影响大的过程库所的完成反应所用时间 h , 在反应正在进行时该值是无法测量的。但对于某过程库所,其 h 是和该库所的状态方程 $x = f(x, u(t), t)$ 有关的,若能给出 x 的解析解 $x = x(t)$, 设该库所开始其过程时状态变量值为 x_0 , 结束时为 x_1 , 分别令 $x(t) = x_0, x(t) = x_1$, 计算出两式的解 t_0 和 t_1 , 则该库所的 $h = t_0 - t_1$ 。但对许多的状态方程 $x = f(x, u(t), t)$, 是无法给出其解析解的,这时可以用数值解法得出 t_0, t_1 和 h 。

设某一过程库所的初始状态方程为

$$\begin{cases} \dot{x}_i = f_i(x_i, u(t), t) \\ y_i = g_i(x_i, u(t), t) \end{cases} \quad (2)$$

被控对象开始运行后,各过程库所开始相应的过程演化,此时我们不断地采样当前的输入 u 和输出 y , 便可以在线地对该库所的状态方程进行辨

识, 即有

$$\begin{cases} \dot{x}_i = f_i(x_i, u(t), t) \\ y_i = g_i(x_i, u(t), t) \end{cases} \quad (3)$$

同时, 依上述计算得到该库所对应变量 h_i 的值, 有 $h_i = h_i(t)$ 。对于其它过程库所也作同样处理, 这样便可得到 $h_1(t), h_2(t), \dots$, 当满足 $\{h_1 | Cr(h_1(t)) = 1\} \{h_2 | Cr(h_2(t)) = 1\} \dots \emptyset$ 时, 不再按原来的激发序列激发输入变迁, 只用激发输出变迁, 并重新计算在新的 h_1, h_2, \dots 下的激发变迁序列。

3 算 法

首先说明一个概念:

不可中断变迁序列 NIT: 如果对于某一关键库所激活它经过的变迁序列依次为: t_1, t_2, \dots, P, t_i 与 t_{i+1} 之间的库所为 $P_{i,i+1}$, 若 $P_{1,2}, P_{2,3}$ 都为过程库所, 则 t_1, t_2, \dots, P 为不可中断变迁序列 NIT; 若有 t_i, t_{i+1}, \dots, t_m 是 t_1, t_2, t_3, \dots, P 的一部分, 则 t_i, t_{i+1}, \dots, t_m 也是不可中断变迁序列, NIT 各变迁之间的库所序列为 P_{NIT} 。

算法:

Step1: 初始化, 用文献[3]中算法计算出在初始状态下的激发序列 A ;

Step2: 开始一个滚动窗口, 依次激发 A 中的各个变迁, 同时对各过程库所进行采样和辨识, 并通过仿真得到各过程库所对调度影响大的变量 h_1, h_2, \dots 的值。若在 t 时刻有 $\{h_1 | Cr(h_1(t)) = 1\} \{h_2 | Cr(h_2(t)) = 1\} \dots \emptyset$, 则设此时正在进行的过程库所集为 P_r , 若 $P_r \cap P_{NIT} = \emptyset$, 则对于不可中断变迁序列, 应完成其对应的各过程; 对于其它的只用激发输出变迁。若 $P_r \cap P_{NIT} \neq \emptyset$, 则只用激发 P_r 集的输出变迁集。设最后激发变迁的激发时刻为 t_1 , 计算 t_1 时刻状态, 这时过程库所中没有标识。

Step3: 在 t_1 时刻的状态下按文献[3]中算法再次算出以该状态为初始态的激发变迁序列 B , 在 t_1 时刻, 结束上一个滚动窗口, 开始下一个滚动窗口, 按序列 B 激发各变迁, 转 Step2。

4 仿真实例

采用文献[3]中的仿真实例。

某一化工生产过程: 物质 A 先加热 1 h, B 加热 1.2 h, 然后 A 与 B 以 1:1 的比例进行反应 1, 用 2 h 产生比例为 3:7 的产物 E 与 C , 其中 E 为第 1 个最终产品; 另一方面, C 与 D 以 1:1 的比例进行反应 2, 用

1.4 h 产生比例为 8:2 的产物 F 与 A , 其中 F 为第 2 个最终产品。另外, B 也可加热后进行碾碎和 G 以 1:1 的比例进行反应 3, 用 2 h 产生第 3 个最终产品 H 。设备有两个加热器 Heater1 和 Heater2, 两个相同的反应器 Reactor1 和 Reactor2 以及一个碾碎器。该问题的 CT CH 模型如图 2 所示。

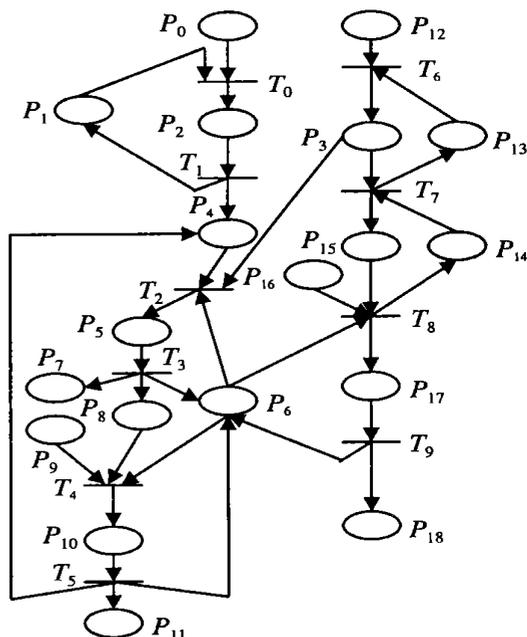


图 2 实例系统的 Petri 网模型

由于以上关于反应时间的各项数据都是经验值, 所以在调度过程中有必要引入监控。首先, 适当变化 P_5 的完成反应所需时间(以下用 h_{p5} 表示), 在第 20 个时刻, 当 $h_{p5} = 1.6$ 时, 调度所得为 26 975 个价格单位; 当 $h_{p5} = 1.8$ 时, 调度所得为 33 240 个价格单位。可以看出, h_{p5} 对调度结果影响较大。 P_5 库所对应的准确的差分方程为

$$y(j) = 2.0209y(j-1) - 2.0410y(j-2) + 1.0101y(j-3) + 0.1202u(j-1) - 0.0972u(j-2) + 0.0516u(j-3) \quad (4)$$

可以算出, 对应该方程, h_{p5} 为 2.5 个时间单位。另外, 给 P_5 库所一个不准确的差分方程

$$y(j) = 2y(j-1) - 2y(j-2) + y(j-3) - 0.1952u(j-1) + 0.171u(j-2) + 0.0678u(j-3) \quad (5)$$

与该不准确方程对应的 h_{p5} 为 2.0 个时间单位。上述两式中, $u(j)$ 为 $j \times T_s$ 时刻的输入, $y(j)$ 为 $j \times T_s$ 时刻的输出。在引入了监控的仿真实验中, 将式 (5) 作为 P_5 库所的初始状态方程, 通过不断地对 P_5

库所的实际输出和输入采样进行辨识,修正式(5),计算出与之对应的 h_{p5} 。其中,对 P_5 库所的实际输出的采样是将输入代入式(4)得到。当其变化较大时,结束上一个滚动窗口,开始下一个滚动窗口。

仿真结果如图 3 ~ 图 6 所示。其中图 3 为不加监控时调度的 Gant 图,且 P_5 库所对应的差分方程为方程(5),其 h_{p5} 一直都为 2.0;图 4 为在调度中加入了监控的 Gant 图,仿真实验中,在 $t = 6.4$ 时刻,其对应的 h_{p5} 由 2.0 变成 2.3,在 $t = 14$ 时刻, h_{p5} 又由 2.3 变成 2.5;图 5 为对应于准确方程(4)调度的 Gant 图;图 6 为 20 个时间单位内的利润变化图,其中点划线对应图 3,点线对应图 4,实线对应图 5。

图 6 中,在 $time = 20$ 时刻,与图 3,图 4 及图 5 分别对应的调度所得利润为 22 615.9, 26 006.9 和 28 422.2 个价格单位,可以看出,加入监控后的调度结果比未加监控时要好得多。虽然图 4 的调度结

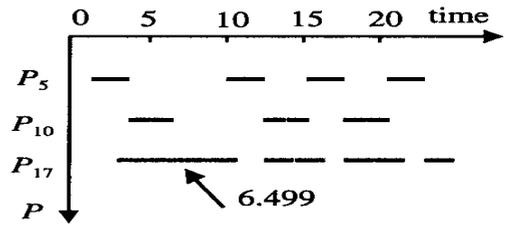


图 5 对应于准确模型的 Gant 图

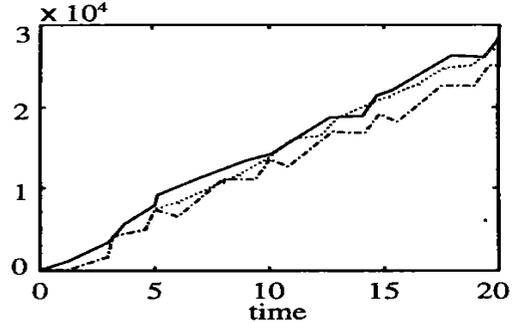


图 6 利润变化图

果比图 5 的结果要差一些,但比较两者可以发现,图 4 中在 $time = 25.301$ 以后的激发序列和图 5 在 $time = 6.499$ 以后的激发序列相同,这说明加入监控后其调度结果向图 5 的结果靠近。

由以上分析可知本文算法是有效的。通过在线辨识不断修正调度所用的模型,改变调度策略,从而达到了监控与调度集成的目的。

参考文献(References):

[1] 王朝晖,陈浩勋,甘文泉,等. Multipurpose 批处理过程短期调度的滚动时域方法[J]. 控制理论与应用(Control Theory & Applications), 1998, 15(4): 567-574.
 [2] 方剑,席裕庚. 基于遗传算法的滚动调度策略[J]. 控制理论与应用(Control Theory & Applications), 1997, 14(4): 589-594.
 [3] 刘斌,曾建潮. 一种基于 CTCH 网的动态调度算法[J]. 自动化学报(Acta Automatica Sinica), 2000, 26(增刊 B): 120-124.

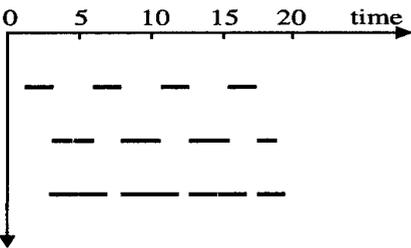


图 3 未引入监控的 Gant 图

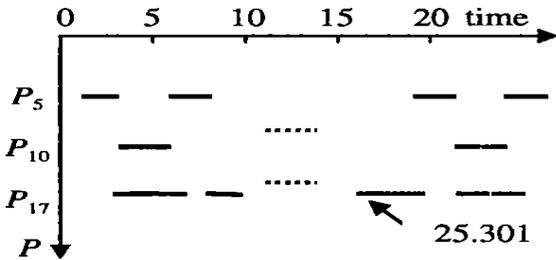


图 4 引入监控的 Gant 图

(上接第 352 页)

[3] 赵明旺. 状态反馈同时镇定的非线性方程数值解[J]. 控制理论与应用(Control Theory & Applications), 1996, 13(5): 667-671.
 [4] 王耀青. 同阶单输入控制系统的同时镇定问题[J]. 武汉科技大学学报(J of Wuhan Univ of Science & Technology), 1999, 22(2): 175-178.

[5] 王耀青. 同时稳定 r 个单输入系统控制问题的最小二乘解法[J]. 信息与控制(Information and Control), 2000, 29(7): 657-660.
 [6] 王耀青. LQ 逆问题解的一种有效算法[J]. 控制理论与应用(Control Theory & Applications), 1992, 9(1): 9-15.