

ОНЛАЙН МОДЕЛИ И АЛГОРИТМЫ ПЛАНИРОВАНИЯ БЕЗ ПРЕРЫВАНИЙ НА ПАРАЛЛЕЛЬНЫХ ПРОЦЕССОРАХ

Тарасова Елизавета Юрьевна

Аспирант, Университет ИТМО, г. Санкт-Петербург
el.u.tarasova@gmail.com

ONLINE MODELS AND ALGORITHMS FOR SCHEDULING WITHOUT INTERRUPTION ON PARALLEL PROCESSORS

E. Tarasova

Summary. The purpose of this work is to study online models of uninterrupted scheduling theory and scheduling algorithms for them. The paper presents an overview of methods for tasks with different objective functions on parallel processors and for tasks with a due date. On the generated data, the MINDL and SPT algorithms were run to minimize the total delay. As a result of the comparison, it was revealed that the algorithms get similar results, except for examples with additional complications.

Keywords: online planning models; model without interruptions; directive term, model on parallel processors; minimization of the total delay.

Аннотация. Целью данной работы является исследование онлайн моделей теории расписаний без прерывания и алгоритмов планирования для них. В работе представлен обзор методов для задач с различными целевыми функциями на параллельных процессорах и для задач с директивным сроком. На сгенерированных данных было проведено алгоритмов MINDL и SPT для минимизации суммарного запаздывания. В результате сравнения было выявлено, что алгоритмы получают близкие результаты за исключением примеров с дополнительными осложнениями.

Ключевые слова: онлайн модели планирования; модель без прерываний; директивный срок, модель на параллельных процессорах; минимизация суммарного запаздывания.

Введение

Основное отличие онлайн планирования от офлайн постановки — это неполнота информации об операциях до их поступления. До начала работы процессора некоторые модели предполагают наличие информации о количестве поступающих заданий. Как и в классических моделях, в онлайн моделях регламентируются следующие параметры: момент поступления на процессор, длительность выполнения, директивные сроки, наличие или отсутствие доставки после выполнения, штрафные функции и запрет или разрешение на прерывания. Целевые функции сохраняются в соответствии с офлайн областью теории расписаний.

В силу неопределённости начальных данных алгоритмы онлайн моделей сводятся к принятию реше-

ния о выборе задачи для постановки на процессор. Вследствие этого фактора эффективность алгоритмов определяется не только целевой функцией и вычислительной сложностью, но и длительностью принятия решения. Обоснование критерия лежит в естественных наблюдениях — время на процессоре не останавливается, длительность выбора задачи (или принятия решения о простое) напрямую влияет на время окончания последующих задач, и так как большинство целевых функций зависят от данного параметра, то и на результаты работы.

Так как в онлайн моделях требуется, в первую очередь, определить порядок действий в каждый момент времени работы процессора, в отличие от офлайн моделей, итоговое расписание будет известно только после завершения обслуживания, и оно будет нести информационный характер.

Среди различных методик оценок эффективности алгоритма подходящими для онлайн моделей выделяются следующие: вычислительная сложность: время принятия решения в каждый требуемый момент времени, сравнение полученной целевой функции с оптимальным значением для аналогичной офлайн задачи.

Литературный обзор

Большой интерес к применению онлайн алгоритмов к прикладным задачам объясняет и тот факт, что большая часть исследований в данной области сводится к частным случаям и с различными дополнительными условиями, выходящими за рамки классических понятий теории расписаний. Однако поиск решения для общих задач является востребованным, так как их область применения шире и на них могут базироваться частные случаи.

В исследовании [6] рассмотрены различные задачи онлайн планирования, базирующиеся на общих задачах теории расписаний. Например, задача на параллельных процессорах, задачи с доставкой, максимизация общего времени выполнения и т.д. Также в работе А.П.А. Вестженса [6] исследуются существующие алгоритмы, а также предложены их модификации для более сложных случаев.

Некоторые классические подходы применимы к онлайн постановке. Например, алгоритм Смита [5] для задачи на параллельных процессорах, алгоритм наибольшего времени доставки — правило Джексона [2] — для частного случая задачи с доставкой.

Материалы и методы

Одной из наиболее распространенных задач планирования является задача на параллельных процессорах без прерываний. Работы выполняются на m параллельных идентичных процессорах. Идентичность процессоров предполагает, что любая операция может быть выполнена на любом из них. Задания, поступающие на процессор, независимы друг от друга (нет частичного порядка), то есть они могут обслуживаться в любом порядке. Каждый процессор в один момент времени может быть выполнять только одну работу. Для каждой поступающей на процессор задачи определены: время поступления r_i , время обработки на процессоре p_i .

В качестве критериев оптимальности могут выбираться различные функции: длина расписания, суммарное время завершения всех заданий $C_{\Sigma}(S) = \sum_{v_i \in V} C_i$, суммарное нахождение заданий на процессоре.

Для некоторых из вышеперечисленных критериев для однопроцессорной модели существуют детерминированные онлайн алгоритмы, например, предложенные в исследованиях [4, 6].

Для рассматриваемой модели с целевой функцией $C_{\Sigma} \rightarrow \min$ существует детерминированный алгоритм наименьшего времени обработки (SPT), предложенный Смитом [5].

Алгоритм SPT: если машина простаивает, из числа доступных заданий выбирается то, которое имеет наименьшее время обработки.

Другим подходом является отложенный алгоритм DSPT, который базируется на SPT и идее разрешения простоя процессора даже при наличии задач, готовых к выполнению [6].

1. Если процессор свободен, определяется работа v_i с наименьшим временем выполнения p_i . Для нее вычисляется промежуток времени t , в течение которого v_i находится на процессоре с момента поступления: $t = \text{time} - r_i$, где time — текущее время на процессоре.
2. В случае, если критерию пункта 1 соответствует несколько задач, выбирается та, у которой наименьшее время поступления на процессор r_i .
3. Если $p_i \leq t$ на процессор ставится r_i ; в противном случае процессор простаивает до времени $t = p_i$ или до прибытия нового задания, в зависимости от того, что произойдет раньше.

Согласно исследованию [6] интуитивное обоснование этого алгоритма заключается в том, что для компенсации неизвестности, какая задача придет в следующий момент времени, DSPT ждет достаточно долго более маленькую задачу. При этом в случае неправильного решения любое маленькое задание v_i , выполненное после работы v_j , которая поставлена на процессор, имеет время завершения C_i , по крайней мере, такое же большое, как C_j , и поэтому его задержка не больше, чем его вклад в $\sum_{v_i \in V} C_i$.

В работе [6] была предложена модификация онлайн алгоритма DSPT для многопроцессорной задачи. Если один из m процессоров свободен, то применяется алгоритм DSPT на всех доступных к выполнению задачах V^* .

Рассмотрим задачу составления расписания для независимых заданий, для каждого из которых определено время поступления r_i время обработки на процессоре p_i и директивный срок D_i . Время окончания каждой операции $v_i \in V$ определяется как $C_i = \tau_i + p_i$, где τ_i

— время начала выполнения. Множество V' включает в себя операции, завершённые до их директивных сроков: $V' = \{v_i \in V: C_i \leq D_i\}$.

Существует несколько моделей с максимизацией целевых функций: суммарное время работ, выполненных вовремя (то есть до их директивных сроков): $E_{\Sigma}(S) = \sum_{v_i \in V'} p_i$, количество заданий, выполненных в срок: $R(S) = |V'|$.

В исследовании [3] был предложен алгоритм SR для целевой функции $E_{\Sigma}(S)$. За время задержки принимается величина $l_i = D_i - p_i - r_i$, задающая промежуток времени, на который может быть отложено выполнение задания v_i без уменьшения целевой функции. Тогда, если p_i — время выполнения работы v_i , то задание может быть отложено, как минимум, на величину k^*p_i .

Основная идея алгоритма заключается в том, что одна из машин выбирается как основная m_{main} и планирование происходит на ней, но далее основная машина меняется по циклу, что позволяет последовательно выбрать задания к исполнению на каждом из m процессоров. Пусть v_j — задание, готовое к выполнению на процессоре m_j . Если на новой текущей машине выполняется одно из заданий, то переключение машины задерживается. Алгоритм сохраняет время завершения этого задания как время пробуждения w_i . Переключение на следующий процессор происходит во время пробуждения. Смена текущей машины также происходит, если некоторое задание $w_{j_0}: p_{w_{j_0}} > p_{j_0}$ поступает до времени пробуждения. В переменной ls хранится наибольшее p_i среди выполненных заданий, A — множество доступных для выполнения заданий. Наконец, когда нет доступных заданий и текущая машина находится в состоянии простоя, алгоритм начинает свою работу заново.

Другой однопроцессорной онлайн моделью с директивными сроками является минимизация суммарного запаздывания всех заданий:

$$L_{\Sigma}(S) = \sum_{v_i \in V} \max(\tau_i + p_i - D_i, 0) \rightarrow \min \quad (1)$$

где τ_i — время начала обработки задания v_i . При этом задания выполненные досрочно не вносят дополнительный вклад в целевую функцию.

Одним из естественных методов решения данной задачи является постановка на процессор задачи с минимальным директивным сроком из тех, которые поступили на процессор к моменту выбора. Выбор подкрепляется интуитивно-понятной идеей: чем раньше начнет выполняться задача с меньшим запасом време-

ни, тем больше вероятность, что она будет выполнена в срок.

Алгоритм MINDL. Пока процессор занят, считываются поступающие задания и добавляются в множество $V^* = \{v_i \in V: r_i \leq time\}$. В момент освобождения процессора выбирается из невыполненных задание v_k такое, что $D_k = \min_{v_i \in V^*} (D_i)$.

В качестве альтернативного алгоритма для данной задачи можно рассмотреть SPT, в котором задания упорядочиваются согласно времени выполнения.

Результаты

Было проведено экспериментальное сравнение алгоритмов MINDL и SPT для однопроцессорной онлайн задачи без прерываний с директивными сроками. Тестирование проводилось на синтетических данных, сгенерированных методом, описанным Дж. Карлье [1]. Сгенерированные примеры составили более 600 задач размерность от 100 до 500 (размерность определяется количеством заданий, поступающих в систему), среди которых были выделены с группы примеров с одним или несколькими заданиями, а также усложненные примеры, согласно [1].

Более, чем на половине примерах алгоритмы получили близкие результаты, на оставшихся наблюдается среднее отклонением в 2% в пользу алгоритма MINDL. Однако алгоритм SPT получил более существенные улучшения целевой функции (на 8%) для примеров с одной и двумя большими задачами.

Заключение

В работе были представлены к рассмотрению несколько различных моделей планирования в условия неопределённости начальных данных. Среди них была выбрана однопроцессорная модель минимизации суммарного запаздывания для задач с директивным сроком без прерываний. Для данной модели было проведено эмпирическое сравнение двух алгоритмов на сгенерированных данных. В результате сравнения было выявлено, что алгоритм SPT получается лучшие результаты для примеров с заданиями большого размера. Для остальных примеров целевые функции получаются близкими для двух алгоритмов с небольшим отклонением в пользу алгоритма MINDL. Таким образом, можно сказать, что в случае, если предполагается наличие больших заданий, может быть рекомендован алгоритм SPT. Дальнейшие исследования включают в себя адаптация других алгоритмов планирования для данной задачи, рассмотрение аналогичной задачи с параллельными процессорами, а также расширение и усложнение данных для тестирования.

ЛИТЕРАТУРА

1. Carlie J. The one machine sequencing WSPtem // European Journal of Operational Research. — 1982. Vol. 11. P. 42–47.
2. Jackson J.R. Scheduling a production line to minimize maximum tardiness // Research Report 43, Management Science Research Project / University of California, Los Angeles. — 1955.
3. Lee J.-H. Online Deadline Scheduling: Multiple Machines and Randomization. // SPAA '03: Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures — 2003. P. 19–23
4. Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmoys D.B. Sequencing and scheduling: Algorithms and complexity // S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin (eds.), Logistics of Production and Inventory, Handbooks in OR & MS4, Elsevier / Amsterdam. — 1993. Ch. 9, P. 445–522.
5. Smith W.E. Various optimizers for single-stage production // Naval Research Logistics Quarterly 3. — 1956. P. 59–66.
6. Vestjens A.P.A. On-line machine scheduling, Ph.D. Thesis / Eindhoven University of Technology / Netherlands. — 1997.

© Тарасова Елизавета Юрьевна (el.u.tarasova@gmail.com).

Журнал «Современная наука: актуальные проблемы теории и практики»

