

УДК 681.34

ОПТИМИЗАЦИЯ СЕТЕВОГО ПЛАНА МЕТОДОМ СЛУЧАЙНОГО ПОИСКА С ПЕРЕСЧЕТОМ С ПЕРЕМЕННОЙ ВЕЛИЧИНОЙ ШАГА

Черниговский А.С., Царев Р.Ю.

ФГАОУ ВПО «Сибирский федеральный университет», Красноярск, Россия (660074, Красноярск, ул. Академика Киренского, 26, УЛК), e-mail: tsarev.sfu@mail.ru

В статье рассматривается подход к формированию оптимального плана выполнения задач информационно-управляющей системы на основе сетевого моделирования. Разработана и описана математическая модель формирования плана выполнения задач информационно-управляющей системы с учетом имеющихся ресурсов, сроков выполнения и трудоемкости как отдельных задач, так и всего проекта в целом. Представлена многоэтапная процедура получения оптимального плана задач, выполняемых в системе, который дает оптимальную загрузку всех видов ресурсов на всем протяжении работы информационно-управляющей системы, а также позволяет минимизировать сроки выполнения проектов. Предложенная процедура реализует метод случайного поиска с пересчетом с переменной величиной шага, осуществляя направленный поиск как при достижении допустимой области поиска, определяемой заданными ограничениями, так и непосредственно в допустимой области.

Ключевые слова: сетевая модель, планирование, метод случайного поиска

OPTIMIZATION OF A NET SCHEDULE BY RANDOM SEARCH METHOD WITH RECALCULATION WITH VARIABLE STEP SIZE

Chernigovskiy A.S., Tsarev R.Y.

Siberian Federal University, Krasnoyarsk, Russia (660074, Krasnoyarsk, street A. Kirenskogo, 26, ULK), e-mail: tsarev.sfu@mail.ru

The article considers the approach to formation of an optimal task schedule for information control system based on network simulation. The mathematical model of formation of the task execution schedule for information control system taking into account available resources, periods of execution and complexity of both separate tasks, and the project in general is developed and described. Multi-stage procedure of formation of the optimal schedule of tasks executed in system is presented. The schedule gives an optimal loading of all types of resources throughout information control system activity, and also allows minimizing periods of execution of the project. Presented procedure based on a random search method with recalculation with variable step size realizes directional search as in case of achievement of the acceptable search region determined by the given limitations, and directly in acceptable region.

Keywords: net model, scheduling, random search method.

Широкое применение алгоритмов параллельной обработки информации в современных информационно-управляющих системах привело к открытию новой области исследований, связанной с проблемами эффективного формирования параллельных процессов в системах данного класса. Между реализацией алгоритмов на классической однопроцессорной архитектуре ЭВМ и реализацией тех же алгоритмов на параллельных вычислительных системах существуют принципиальные различия [1, 7]. В [8] показано, что, чем выше быстродействие параллельной вычислительной системы, достигаемое за счет архитектурных решений, тем более ограничен класс управляющих алгоритмов информационно-управляющих систем, которые на ней могут быть эффективно реализованы.

Эффективное формирование параллельных процессов программного обеспечения является сложной, но крайне важной и актуальной на сегодняшний день задачей [9]. На ее

решение существенно влияют индивидуальные особенности параллельных систем. Несомненно, такие задачи можно решать для каждой информационно-управляющей системы в отдельности, но уже сейчас существует большое количество различных методов и средств, применяемых в информационно-управляющих системах, и создание конкретного программного обеспечения для каждой системы является неприемлемым. Поэтому эффективное формирование параллельных процессов должно начинаться с построения математической модели работы большого числа узлов с изменяемыми (в общем случае) связями и с анализа на основе данной модели особенностей информационных потоков, проходящих через всю совокупность узлов [6]. Это позволит разработать методы и алгоритмы формирования параллельных процессов в информационно-управляющих системах, а также предложить методы их оптимизации [4].

Одним из способов описания процессов в информационно-управляющей системе является представление структуры выполнения задач системы с помощью сетевой модели [3]. Математический аппарат сетевого моделирования позволяет описать процессы, протекающие в системе, и произвести оптимизацию с целью получения такого плана выполнения работ системы, который бы позволял рационально использовать ресурсы системы и минимизировать время выполнения всего проекта [2, 5].

Далее будет рассмотрена задача формирования оптимального расписания выполняемых операций в информационно-управляющей системе, объединенных общими ограничениями на ресурсы.

Модель формирования оптимального плана

Положим, что имеется план, который содержит множество логически взаимосвязанных между собой операций (задач) V . В информационно-управляющей системе присутствует L видов ресурсов, все они используются для выполнения данного плана задач.

Операции (задачи) обозначаются как (i, j) , где i – начальное, а j – конечное событие задачи. Каждая операция может выполняться только одним определенным типом ресурса.

Работы, выполняемые одним видом ресурса, выделяются во множество V^l ($\bigcup_{l=1}^L V^l = V$).

Далее, для каждой операции (i, j) обозначим: $t(i, j)$ – продолжительность ее выполнения, $t_{рн}(i, j)$ – наиболее ранний момент начала операции, $t_{по}(i, j)$ – наиболее поздний момент окончания операции. Известны директивные сроки начала D_n и окончания D_o проекта. Для корректности задачи необходимо выполнение условия $K \leq D_o - D_n$, где K – длина критического пути плана.

Планируемый период $[T_n, T_o]$ (T_n и T_o совпадают с директивными сроками) разбивается на f интервалов, каждый продолжительностью τ_k ($k = \overline{1, f}$). Задается наличие

количество используемых ресурсов для каждого вида S_i^k на k -м интервале, суммарная трудоемкость выражается как $W_i^k = S_i^k \tau_k$.

Примем следующие обозначения: $t_n(i, j)$ и $t_o(i, j)$ – моменты времени начала и окончания работ, $w(i, j)^k$ – трудоемкость, $c(i, j)^k$ – полезные трудозатраты в стоимостном выражении. Данные характеристики относятся к той части продолжительности работы $t(i, j)$, которая покрывается результатом τ_k . В связи с тем, что в произвольный момент времени состояние процесса формирования характеризуется значениями $t_n(i, j)$ и $t_o(i, j)$, $\forall(i, j)$, эти значения можно рассматривать как фазовые координаты объекта управления.

Получаем задачу нахождения таких значений $t_n(i, j)$ и $t_o(i, j)$, $\forall(i, j)$, которые позволяли бы достичь с учетом всех ограничений номинальную и, по возможности, равномерную загрузку всех видов ресурсов на протяжении всего планируемого периода.

Представленную выше задачу формализуем следующим образом.

Необходимо, чтобы движение объекта в фазовом пространстве удовлетворяло ограничениям на сроки выполнения проекта:

$$\max_{(i,j) \in V} (t_n(i, j) + t(i, j)) \leq D_o, \quad \min_{(i,j) \in V} t_n(i, j) \geq D_o, \quad (1)$$

и ограничениям на порядок выполнения задач, которые заданы топологией сетевой модели (комплексом информационно и по управлению взаимосвязанных задач информационно-управляющей системы):

$$G(J, V), \quad (2)$$

где J – множество вершин.

Положим R^k – подмножество задач, которые покрываются k -м интервалом времени длительностью τ_k , $k = \overline{1, f}$, $R^k = \bigcup_{l=1}^L R^l$.

Данная задача представляет собой задачу сетевого планирования на подмножестве R^l с учетом (1) и (2), которую можно представить в виде целочисленной модели:

$$\sum_{(i,j) \in V^l} \varphi_k(l) c(i, j)^k x(i, j)^k, \quad (3)$$

$$\sum_{(i,j) \in V^l} w(i, j)^k x(i, j)^k \leq w_i^k, l = \overline{1, L}, \quad (4)$$

где

$$x(i, j)^k = \begin{cases} 1, & \text{если } (i, j) \text{ выполняется на } k \text{ - м интервале,} \\ 0, & \text{в противном случае.} \end{cases}$$

Здесь $\varphi_k(l)$ – функция штрафа, далее о ней будет сказано подробнее.

Чтобы достичь оптимального управления на всем планируемом периоде $[T_n, T_0]$, необходимо также достичь оптимальности управления на каждом интервале τ_k . Получаем функцию, характеризующую эффективность управления, следующего вида:

$$F = \sum_{k=1}^f \sum_{l=1}^L \sum_{(i,j)^k \in V^l} \varphi_k(l) c(i,j)^k x(i,j)^k. \quad (5)$$

Задача (1)-(5) полностью охватывает специфику процессов информационно-управляющих систем, а также их особенности.

В результате, максимизируя функционал (5), можно получить минимум себестоимости выполнения всех операций.

Для решения описанной выше задачи предлагается использовать метод случайного поиска с пересчетом с переменной величиной шага. Данный метод осуществляет направленный поиск как при достижении допустимой области поиска, определяемой заданными ограничениями, так и непосредственно в допустимой области.

Перейдем к поэтапному описанию процедуры решения задачи.

Многоэтапная процедура оптимизации сетевой модели

Этап I. На основе топологии сети (2) и известных $t_{рн}(i, j)$ и $t_{по}(i, j)$ производим расчет длины критического пути – k , а также резервов времени всех работ:

$$P(i) = t_{по}(i, j) - t_{рн}(i, j) + D_0 - D_n - k.$$

Берем величину шага $\Pi_0 = \frac{1}{2} \max P(i)$. Максимум вычисляется по всем задачам плана.

Этап II. Положим начальную точку поиска T_0 (положение начал задач) по следующему правилу:

$$t_n^0(i, j) = t_{рн}(i, j) + \frac{1}{2} P(i), \forall (i, j).$$

Этап III. В полученной на предыдущем шаге точке проверяем, выполняется ли условие (4) для каждого k -го интервала $k = \overline{1, f}$. Это необходимо, чтобы проверить, не превышает ли ожидаемая потребная трудоемкость наличную трудоемкость по каждому виду ресурса ($l = \overline{1, L}$):

$$\delta_k = \begin{cases} 0, \text{ если } \sum_{(i,j) \in V^l} w(i,j)^k x(i,j) \leq w_l^k \forall (i,j), l = \overline{1, L}, \\ 1, \text{ в противном случае} \end{cases}$$

Этап IV. На данном этапе выясним, на каких интервалах условие (4) не выполняется.

Определим

$$\Delta = \sum_{k=1}^f \delta_r.$$

Целочисленная функция Δ представляет собой функцию качества при достижении допустимой области. Если значение Δ , полученное в новой точке, меньше, чем в предыдущей, значит, шаг является подходящим.

При $\Delta > 0$ переходим к этапу VIII, поскольку допустимая область уже достигнута; если же $\Delta = 0$, то переходим к вычислению функции качества (5), так как найденная точка находится внутри допустимой области.

Этап V. В функционале (5) $\varphi_k(l)$ – функция штрафа за невыполнение условия:

$$s_l^k - \Delta_2 \leq \sum_{(i,j) \in V^l} n(i,j)^k x(i,j)^k \leq s_l^k + \Delta_1, \quad l = \overline{1, L}, \quad k = \overline{1, f}. \quad (6)$$

Это условие устанавливает некоторые допуски Δ_1 и Δ_2 для наличного ресурса l -го вида в произвольный момент.

Функция $\varphi_k(l)$ определяется по формуле:

$$\varphi_k(l) = 1 - \left[b_l' \sum_{t_i \in \tau_k} \left\{ s_l(t_i) - \tilde{s}(t_i) + \tau_k \right\} + b_l'' \sum_{t_i \in \tau_k} \left\{ \tilde{s}(t_i) - s_l(t_i) - \tau_k \right\} \right] / I_l,$$

где t_i – моменты времени, в которые происходит изменение потребного количества ресурса ($t_i = t_n(i, j)$ или $t_i = t_n(i, j) + t(i, j)$); $\tilde{s}(t_i)$ – суммарное количество l -го вида ресурса, требуемого в момент времени t_i ; \sum' – сумма по тем t_i , в которых происходит нарушение условия (6) справа (перегрузка); b_l' и b_l'' – коэффициенты штрафа за недогрузку и перегрузку, соответственно, I_l – число моментов t_i на k -м интервале – τ_k .

Запоминаем значение $F_0 = F(T_0)$ и переходим этапу VI. Если $\eta \geq 1$ (η – номер шага поиска), то переходим к этапу X.

Этап VI. На данном этапе моделируется M -мерный единичный случайный вектор $\bar{\xi}$. Данный вектор является равномерно распределенным по всем направлениям фазового пространства. Обозначим координату этого вектора как $\xi(i, j)$, она соответствует работе (i, j) .

Этап VII. Определим переход из точки T_η в точку $T_{\eta+1}$ по формулам:

$$\hat{t}_{\text{H}}^{\eta+1}(i, j) = \begin{cases} t_{\text{H}}^{\eta}(i, j) + \text{Ш}_\eta \xi(i, j), & \text{если } t_{\text{PH}}(i, j) \leq t_{\text{H}}^{\eta}(i, j) + \text{Ш}_\eta \xi(i, j) \leq t_{\text{HO}}(i, j); \\ t_{\text{H}}^{\eta}(i, j), & \text{если } t_{\text{H}}(i, j) + \text{Ш}_\eta \xi(i, j) < t_{\text{PH}}^{\eta}(i, j) \\ & \text{или } t_{\text{H}}(i, j) + \text{Ш}_\eta \xi(i, j) < t_{\text{HO}}^{\eta}(i, j). \end{cases}$$

$$t_{\text{H}}^{\eta+1}(i, j) = \begin{cases} \hat{t}_{\text{H}}^{\eta+1}(i, j), & \text{если } \hat{t}_{\text{H}}^{\eta+1}(i, j) \leq \max_{(i,j) \in V} (t_{\text{PH}}(i, j) + t(i, j)); \\ \max_{(i,j) \in V} (t_{\text{PH}}(i, j) + t(i, j)), & \forall (i, j), \text{ в противном случае.} \end{cases}$$

Для осуществления возврата в точку T_η необходимо вычислить величины

$$\delta(i, j) = t_{\text{н}}^{\eta+1}(i, j) - t_{\text{н}}^{\eta}(i, j), \forall(i, j),$$

которые хранятся в памяти до следующего шага. Переходим к этапу III.

Этап VIII. Функция Δ вычисляется в первый раз в начале поиска или же после вычисления функции качества (5) (алгоритм работал в допустимой зоне). В этом случае, когда Δ вычислена впервые, присваиваем Δ_0 значение Δ ($\Delta_0 = \Delta$) и переходим к этапу VI. Иначе рассчитываем новую величину шага по следующей формуле:

$$\text{Ш}_{\eta} = \begin{cases} \text{Ш}_{\eta-1}, & \text{если } \Delta < \Delta_0 \text{ и } Y(\Delta) < \delta(\Delta) \text{ или } \Delta \geq \Delta_0 \text{ и } H(\Delta) < \chi(\delta); \\ \alpha(\Delta)\text{Ш}_{\eta-1}, & \text{если } \Delta < \Delta_0 \text{ и } Y(\Delta) = \delta(\Delta); \\ \beta(\Delta)\text{Ш}_{\eta-1}, & \text{если } \Delta \geq \Delta_0, H(\Delta) = \chi(\Delta), \text{Ш}_{\eta-1} \geq \varepsilon(\Delta); \\ \text{Ш}_0, & \text{если } \Delta \geq \Delta_0, H(\Delta) = \chi(\delta), \beta\text{Ш}_{\eta-1} < \varepsilon(\Delta) < \lambda(\Delta). \end{cases} \quad (7)$$

Здесь $Y(\Delta)$ – число удачных подряд шагов, $\delta(\Delta)$ – целое, заранее известное число, определяющее количество удачных подряд шагов для увеличения величины шага, $\alpha(\Delta)$ – коэффициент увеличения шага, $\alpha(\Delta) > 1$; $H(\Delta)$ – число неудачных подряд шагов, $\chi(\delta)$ – целое, заранее известное число, которое определяет число неудачных шагов для уменьшения шага, $\beta(\Delta)$ – коэффициент уменьшения шага, $\beta(\Delta) > 1$; $\varepsilon(\Delta)$ – заданная точность, делать шаг меньше которой не имеет смысла; целое число $\rho(\Delta)$ определяет, сколько раз шаг уменьшался до заданной точности, если $\rho(\Delta) = \lambda(\Delta)$, которое задается заранее, то поиск прекращается, и точка T_{η} принимается за оптимальную.

В случае, когда был получен удачный шаг ($\Delta < \Delta_0$), значение $H(\Delta)$ необходимо обнулить и перейти к этапу VI; иначе, когда шаг получен неудачный ($\Delta \geq \Delta_0$), необходимо обнулить $Y(\Delta)$ и перейти к этапу IX.

При попадании в допустимую область поиска (этап V) обнуляется как $Y(\Delta)$, так и $H(\Delta)$. Для того чтобы достичь наибольшей надежности поиска, необходимо обнулять и $\rho(\Delta)$.

Этап IX. Здесь происходит возврат в исходную точку по следующей формуле:

$$t_{\text{н}}^{\eta+2}(i, j) = t_{\text{н}}^{\eta+1}(i, j) - \delta(i, j), \forall(i, j)$$

и переход на этап VI.

Этап X. После того как было получено новое значение функции качества $F(T_{\eta})$, его необходимо сравнить с наилучшим значением, которые было занесено в память – F_0 . После этого формируется величина шага поиска в допустимой зоне.

В том случае, когда $F(T_{\eta}) > F_0$, считается, что шаг выбран удачно, и F_0 необходимо изменить на значение $F(T_{\eta})$; иначе шаг является неудачным, и F_0 остается без изменения. В результате этого наилучшее значение F_0 постоянно хранится в памяти.

Процедура формирования шага аналогична правилу (7):

$$Ш_{\eta} = \begin{cases} Ш_{\eta-1}, \text{ если } F(T_{\eta}) > F_0 \text{ и } Y(F) < \chi(F) \text{ или } F(T_{\eta}) \leq F_0 \text{ и } H(F) < \delta(F); \\ \alpha(F)Ш_{\eta-1}, \text{ если } F(T_{\eta}) > F_0 \text{ и } Y(F) = \chi(F); \\ \beta(\Delta)Ш_{\eta-1}, \text{ если } F(T_{\eta}) \leq F_0, H(F) = \delta(F), \beta(F)Ш_{\eta-1} \geq \varepsilon(F); \\ Ш_0, \text{ если } F(T_{\eta}) \leq F_0, H(F) = \delta(F), \beta(F)Ш_{\eta-1} < \varepsilon(F) < \varepsilon(F), \rho(F) < \lambda(\Delta). \end{cases}$$

где все параметры имеют тот же смысл, что и в (7).

При получении удачного шага переходим к этапу VI, иначе к этапу IX. При $\rho(F) = \lambda(F)$ в качестве оптимальной принимается точка (совокупность моментов начал всех операций задач в информационно-управляющей системе) $t_n(i, j)$, соответствующая F_0 , которая хранится в памяти вместе со значением T_0 .

Имея набор величин $t_n(i, j)$, соответствующих оптимальному технологическому процессу, и зная продолжительность всех работ – $t(i, j)$, нетрудно рассчитать сроки окончания $t_0(i, j)$, и оптимальный план будет определен полностью.

Заключение

В статье предложены целочисленная модель и процедура формирования оптимального плана выполнения задач информационно-управляющей системы, реализующая метод случайного поиска с пересчетом с переменной величиной шага. Формируемый план задач информационно-управляющей системы представлен в виде сети. Предлагаемый подход позволяет получить оптимальный сетевой план задач информационно-управляющей системы в виде расписания начала времени выполнения задач системы, соответствующий минимальному времени выполнения всех задач информационно-управляющей системы и оптимальному использованию вычислительных ресурсов.

Список литературы

1. Ковалев И. В. Параллельные процессы в информационно-управляющих системах. Формирование и оптимизация: монография / И. В. Ковалев, Р. Ю. Царев, Ю. Г. Шиповалов // Под ред. д.т.н., проф. А.В. Медведева. – Красноярск: НИИ СУВПТ, 2001. – 143 с.
2. Планирование периодических задач при распределенной обработке информации / Н. А. Алексеев, О. В. Богданова, И. В. Ковалев, Р. Ю. Царев // Информационно-измерительные и управляющие системы. – 2010. – Т. 8. – № 3. – С. 11–14.
3. Царев М. Ю. Модификация GERT-сети для анализа временных характеристик сетевых моделей / М. Ю. Царев, Р. Ю. Царев, С. Ф. Шевчук // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. – 2009. – № 1-2. – С. 74–78.

4. Черниговский А. С. К проблеме формирования планов выполнения задач в системах управления реального времени / А. С. Черниговский // Теоретические и прикладные аспекты современной науки. – 2015. – № 7-3. – С. 145–147.
5. Bendraouche M., Boudhar, M., Oulamara, A. Scheduling: Agreement graph vs resource constraints (2015) *European Journal of Operational Research*, 240 (2), pp. 355–360.
6. Koehler F., Khuller, S. Optimal batch schedules for parallel machines (2013) *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8037 LNCS, pp. 475–486.
7. Kushwaha S., Kumar, S. Analysis of list scheduling algorithms for parallel system (2015) 2014 *International Conference on High Performance Computing and Applications, ICHPCA 2014*, art. no. 7045327.
8. Li, Z. The study of key technologies and the performance of the parallel computer system (2014) *WIT Transactions on Information and Communication Technologies*, 46 VOLUME 3, pp. 2011–2016.
9. Shah D., Tse, D.N.C., Tsitsiklis, J.N. Hardness of low delay network scheduling (2011) *IEEE Transactions on Information Theory*, 57 (12), art. no. 6094268, pp. 7810–7817.

Рецензенты:

Бронов С.А., д.т.н., профессор, руководитель научно-учебной лаборатории систем автоматизированного проектирования кафедры систем искусственного интеллекта Сибирского федерального университета, г. Красноярск;

Ченцов С.В., д.т.н., зав. кафедрой «Системы автоматики, автоматизированное управление и проектирование» Сибирского федерального университета, г. Красноярск.