Фуругян М.Г.

Планирование работ в многопроцессорной АСУ реального времени в условиях неопределенности

Аннотация: Рассматривается задача планирования работ без прерываний и переключений в многопроцессорной АСУ для случая, когда на множестве работ задано отношение предшествования, имеется общий директивный срок для всех работ и, кроме того, задается распределение заданий по процессорам. В неопределенные моменты времени могут поступать запросы на выполнение более приоритетных работ, для которых на фиксированное время освобождается часть процессоров. В результате этого выполнение исходной совокупности заданий переносится на более позднее время и тем самым нарушается построенное ранее расписание. Разработана такая стратегия построения допустимого расписания, при которой вероятность его нарушения вследствие поступления запросов на выполнение дополнительных работ минимальна.

Ключевые слова: многопроцессорная система, директивный интервал, допустимое расписание, антагонистическая игра

1. Введение

При разработке математического и программного обеспечения для многопроцессорных автоматизированных систем управления (АСУ), в частности, систем реального времени, одной из основных задач является задача планирования вычислений и построения допустимых расписаний выполнения программных модулей. При испытаниях и эксплуатации сложных технических объектов нередко возникает нештатная ситуация, когда помимо выполнения основных программных модулей могут поступать запросы на выполнение дополнительных, но более приоритетных заданий. Если такие запросы поступают в то время, когда выполняются основные работы и при этом требуется освободить занимаемые ими процессоры, то это приводит к нарушению построенного ранее расписания для основных работ и целью в этом случае является минимизации вероятности подобного рада нарушений. Указанные ситуации возникают при испытаниях самолетов, космических систем, ядерных реакторов и другой сложной техники. При этом возникают задачи составления допустимого расписания при наличии неопределенных факторов, которые сводятся к игровым задачам.

В [1] рассматривалась задача распределения не возобновляемых ресурсов при наличии неопределенных факторов и для ее решения была использована методика построения многогранников устойчивости, разработанная в [2]. Рассмотренная в [1] задача является *NP*-трудной и предложенный алгоритм ее решения имеет переборный характер. В настоящей статье предполагается, что каждая работа закреплена за определенным процессором, а на множестве работ задано отношение частичного порядка их выполнения. Благодаря этому множество допустимых расписаний описывается с помощью полиномиального алгоритма. С помощью этого множества строится антагонистическая игра, решение которой определяет решение исходной задачи.

2. Постановка задачи

Рассматривается АСУ, состоящая из m процессоров, и совокупность непрерываемых работ (заданий), подлежащих выполнению, $W=\{w_{ij}: i=\overline{1,k_j},\ j=\overline{1,m}\}$, k_j – заданные величины. Работа w_{ij} приписана процессору j и ее длительность составляет t_{ij}. Задан общий для всех работ директивный интервал $[0,\ T]$. На множестве W задан ориентированный граф без циклов $G=(W,\ A)$, определяющий частичный порядок выполнения работ, где W – множество узлов, A – множество ориентированных дуг. Если $(w,\overline{w})\in A,$ то работа $\overline{w}$ может быть начата только после завершения выполнения работы w $(w\to\overline{w})$. Задания, приписанные процессору j, упорядочены следующим образом: $w_{1j}\to w_{2j}\to\ldots\to w_{k_j j}$. Кроме того, каждая работа может иметь непосредственных предшественников, приписанных другим процессорам.

В некоторые неопределенные моменты времени $y_{hj}\in[0,T]$, $0\le y_{1j}<y_{2j}<\ldots<y_{p_j j}\le T,\ h=\overline{1,p_j},\ j=\overline{1,m},\ p_j$ – заданные

величины, могут поступать запросы на выполнение дополнительных, более приоритетных, непрерываемых работ $V=\{v_{hj}: h=\overline{1,p_j}, j=\overline{1,m}\}$. Работа v_{hj} приписана процессору j и ее длительность составляет s_{hj}, т.е. она выполняется в интервале $[y_{hj},\ y_{hj}+s_{hj}]$. Если в этом интервале должна выполняться некоторая работа w_{ij}, то ее выполнение прекращается и переносится на более позднее время. Тем самым нарушается построенное ранее расписание выполнения работ W. Процесс выполнения множества работ W и поступления запросов на выполнение дополнительных работ V повторяется многократно. Задача заключается в выработке стратегии построения таких расписаний выполнения работ W, которые удовлетворяют следующим условиям: 1) каждая работа w_{ij} выполняется процессором j; 2) не нарушаются ограничения частичного порядка выполнения работ W, задаваемые графом G; 3) если в интервале $[0,\ T]$ не поступило ни одного запроса на выполнение дополнительных работ W, то все работы W завершаются не позднее момента времени T; 4) вероятность того, что расписание работ W будет нарушено вследствие поступления запросов на выполнение дополнительных работ V, минимальна.

3. Алгоритм решения задачи

Вершины W графа G стандартным образом разбиваются на уровни $W_l,\ l=1,\ldots,L$. Для работы $w_{ij}\in W$ символами $\tau(w_{ij})=\tau_{ij}$ и $\theta(w_{ij})=\theta_{ij}$ будем обозначать соответственно наиболее ранний возможный и наиболее поздний допустимый сроки начала выполнения. Пусть x_{ij} – момент начала выполнения работы w_{ij}, причем выполняются условия 1 – 3. Множество $X=\{x_{ij}: i=\overline{1,k_j}, j=\overline{1,m}\}$ описывает все возможные моменты начала выполнения работ W, при которых существует допустимое расписание, и строится следующим образом. Для $w_{ij} \in W_0$ $x_{ij} \in [0;\ \theta_{ij}]$. Для $l = \overline{1, L}$, $w_{ij} \in W_l$, $x_{ij} \in [\max(x_{pq} + t_{pq});\ \theta_{ij}]$. Каждый элемент $x \in X$ определяет допустимое расписание S_x, при котором момент начала выполнения работы w_{ij} равен x_{ij}. Множество Y описывает все возможные моменты поступления запросов на выполнение дополнительных работ V: $Y = \{y_{hj} : h = 1,\ldots p_j,\ j = \overline{1, m} : 0 \le y_{1j} < y_{2j} < \ldots < y_{p_j j} \le T\}$. На множестве $X \times Y$ будем рассматривать антагонистическую игру с платежной функцией $K(x, y)$, $x \in X,\ y \in Y$, определенной следующим образом:

$$K(x, y) = \begin{cases} 1, \text{ если } x_{ij} + t_{ij} < y_{hj} \text{ или } x_{ij} > y_{hj} + s_{ij} \text{ при всех} \\ j = \overline{1, m},\ i = \overline{1, k_j},\ h = \overline{1, p_j}; \\ 0 \quad \text{в противном случае.} \end{cases} \tag{1}$$

Если для каждого $j = \overline{1, m}$ выполнение ни одной из дополнительных работ v_{kj} не препятствует выполнению ни одной из работ w_{ij}, т.е. если интервалы $[x_{ij},\ x_{ij} + t_{ij}]$ и $[y_{hj},\ y_{hj} + s_{hj}]$ не пересекаются, то $K(x, y) = 1$. Если же для некоторого j, $1 \le j \le m$, выполнение хотя бы одной работы v_{kj} препятствует выполнению хотя бы одной работы w_{ij}, то $K(x, y) = 0$. Пусть $\nu(x)$ − некоторая смешанная стратегия первого игрока в игре с платежной функцией $K(x, y),\ x \in X,\ y \in Y$. Тогда при фиксированном $y \in Y$ величина $\int_X K(x, y)\,d\nu(x)$ равна вероятности того, что расписание работ W не будет нарушено вследствие поступления запросов на выполнение дополнительных работ V. Оптимальная смешанная стратегия $\nu^*(x)$ первого игрока максимизирует величину $\inf_{y \in Y} \int_X K(x, y)\,d\nu(x)$ и определяет искомую стратегию построения расписаний S_x. Метод решения антагонистической игры (1), основанный на аппроксимации ее конечными играми, описан в [4].

Литература:

1. *Фуругян М.Г.* Решение одной задачи распределения ресурсов в АСУ реального времени при наличии неопределенных факторов // Автоматика и телемеханика. — 2002. — № 11. — С. 167-171.

2. *Мищенко А.В., Сушков Б.Г.* Минимизация времени выполнения работ, представленных сетевой моделью, при нефиксированных параметрах сети. – М.: ВЦ АН СССР, 1980. – 25 с.

3. *Филлипс Д., Гарсиа-Диас А.* Методы анализа сетей. – М.: Мир. — 1984. – 496 с.

4. *Фуругян М.Г.* Приближенное решение одного класса бесконечных антагонистических игр с полунепрерывной платежной функцией // Вестн. МГУ. Сер.15. — 1980. — № 2. —С. 66- 69.