ТЕХНИЧЕСКАЯ
КИБЕРНЕТИКА
№ 4 · 1982

УДК 519.83

ДЕТЕРМИНИРОВАННЫЕ ЗАДАЧИ ПЛАНИРОВАНИЯ ПРИ ИДЕНТИЧНЫХ ПРОЦЕССОРАХ И ОДНОВРЕМЕННОМ ПОСТУПЛЕНИИ ЗАЯВОК

ЛЕВИН М. Ш.

Введение. При разработке вычислительных систем большое значение имеет выбор математических моделей и экономных алгоритмов упорядочения заявок (работ, заданий) в процессе составления расписания. Один из важных подклассов моделей планирования вычислительного процесса составляют детерминированные задачи теории расписаний [1—3]. Многие из подобных задач обладают большой вычислительной сложностью и принадлежат к классу универсальных переборных проблем [4—10, 36, 40, 41]. В последние годы наряду с исследованием сложности детерминированных задач теории расписаний [7—11] большое внимание уделяется построению экономных приближенных алгоритмов, в которых за счет снижения требований к точности решения уменьшается трудоемкость вычислений [8, 12—16, 37, 40—42].

Модели распределения работ по идентичным процессорам соответствуют важному с точки зрения практики классу реальных вычислительных систем [17]. В данной работе рассмотрены постановки детерминированных задач составления расписаний при идентичных процессорах и одновременном поступлении заявок с учетом типов критериев, ограничений предшествования, различного числа процессоров и соотношений между временами выполнения работ. Приведены известные автору характеристики сложности задач этого типа, эффективных точных алгоритмов и приближенных методов решения с гарантированной погрешностью.

1. Основные определения.

Определение 1. [4—6, 8, 10, 11]. Алгоритм называется эффективным, если его трудоемкость и требуемая память полиномиально зависят от размерности задачи. (Размерность задачи — это длина двоичной записи всех исходных данных.)

Большое число задач планирования вычислительного процесса относится к числу универсальных переборных. Широко распространена гипотеза, что задачи этого класса неразрешимы за полиномиальное время [4—6, 8]. В случае, когда задача планирования принадлежит к классу универсальных переборных, целесообразным является применение полиномиального приближенного алгоритма, если такой известен. Следует иметь в виду, что для некоторых универсальных переборных задач поиск приближенного решения с гарантированной погрешностью является такой же сложной проблемой, как и поиск точного решения [8, 38, 40].

Определение 2. [8, 41]. Алгоритм называется ε-приближенным (или, соответственно, ε_a-приближенным) для массовой задачи Z, если для любой индивидуальной задачи I_z он вырабатывает решение $\bar{x}$ такое, что $|f(I_z, \bar{x})-f^*(I_z)|\leqslant\varepsilon f^*(I_z)$ (или, соответственно $|f(I_z, \bar{x})-f^*(I_z)|\leqslant\varepsilon_a$), где $f(I_z, \bar{x})$ — значение функции на решении, вырабатываемом алгоритмом, $f^*(I_z)$ — оптимальное значение целевой функции.

Могут использоваться другие определения приближенных алгоритмов [37].

Сравнение задач планирования далее проводится по следующим признакам:

1) существование эффективного точного алгоритма;

2) принадлежность задачи к классу универсальных переборных проблем;

3) наличие известного ε-приближенного эффективного алгоритма.

2. Постановки задач. Пусть имеется множество заданий $R=\{1,\ldots,n\}$ и множество идентичных процессоров $P=\{P_1,\ldots,P_m\}$. Каждое задание $i\in R$ характеризуется временем выполнения τ_i; директивным сроком d_i, к которому необходимо выполнить задание; моментом окончания выполнения C_i; коэффициентом важности a_i; функцией штрафа $\varphi_i(t)$. В качестве аргумента функции штрафа часто используется C_i. Функция штрафа обычно имеет линейный или экспоненциальный вид, однако возможны и более сложные виды зависимости [1—3, 10, 11, 18, 19]. Рассмотрим следующие два типа критериев качества составления расписаний [10]:

$f_1=\max\limits_{1\leqslant i\leqslant n}\{\varphi_i(C_i)\}$ и $f_2=\sum\limits_{i=1}^{n}\varphi_i(C_i)$ В первом случае минимизируется максимальное значение функции штрафа, во втором — суммарное или среднее. В качестве критериев далее будут рассматриваться

$$C_{\max}=\max_{1\leqslant i\leqslant n}C_i;\quad \sum C_i=\sum_{i=1}^{n}C_i;$$

$$\sum a_iC_i=\sum_{i=1}^{n}a_iC_i;\quad \sum a_ie^{\lambda C_i}=\sum_{i=1}^{n}a_ie^{\lambda C_i};\quad \sum U_i=\sum_{i=1}^{n}U_i,$$

$$\text{где}\quad U_i=\max(0,C_i-d_i);\quad \sum a_iU_i=\sum_{i=1}^{n}a_iU_i;$$

$\sum T_i=\sum\limits_{i=1}^{n}T_i$, где T_i равно 0 при $C_i\leqslant d_i$ и 1 в противном случае;

$\sum a_iT_i=\sum\limits_{i=1}^{n}a_iT_i$. Перечисленные критерии соответствуют задачам минимизации следующих величин: общего времени выполнения заданий, суммы (или среднего значения) моментов, окончания выполнения заданий, взвешенной суммы (или средневзвешенного значения) моментов окончания выполнения заданий, взвешенной суммы (или средневзвешенного значения) величин, экспоненциально зависящих от моментов окончания выполнения заданий, суммы запаздываний, взвешенной суммы запаздываний, числа запаздывающих работ, взвешенного числа запаздывающих работ.

Следует заметить, что число моделей теории расписаний, используемых при детерминированном планировании вычислительных процессов, велико. В практических ситуациях между упорядочиваемыми программами часто существуют информационные связи, что приводит к необходимости учета ограничений предшествования при обслуживании заявок. В последние годы возрос интерес к моделям организации вычислительного процесса, в которых учитываются дополнительно требуемый при выполнении заявки ресурс, например, память [16, 36].

3. Таблица. Введем следующее позиционное обозначение постановок задач: А|Б|В|Г, где в позиции А проставляется число процессоров; в позиции Б — типы ограничения предшествования (если в этой позиции пропуск, то ограничение предшествования отсутствует); в позиции В — возможные значения времен выполнения работ; в позиции Г — вид критерия (целевой функции)

$$C_{\max},\ \sum C_i,\ \sum a_i e^{\lambda C_i},\ \sum a_i C_i,\ \sum U_i,\ \sum a_i U_i,\ \sum T_i,\ \sum a_i T_i.$$

Число процессоров может быть равно 1, 2 или $m(m>2)$. Возможны следующие обозначения типов ограничений предшествования: дерево (совокупность деревьев), П-сеть (параллельно-последовательная сеть), G (ациклический направленный граф). Значения времен выполнения работ могут характеризоваться тремя вариантами обозначений: 1 (все времена выполнения равны); 1, 2 (времена выполнения равны некоторой константе или ее удвоенному значению); τ (времена выполнения произвольны из интервала $(0, \infty)$).

Каждой из введенных задач планирования соответствует клетка таблицы. В каждой клетке приведено обозначение порядка сложности задачи: «*» — существует эффективный точный алгоритм «!» — задача является универсальной переборной, «?» — вопрос о сложности задачи в настоящее время является открытым, «a» — существует эффективный приближенный алгоритм с гарантированной точностью.

4. Анализ таблицы. Очевидно, что если задача является универсальной переборной, то порядок ее сложности не изменяется при следующих изменениях в постановке:

— переход от одного процессора к двум и более;

— переход от равных времен выполнения работ к неравным;

— учет ограничения предшествования более сложного вида (от отсутствия ограничения предшествования к дереву, П-сети или ациклическому направленному графу).

Если же для задачи существует полиномиальный аглоритм, то он применим и при изменениях в постановке в обратных направлениях.

Для некоторых задач в таблице указаны ссылки па литературный источник, в котором показывается принадлежность задачи к классу универсальных переборных проблем или приводится полиномиальный точный алгоритм решения. В последнем случае указана также минимальная из известных оценка трудоемкости алгоритма. На основе этих задач, как исходных, в таблице указан порядок сложности других, близких по постановке. Дополнительно, в качестве исходных использованы задачи $m\|1|C_{\max}$, $m\|1|\sum a_iC_i$, $m\|1|\sum a_ie^{\lambda C_i}$: т. к. для них существуют очевидные точные аглоритмы решения. Следует также заметить, что задачи с критериями $\sum a_iC_i$ и $\sum a_ie^{\lambda C_i}$ очевидно близки по сложности.

Приближенные эффективные алгоритмы с гарантированной точностью разработаны для шести исходных задач (в таблице для них указаны ссылки на литературные источники). Для задачи $2\|\tau|C_{\max}$ трудоемкость ε-приближенного алгоритма не превосходит $O(n/\varepsilon)$, для задачи $1\|1|\sum a_iU_i$ — $O(n^3/\varepsilon)$ [12]. Аналогичный алгоритм для задачи $m\|\tau|C_{\max}$ имеет трудоемкость $O(n^{m-1}/\varepsilon^{2m-2})$ [14], для задачи $m|\text{дерево}|\tau|\sum a_iC_i$ построен ε_a-приближенный алгоритм с трудоемкостью $O(n\log n)$, в котором $\varepsilon_a = k-(k/m)$, где $k=\max\limits_{i\leqslant i\leqslant n}\tau_i$ [20]. Алгоритм с трудоемкостью $O(n^m)$, обеспечивающий относительную ошибку не более $^1/_2$, построен для задачи $m|G|\tau|C_{\max}$ [15], в работе [42] для этой задачи предлагается алгоритм, обеспечивающий аналогичную погрешность при трудоемкости $O(n^4)$.

Таблица А

Без ограничения предшествования

Число процессоров	Время выполнения работы	Вид целевой функции: ΣC_i	C_{max}	$\Sigma a_i e^{\lambda C_i}$	$\Sigma a_i C_i$	ΣU_i	$\Sigma a_i U_i$	ΣT_i	$\Sigma a_i T_i$
1	1	*	*	*	*	*	* O(n log n) [36]	*	?
	1,2	*	*	*	*	?	?	*	?
	(0, ∞)	*	*	* O(n log n) [19]	* O(n log n) [35]	?	! [36]	* O(n log n) [36]	! [43, 44] a [12]
2	1	*	*	*	*	?	?	?	?
	1,2	*	*	?	?	?	?	?	?
	(0, ∞)	*	! [19] a [12]	?	! [9] a	?	!	?	!
$m \geqslant 3$	1	*	*	* O(n log n)	* O(n log n)	?	?	?	?
	1,2	*	* O(n)	?	?	?	?	?	?
	(0, ∞)	* O(n log n) [18]	! a [14]	?	! a [13, 14]	?	!	?	!

Таблица Б

Ограничение предшествования – дерево

Число процессоров	Время выполнения работы	Вид целевой функции							
		ΣC_i	C_{max}	$\Sigma a_i e^{\lambda C_i}$	$\Sigma a_i C_i$	ΣU_i	$\Sigma a_i U_i$	ΣT_i	$\Sigma a_i T_i$
1	1	*	*	*	*	?	?	! [36]	!
	1,2	*	*	*	*	?	?	!	!
	(0, ∞)	*	*	* O(n log n) [22, 26, 33]	* O(n log n) [26, 29, 31]	?	!	!	!
2	1	*	*	?	?	?	?	!	!
	1,2	?	?	?	?	?	?	!	!
	(0, ∞)	?	!a	?	!a	?	!	!	!
$m \geqslant 3$	1	* O(n) [9]	* O(n) [21, 27]	?	?	?	?	!	!
	1,2	?	?	?	!	?	?	!	!
	(0, ∞)	?	!a	?	! a [13]	?	!	!	!

Таблица В

Ограничение предшествования – П-сеть

Число процессоров	Время выполнения работ	Вид целевой функции							
		ΣC_i	C_{max}	$\Sigma a_i e^{\lambda C_i}$	$\Sigma a_i C_i$	ΣU_i	$\Sigma a_i U_i$	ΣT_i	$\Sigma a_i T_i$
1	1	*	*	*	*	?	?	!	!
	1,2	*	*	*	*	?	?	!	!
	(0, ∞)	*	*	* O(n log n) [23, 25, 33]	* O(n log n) [23, 25, 28, 33, 34]	?	!	!	!
2	1	*	*	?	?	?	?	!	!
	1,2	?	?	?	?	?	?	!	!
	(0, ∞)	?	!a	?	!	?	!	!	!
$m \geqslant 3$	1	?	?	?	?	?	?	!	!
	1,2	?	?	?	?	?	?	!	!
	(0, ∞)	?	!a	?	!	?	!	!	!

7. Выводы.

1. В работе систематизированы детерминированные задачи планирования при идентичных процессорах и одновременном поступлении заявок для восьми видов критериев. Это позволяет проводить на основе сравнения оценку порядка сложности возникающих задач планирования и осуществлять выбор известных методов решения.

2. В теоретическом плане представляется целесообразным исследование сложности задач, для которых вопрос о сложности остается открытым, построение эффективных, приближенных алгоритмов, а также систематизация других типов задач.

Таблица Г

Ограничение предшествования – ациклический направленный граф

Число процессоров	Время выполнения работ	Вид целевой фукции							
		ΣC_i	$C_{\max}$	$\Sigma a_i e^{\lambda C_i}$	$\Sigma a_i C_i$	ΣU_i	$\Sigma a_i U_i$	ΣT_i	$\Sigma a_i T_i$
1	1	*	*	?	! [9]	! [36]	!	!	!
	1,2	! [9]	*	?	!	!	!	!	!
	(0, ∞)	!	* O ($n \log n$) [32, 39]	?	!	!	!	!	!
2	1	* O (n^2) [9]	* O (n^2) [30]	?	!	!	!	!	!
	1,2	! [9]	! [7]	?	!	!	!	!	!
	(0, ∞)	!	!a	?	!	!	!	!	!
$m \geqslant 3$	1	! [9]	! [7] a	?	!	!	!	!	!
	1,2	!	!a	?	!	!	!	!	!
	(0, ∞)	!	! a [15, 42]	?	!	!	!	!	!

ЛИТЕРАТУРА

1. *Авен О. И., Коган Я. А.* Управление вычислительным процессом в ЭВМ. М.: Энергия, 1978.
2. *Липаев В. В.* Распределение ресурсов в вычислительных системах. М.: Статистика, 1979.
3. *Липаев В. В., Яшков С. Ф.* Эффективность методов организации вычислительного процесса в АСУ. М.: Статистика, 1975.
4. Современное состояние теории исследования операций. М.: Наука, 1979.
5. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
6. *Рейнгольд Э., Пивергельт Ю., Део Н.* Комбинаторные алгоритмы. Теория и практика. М.: Мир, 1980.
7. *Ullman J. D.* NP-Complete Scheduling Problems. J. Computer and System Sci., 1975, vol. 10.
8. *Левнер Е. В., Генс Г. В.* Дискретные оптимизационные задачи и эффективные приближенные алгоритмы. Препринт, М.: ЦЭМИ АН СССР, 1978.
9. *Lenstra J. K., Rinnooy Kan A. H. G.* Complexity of Scheduling under Precedence Constraints. Oper Res., 1978, vol. 1, No. 1.
10. *Lenstra J. K.* Sequencing by Enumerative Methods. Mathematish Centrum, Amsterdam, 1977.
11. *Coffman E. G.*, Jr. (ed). Computer and Job / Shop Scheduling Theory, New York: J. Wiley and Sons, 1976.
12. *Генс Г. В., Левнер Е. В.* Об эффективных ε-алгоритмах для некоторых универсальных задач теории расписаний.– Изв. АН СССР, Техн. кибернет., 1978, № 6.
13. *Sahni S.* Algorithms for scheduling independent tasks. J. ACM, 1976, 23, N 1.
14. *Horowits E., Sahni S.* Exact and approximate algorithms for scheduling non-identical processors. J. ACM, 1976, 23.
15. *Ecker K.* Kritischer Vergleich von Algorithmen für ein Scheduling problem-Lecture Notes on Computer Sciences 1975, v. 34.
16. *Graham R. L., Lawler E. L., Lenstra J. K., Rinnooy Kan A. H. G.* Optimization and approximation in deterministic sequencing and scheduling: a survey Annals of Discrete Mathematics, 1979, N 5.
17. *Евреинов Э. В., Хорошевский В. Г.* Однородные вычислительные системы. Новосибирск: Наука, 1978.
18. *Конвей Р. В., Максвелл В. Л., Миллер Л. В.* Теория расписаний. М.: Наука, 1975.
19. *Танаев В. С., Шкурба В. В.* Введение в теорию расписаний. М.: Наука, 1975.
20. *Лившиц Э. М.* Исследование некоторых алгоритмов оптимизации сетевых моделей.– Экономика и математические методы. 1968, т. IV, вып. 5.

21. *Гольдгабер Е. М.* Задача минимизации времени исполнения проекта работ, заданного деревом.— Кибернетика, 1977, № 2.
22. *Гордон В. С., Танаев В. С.* Детерминированные системы обслуживания с одним прибором, древовидным упорядочением требований и экспоненциальными функциями штрафа.— В сб.: Вычислительная техника в машиностроении. Изд-во Ин-та технической кибернетики АН БССР, Минск, 1973.
23. *Бурдюк В. Я., Рева В. Н.* Об одном методе оптимизации функционалов от перестановок при наличии ограничений.— Кибернетика, 1980, № 1.
24. *Гордон В. С., Шафранский Я. М.* Оптимальное упорядочение при последовательно параллельных ограничениях предшествования.— Доклады АН БССР, 1978, т. 22, № 3.
25. *Левин М. Ш.* Об эффективном решении некоторых задач теории расписаний на сетях.— Кибернетика, 1980, № 1.
26. *Левин М. Ш., Левнер Е. В.* Об эффективном решении задачи Беллмана – Джонсона на сети, имеющей вид дерева.— Автоматика и телемеханика, 1978, № 10.
27. *Ху Т. С.* Параллельное упорядочение и проблемы линии сборки. Кибернетический сборник. Новая серия, 1967, вып. 4.
28. *Adolphson D. L.* Single Machine Sequencing with Precedence Constrants, SIAM J. Comput., 1977, 6.
29. *Adolphson D., Hu T. C.* Optimal Linear Ordering, SIAM J. Appl. Math., 1973, v. 25, No. 3.
30. *Coffman E. G., Graham R. L.* Optimal Scheduling for Two-Processor System, Acta Information, 1972, 1.
31. *Horn W. A.* Single-Machine Job Sequencing with Treelike Precedence Ordering and Linear Delay Penalties. SIAM J. Appl. Math., 1972, v. 23, No. 2.
32. *Lawler E. L.* Optimal Sequencing of a Single Machine Subject to Precedence Constraints, Management Sci., 1973, 19.
33. *Lawler E. L.*, Sivazlian B. D. Minimization of Time-Varying Costs in Single-Machine Scheduling, Oper Res., 1978, vol. 26, No. 4.
34. *Monma C. L., Sidney J. B.* A General Algorithms for Optimal Job Sequencing with Series-Parallel Precedence Constraints, Tech. Report., No. 347, School of Operations Research and Industrial Engineering, Cornell University, July, 1977.
35. *Smith W. E.* Various Optimizers for Single Stage Production Naval Res. Logist. Quart., 1956, vol. 3, No. 1.
36. *Carey M. R., Johnson D. S.* Computers and Intractability. A Guide to the Theory of NP-Completeness, San Francisco, W. H. Freeman and Co., 1979.
37. *Немировский А. С., Юдин Д. Б.* Сложность задач и эффективность методов оптимизации. М.: Наука, 1979.
38. *Нигматуллин Р. Г.* О приближенных алгоритмах с ограниченной абсолютной погрешностью для дискретных экстремальных задач.— Кибернетика, 1978, № 1.
39. *Лившиц Э. М.* Минимизация максимального штрафа в задаче одного станка.— В сб:. Труды I зимней школы по математическому программированию. М.: ЦЭМИ АН СССР, 1969, вып. 3.
40. *Генс Г. В., Левнер Е. В.* Эффективные приближенные алгоритмы для комбинаторных задач. Препринт. М.: ЦЭМИ АН СССР, 1981.
41. *Генс Г. В., Левнер Е. В.* Эффективные приближенные алгоритмы для комбинаторных задач: обзор.— Изв. АН СССР, Техн. кибернет., 1979, № 6.
42. *Турчина В. А.* Алгоритмы с оценками для задачи параллельного упорядочения.— В сб.: Краткие тезисы докладов VI Всесоюзного симпозиума Системы программного обеспечения решения задач оптимального планирования, М.: ЦЭМИ АН СССР, 1980.
43. *Карп Р. М.* Сводимость комбинаторных проблем. Кибернетический сборник. Новая серия. М.: Мир, 1975, вып. 12.
44. *Лившиц Э. М., Рублинецкий В. И.* О сравнительной сложности некоторых задач дискретной оптимизации.— В сб.: Вычислительная математика и вычислительная техника. Харьков: Физ.-техн. ин-т низких температур АН УССР, вып. 3, 1972.

Москва

Поступила в редакцию
26.VI.1981