ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ И ИННОВАЦИОННОГО РАЗВИТИЯ ВНУТРЕННИХ ВОДНЫХ ПУТЕЙ В БАССЕЙНАХ ВЕЛИКИХ РЕК
Интернет журнал широкой научной тематики.
Выпуск 5, 2016 г.
ISBN 978-5-901772-51-0

УДК 519.987

М.А. Трухина, ст. преподавтель ФГБОУ ВО «ВГУВТ»
Ю.С. Федосенко, д.т.н., профессор ФГБОУ ВО «ВГУВТ»
А.В. Шеянов, к.т.н., ст. преподаватель ФГБОУ ВО «ВГУВТ»
603950, г. Нижний Новгород, ул. Нестерова, 5

УЧЕТ ДИРЕКТИВНЫХ СРОКОВ В ЗАДАЧЕ ОДНОПРОЦЕССОРНОГО ОБСЛУЖИВАНИЯ МУЛЬТИПОТОКА ОБЪЕКТОВ

Ключевые слова: теория расписаний, оптимальность, конечный детерминированный поток объектов, динамическое программирование, ветви и границы, директивные сроки

Рассматриваются два варианта учета ограничения типа «директивные сроки» в оптимизационной задаче однопроцессорного обслуживания мобильным процессором конечного детерминированного мультипотока объектов. Предлагаются конструктивные варианты реализации решающих алгоритмов на основе концепций динамического программирования и ветвей и границ.

Рассматривается математическая модель локальной системы транспортной логистики, в которой мобильный процессор должен реализовать однократное обслуживание мультипотока объектов. Одна из возможных содержательных интерпретаций: подпотоки объектов поступают для обслуживания к нескольким пространственно разнесенным терминалам, а обслуживающий процессор затрачивает время на перемещение между терминалами (переналадка). В отличие от моделей, рассмотренных в [1, 2], на обслуживание объектов накладывается дополнительное ограничение в виде директивных сроков.

Считается заданным поток $O_n = \{o_1, o_2, \ldots, o_n\}$ объектов, подлежащих однофазному непрерывному обслуживанию процессором *P*.

Каждый объект o_i, $i = \overline{1,n}$ характеризуется следующими параметрами:

t_i − момент времени готовности к обслуживанию;

τ_i − продолжительность обслуживания;

a_i − величина штрафа за единицу времени пребывания объекта в системе обслуживания.

Процессор *P* характеризуется моментом времени *T* готовности к обслуживанию.

Не ограничивая общности, полагаем объекты упорядоченными по моменту времени поступления для обслуживания, т.е. $t_1 \le t_2 \ldots \le t_n$.

Обслуживание каждого объекта реализуется без прерываний; в каждый момент времени процессор может обслуживать только один объект потока O_n; по завершению обслуживания объект немедленно освобождает процессор; немотивированные простои процессора и объектов запрещены [3].

Под расписанием обслуживания обычно понимают однозначное установление порядка выполнения подлежащих диспетчеризации технологических операций, например, путем указания моментов времени начала каждой операции.

В силу принятых выше ограничений на процесс обслуживания потока O_n расписание ρ является компактным [1] и, следовательно, его можно однозначно отобразить перестановкой номеров объектов (i_1, i_2, ..., i_n) в порядке их взаимодействия с процессором.

Обозначим через $t_{i_k}^{нач}$ и $t_{i_k}^{кон}$ соответственно моменты начала и завершения обслуживания объекта o_{i_k} при реализации расписания ρ, $k = \overline{1,n}$.

Из компактности расписания ρ следуют соотношения

$$t_{i_1}^{нач} = \max(t_{i_1}, T),\ \ t_{i_k}^{нач} = \max\left[t_{i_{k-1}}^{кон}, t_{i_k}\right],\ k = \overline{2,n},\ \ t_{i_k}^{кон} = t_{i_k}^{нач} + \tau_{i_k},\ k = \overline{1,n}.$$

Расписание обслуживания потока однозначно определяет для каждого объекта o_i продолжительность его пребывания в системе $-$ $\Delta_{i_k} = t_{i_k}^{кон} - t_{i_k}$.

Значения функции штрафа $\varphi_i(\Delta_{i_k})$ за пребывание объекта o_i, $i = \overline{1,n}$ в системе обслуживания по расписанию ρ записывается в виде $a_i \cdot \Delta_{i_k}$ и, соответственно, линейный критерий $K(\rho)$ оценки качества расписания записывается как $\sum_{i=1}^{n} \varphi_i(\Delta_{i_k})$.

Задача оптимизации состоит в нахождении расписания ρ^*, минимизирующего значение критерия $\rho^* = \min\limits_{\rho} K(\rho)$.

Наличие m подпотоков формализуется добавлением к характеристикам объекта признака g_i принадлежности к одному из них.

К параметрам, характеризующих процессор P в этом случае добавляется признак начальной настройки g_0 и матрица $H(h(\alpha,\beta))$ размерности $m \times m$ продолжительностей перенастройки с режима обслуживания подпотока O_α на режим обслуживания подпотока O_β; $h(\alpha,\beta) > 0$ при $\alpha \neq \beta$ и $h(\alpha,\beta) = 0$ при $\alpha = \beta$.

Учет необходимости перенастройки процессора приводит к следующим соотношениям:

$$t_{i_1}^{нач} = \max(t_{i_1}, T + h(g_0, g_{i_1})),\ t_{i_k}^{нач} = \max\left[t_{i_{k-1}}^{кон} + h(g_{i_{k-1}}, g_{i_k}), t_{i_k}\right],\ k = \overline{2,n},\ t_{i_k}^{кон} = t_{i_k}^{нач} + \tau_{i_k},\ k = \overline{1,n}.$$

Для каждого объекта $o_i, i = \overline{1,n}$ имеется директивный срок t_i^d, $t_i^d > t_i$. В случае так называемого «мягкого директивного срока» его превышение влечет за собой увеличение до b_i значения коэффициента штрафа за единицу времени $b_i = a_i + d_i$,. В этом случае величина индивидуального штрафа по объекту o_i рассчитывается по соотношению

$$\varphi_i = \begin{cases} a_i(t_i^{кон} - t_i), & \text{при } t_i^{кон} \le t_i, \\ a_i(t_i^d - t_i^{кон}) + b_i(t_i^{кон} - t_i^d), & \text{при } t_i^{кон} > t_i. \end{cases}$$

Если вторую строку в этом выражении переписать в виде $\varphi_i = a_i(t_i^{кон} - t_i) + d_i(t_i^{кон} - t_i^d)$, то с учетом определения согласно выражению

$$x \Theta y = \begin{cases} x - y, & x > y, \\ 0, & x \le y, \end{cases}$$

операции «положительная разность» получим запись вида $\varphi_i = a_i(t_i^{кон} - t_i) + d_i(t_i^{кон} \Theta t_i^d)$.

Для решения рассматриваемых задач можно применить алгоритмы, основанные на концепции динамического программирования (ДП) [4] и ветвей и границ (ВиГ) [5]. В обоих случаях базой является рассмотрение процесса синтеза расписания ρ^* как последовательность шагов по выбору для обслуживания следующего объекта α. Решение принимается, когда обслуживание предыдущего объекта завершено и процессор свободен.

Ситуация принятия решения характеризуется моментом принятия решения t и множеством Z объектов, ожидающих обслуживания. Для учета ситуации «отказ от обслуживания в связи с ожиданием следующего объекта» к Z добавляется фиктивный объект, моделирующий процесс ожидания.

Для отображения ситуации наличия мультипотока к состоянию (t, Z) добавляется текущая настройка процессора g, а к множеству Z добавляются фиктивные объекты, отображающие перенастройку на другой подпоток. В итоге для алгоритма ДП [2] рекуррентные соотношения ДП записываются в виде

$$\Psi(t, g, Z) = \min_{\alpha\in\gamma_t}\begin{cases} G(t,g,\gamma_t,\alpha) + \Psi(t^{\text{кон}}(\alpha), g, (Z\setminus\{\alpha\})\cup D(t, t^{\text{кон}},(\alpha)) \mid \alpha\in Z, \text{обычный объект;}\\ G(t,g,\gamma_t,\alpha) + \Psi(t^{\text{кон}}(\alpha), j, Z\cup D(t, t^{\text{кон}}(\alpha)) \mid \alpha = f_j,\ j\neq g \text{ переналадка;}\\ G(t,g,\gamma_t,\alpha) + \Psi(t^{\text{кон}}(\alpha), g, Z\cup D(t, t^{\text{кон}}(\alpha)) \mid \alpha = f_w, \text{ ожидание.}\end{cases}$$

Здесь $\Psi(t, g, Z)$ − функция Беллмана, представляющая собой минимальную величину суммарного штрафа за период от момента времени t до завершения обслуживания в ситуации, определяемой тройкой (t, g, Z) при оптимальном управлении;

γ_t – множество объектов α, подходящих для постановки следующим в расписание;

$D(t, t^{\text{кон}}(\alpha))$ – множество объектов, поступающих на обслуживание за период обслуживания объекта α при условии выбора его в качестве очередного;

$G(t, g, \gamma_t, \alpha)$ – величина суммарного штрафа за обслуживание объекта α на временном отрезке $[t^{\text{нач}}, t^{\text{кон}}]$;

$$G(t,g,\Upsilon_t,\alpha) = \sum_{\beta\in\Upsilon_t} a(\beta)\cdot(t^{\text{кон}}(\alpha)-t) + \sum_{\nu=t+1}^{t^{\text{кон}}(\alpha)-1}\ \sum_{\gamma\in F(\nu)} a(\gamma)\cdot(t^{\text{кон}}(\alpha)-\nu).$$

Реализация соотношений предполагает расчет $\Psi(t, g, Z)$ для всех возможных наборов (t, g, Z) по уменьшению параметра t. Процесс завершается получением $\Psi(T, g_0, D(0, T))$ – значения критерия на оптимальном расписании ρ^*, которое восстанавливается при обратном проходе от ситуации $(T, g_0, D(0, T))$; для этого в процессе вычислений следует фиксировать решение в ситуации (t, g, Z) – объект α^*, на котором достигается минимум.

Особенностью задачи с линейными функциями индивидуального штрафа является выполнение соотношения $\Psi(t_n, g, Z) = \Psi(t_n+\Delta, g, Z)$ при любом $\Delta>0$. С учетом данного обстоятельства расчет начинается с ситуаций $(t_n, g, \varnothing)$ для всех возможных подпотоков g и далее t из меняется до T.

В статье [2] вычислительная сложность алгоритма ДП оценивается как $O(m{*}2^n n^3)$. Для $n = 22$, $m = 6$ вычислительный эксперимент показал среднюю продолжительность расчета ρ^*, равную двум минутам.

Аналогичным образом с явным выделением ожидающих объектов, объекта-ожидания и объектов-переналадок реализуется метод ветвей и границ.

Поскольку "мягкие" директивные сроки не ограничивают пространство поиска, а влияют только на подсчет штрафа по расписанию обслуживания, можно сделать предположение, что это изменение не должно существенно влиять на производительность алгоритмов. В частности, для алгоритма ДП единственный его фрагмент, подверженный изменению – это вычисление значений $G(t, g, \gamma_t, \alpha)$. При нарушении линейности функций индивидуального штрафа правило «$\Psi(t_n, g, Z) = \Psi(t_n+\Delta, g, Z)$ для любого $\Delta >0$» перестает действовать. Это приведет к необходимости заполнять таблицу значений функции $\Psi(t, g, Z)$ в значительно большем объеме (до максимально возможного значения t). Соответственно, существенно возрастут требования алгоритма к памяти решателя, и снизится размерность поддающихся решению за выделенное время задач.

Аналогично влияние учета директивных сроков на алгоритм ВиГ ограничивается, на первый взгляд, изменением функции подсчета штрафа, включая оценки. На практике, можно выделить факторы, могущие как улучшить, так и ухудшить временные характеристики алгоритма: может ухудшиться качество оценок, но при этом может увеличиться диапазон значений критерия между лучшими и худшими расписаниями, увеличивающий эффективность отсеивания.

Вариант ограничения, известный как «жесткий директивный срок», запрещает обслуживать объекты после заданного момента времени t_i^d. Расписание, при реализации которого ограничение нарушается, считается недопустимым. Таким образом, возможна ситуация отсутствия у задачи допустимого решения.

Имеется возможность использовать алгоритм решения задачи с учетом «мягких» директивных сроков с целью учета «жестких». Для этого устанавливаем добавку за превышение директивного срока d_i, равную достаточно большому числу с тем, чтобы однократное превышение директивного срока приводило к превышению значений суммарной функции штрафа некоторого порогового значения, а расписания без нарушения ограничений не превышали этот порог. В этом случае получение в качестве результата значения суммарной функции штрафа, превышающего установленного порогового значения, говорит об отсутствии в задаче допустимых решений.

Тем не менее, учет ограничения «жесткие директивные сроки» в алгоритмах ДП и ВГ в явном виде может улучшить временные характеристики их работы за счет отсева путей поиска, начиная с ситуации, на которой ограничение нарушено. Для того чтобы это сработало в алгоритме ДП, нужно применить метод с предварительной разметкой – отсев произойдет на этапе разметки.

Статья подготовлена по результатам исследований, выполненных при финансовой поддержке РФФИ в рамках проекта № 15-07-03141.

Список литературы:

[1] Коган Д.И., Федосенко Ю.С. Задача диспетчеризации: анализ вычислительной сложности и полиномиально разрешимые подклассы // Дискретная математика. – 1996. Т. 8. Вып. 3. – С. 135–147.
[2] Коган Д.И., Трухина М.А., Федосенко Ю.С. Шеянов А.В. Модель и алгоритм синтеза стратегий обслуживания мультипотока объектов мобильным процессором // Системы управления и информационные технологии. – 2015. – №2(60). – С.40–43.
[3] Дуничкина Н.А., Коган Д.И., Федосенко Ю.С. Задача синтеза оптимальной по быстродействию стратегии обслуживания мобильным процессором линейно рассредоточенной группировки стационарных объектов // Системы управления и информационные технологии. – 2015. – №1.1(59). – С. 136–140.
[4] Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М.: Наука, 1965. 460 с.
[5] Land A.H. and Doig A.G. An automatic method of solving discrete programming problems // Econometrica. –1960. Vol. 28. №3. – pp. 497–520.

ADDING DUE DATE CONSTRAINTS TO THE PROBLEM OF SINGLE-MACHINE SERVICING OF MULTI-FLOW OF OBJECTS

M.A. Trukhina, Yu.S. Fedosenko, A.V. Sheyanov

Keywords: scheduling theory, optimality, finite deterministic flow of objects, dynamic programming, branch and bound, deadlines

The article considers two options for adding «due dates» constraints to the problem of servicing finite deterministic multi-flow of objects by mobile processor. Constructive implementations of solving algorithms based on dynamic programming and branch and bound ideology are offered.