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. преподаватель ФГБОУ ВО «ВГУВТ»
603950, г. Нижний Новгород, ул. Нестерова, 5

СИНТЕЗ РАСПИСАНИЙ ОБСЛУЖИВАНИЯ ПОТОКА ПАКЕТОВ ИДЕНТИЧНЫХ ОБЪЕКТОВ

Ключевые слова: теория расписаний, оптимальность, конечный детерминированный поток объектов, алгоритм ветвей и границ

Рассматривается модель однофазного обслуживания стационарным процессором детерминированного потока объектов, поступающих в составе пакетов. Пакет считается обслуженным, если завершены обслуживанием все входящие в него объекты. С каждым пакетом ассоциируется линейная функция штрафа от момента завершения его обслуживания. Все объекты одного пакета являются идентичными. Ставятся задачи синтеза расписаний обслуживания и на основе концепции ветвей и границ конструируется решающий алгоритм.

Рассматривается h-элементный поток O^h независимых пакетов $O(s)$, $s = \overline{1,h}$, объектов; каждый пакет $O(s)$ включает в себя один ведущий объект o^s и $n(s)$ $(n(s) \geq 1)$ единиц идентичных ведомых объектов o_s; объект o^s обеспечивает перемещение всех входящих в пакет $O(s)$ ведомых объектов, которые (в отличие от ведущего объекта) подлежат однофазному однократному обслуживанию стационарным процессором; норма длительности обслуживания объекта o_s определяется значением параметра τ_s, $\sum_{s=1}^{h} n(s) = n$ − общее количество ведомых объектов в потоке O^h.

Пакет $O(s)$, т.е. агрегация всех входящих в его состав ведомых объектов o_s под управлением ведущего объекта o^s в момент времени t_s поступает в очередь для обслуживания процессором. Не ограничивая общности, полагаем, что:

− выполняется цепочка неравенств: $0 \leq t_1 \leq \ldots t_s \ldots \leq t_h$;

− в начальный момент времени $t = 0$ процессор свободен и находится в состоянии готовности к выполнению операций обслуживания ведомых объектов o_s пакетов $O(s)$, $s = \overline{1,h}$;

− обслуживание объектов пакета $O(s)$ может быть начато свободным процессором в любой момент времени t $(t \geq t_s)$ и осуществляется без прерываний, $s = \overline{1,h}$;

− необслуженный объект не может покинуть очередь;

− одновременное обслуживание процессором двух и более ведомых объектов и его непроизводительные простои запрещены;

Поток O^h пакетов считается обслуженным в том и только в том случае, если завершены обслуживанием все образующие его пакеты $O(s)$; в свою очередь, пакет объектов $O(s)$ считается обслуженным в том и только в том случае, если завершены обслуживанием все входящие в него ведомые объекты o_s.

Момент времени $t_s^к$ завершения обслуживания пакета $O(s)$ определяется как момент завершения обслуживания последнего объекта o_s.

За каждую единицу времени нахождения в системе ведущего объекта o^s в ожидании завершения обслуживания всех ведомых объектов пакета $O(s)$ взимается штраф в размере a_s.

В задаче общего вида [1], без ограничения на идентичность ведомых объектов стратегия обслуживания S представляется в виде перестановки номеров ведомых объектов. Общее число возможных перестановок – $n!$.

В рассматриваемой задаче стратегия обслуживания S потока O^h через индексы ведущих объектов входящих в его состав пакетов $O(s)$, $s=\overline{1,h}$, представляется как произвольная перестановка с повторениями. Индекс s должен входить в перестановку ровно $n(s)$ раз. Общее число возможных перестановок определяется соотношением $n!\Big/\prod_{s=1}^{h} n(s)!$.

Множество допустимых стратегий обозначим Ω. Оценку той или иной стратегии обслуживания S лицо, принимающее решения может проводить на основе значений различных учитывающих условия складывающейся ситуации количественных критериев. В качестве таких критериев обычно выступают следующие [2]:

$K_1(S)=\sum_{s=1}^{h} a_s(t_s{}^k(S)-t_s)$ − суммарный по всем ведущим объектам штраф за время завершения обслуживания потока O^h;

$K_2(S)=\max\limits_{1\le s\le h}(a_s(t_s{}^k(S)-t_s)$ − максимальный штраф по всем ведущим объектам пакетов потока O^h за время ожидания завершения его обслуживания;

С позиций повышения эффективности управления обслуживанием потока O^h рассматриваются задача 1: $\min\limits_{S\in\Omega} K_1(S)$ и задача 2: $\min\limits_{S\in\Omega} K_2(S)$.

Для решения этих задач ниже конструируется алгоритм на основе концепции ветвей и границ [3], который заключается в построении дерева поиска с отсечением бесперспективных вершин. Каждая вершина дерева соответствует промежуточной ситуации принятия решения о постановке на обслуживание очередного объекта из множества ожидающих обслуживания, принимаемого в момент, когда процессор свободен. При этом вершина дерева содержит фрагмент перестановки (расписания) ρ длиной от 0 до n, текущее время t, верхнюю $ВО$ и нижнюю $НО$ оценки значений критерия $K_1(S)$ (или $K_2(S)$). Вершины, из которых будет производиться ветвление, называются открытыми и хранятся в списке открытых вершин. Вершина, содержащая полное расписание, является закрытой и хранится в списке закрытых. Ветвление из нее не производится. Отсеянные вершины не хранятся.

Схему алгоритма можно описать следующим образом.

Шаг 1. Начало. Формирование корня дерева. Расписание в этой вершине пустое, время t=0, $ВО$ − штраф на расписании, в котором пакеты обслуживаются в порядке поступления, $НО=\sum_{i=1}^{h} a_i\tau_i n(i)$. Список открытых инициализируется корнем дерева.

Шаг 2. Ветвление. Для ветвления выбирается открытая вершина. Ветвление в текущей вершине происходит по всем объектам, присутствующим в момент t в системе (уже пришел, но еще не обслужен), а также по фиктивному объекту f, означающему простой в ожидании поступления очередного пакета объектов; объект f не используется, если в момент времени t все пакеты потока O^h уже поступили. В результате формируются вершины следующего уровня. В каждой новой вершине содержится фрагмент расписания ρ длины L+1 (где L – длина фрагмента расписания родительской вершины) и текущее время, равное t+τ, где τ – продолжительность обслуживания выбранного объекта. В случае если ветвление осуществляется по фиктивному объекту, фрагмент расписания ρ не меняется, а время t совпадает с временем t_s поступления в систему очередного пакета.

Заметим, что если текущая вершина получена в результате выбора фиктивного объекта f, то очередное ветвление целесообразно проводить только по объектам вновь поступивших пакетов, т.е. таких, у которых момент времени $t_s = t$, и по фиктивному объекту.

Можно также существенно ускорить работу алгоритма, если исключить построение поддерева при $t \geq t_h$ и определять оптимальную последовательность обслуживания всех оставшихся объектов в соответствии с известным решающим правилом [4]; при этом необслуженные пакеты сортируются по убыванию отношения коэффициента штрафа a_s к суммарной длительности обслуживания оставшихся объектов пакета.

Для каждой открытой вершины рассчитываются оценки. BO представляет собой штраф на расписании, начальный фрагмент которого хранится в вершине, а все остальные объекты обслуживаются в порядке поступления. Для расчета HO штраф по фрагменту расписания ρ до момента t рассчитывается по формулам для $K_1(S)$ и $K_2(S)$, а далее снимается формальный запрет на одновременное обслуживание нескольких объектов. Имеющиеся в системе объекты считаются обслуживаемыми, начиная с момента t, а остальные − с момента поступления в систему. Для закрытых вершин значения BO и HO совпадают.

Шаг 3. Отсев. При работе алгоритма поддерживается значение текущего минимума верхней оценки – рекорд BO^*.

Если в очередной вершине значение HO превышает BO^*, то эта вершина отсеивается, так как оптимальная стратегия S не может быть получена из этой вершины. Если же у очередной вершины BO меньше BO^*, то обновляется значение рекорда и запускается процедура отсева по условию $HO > BO^*$ для всех вершин дерева.

Шаг 4. Завершение. Если множество открытых вершин непусто, выполняется шаг 2. В противном случае каждая закрытая вершина содержит оптимальное расписание обслуживания исходного потока пакетов объектов.

Отметим, что обе поставленные оптимизационные задачи являются обобщением рассмотренной в [5] канонической задачи диспетчеризации и, соответственно, относится к категории NP-трудных. Вместе с тем, вычислительные эксперименты показали, что учет идентичности объектов в вышеописанном алгоритме приводит к существенному сокращению продолжительности синтеза оптимальных расписаний и вполне приемлем для реализации в компьютерных системах поддержки управления транспортными процессами.

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

Список литературы:

[1] Трухина М.А., Федосенко Ю.С. Каноническая модель и задача синтеза стратегий однопроцессорного обслуживания потока пакетов объектов. Сб. Материалов IV Международной научно-практической конференции «Информационные управляющие системы и технологии». Одесса: ОНМУ, 2015. С. 83-85.
[2] Коган Д.И., Федосенко Ю.С., Дуничкина Н.А. Бикритериальные задачи обслуживания стационарных объектов в одномерной рабочей зоне процессора // Автоматика и телемеханика. – 2012. №10. – С. 93–110.
[3] Land A.H. and Doig A.G. An automatic method of solving discrete programming problems // Econometrica. –1960. Vol. 28. №3. – pp. 497–520.

[4] Танаев В.С., Гордон Я.М., Шафранский В.С. Теория расписаний. Одностадийные системы. –М.: Наука, 1984. – 382 с.
[5] 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.

SERVICE SCHEDULE SYNTHESIS FOR THE PACKET STREAM OF IDENTICAL OBJECTS

M.A. Trukhina

Keywords: scheduling theory, optimality, finite deterministic flow of objects, branch and bound algorithm

The model of a single-stage servicing by stationary processor of deterministic stream of objects coming in packets is considered. The packet is considered to be served if service is completed for all of its constituent objects. Each packet has an associated linear penalty function of the time moment of its service completion. All the objects that make up the package are considered to be identical. Problems of service schedule synthesis are stated. Based on branch-and-bounds ideology solving algorithm is constructed.