Управляемые системы, ИМ СО АН СССР, 1988 г., вып. 28

УДК 519.1

ПОЛИНОМИАЛЬНЫЙ АЛГОРИТМ ПРИБЛИЖЕННОГО РЕШЕНИЯ ОДНОЙ ЗАДАЧИ ТЕОРИИ РАСПИСАНИЯ

В.А.Аксенов

Рассматриваемая задача теории расписаний, известная под названием **"open shop problem"** (OS-задача), формулируется следующим образом: имеется n станков и m деталей, известно время обработки j-й детали на i-м станке - t_{ij} ; требуется построить кратчайшее расписание обработки деталей на станках, удовлетворяющее следующим условиям:

а) каждая деталь должна быть обработана на каждом станке, при этом порядок обработки фиксированной детали на станках произволен;

б) одна деталь не может обрабатываться одновременно на двух и более станках;

в) на одном станке не могут обрабатываться одновременно две и более деталей.

Поиск такого расписания, несмотря на ослабляющее допущение о произвольном порядке обработки деталей на станках, представляет определенную трудность: в [1] доказано, что при $n, m \geq 3$ OS-задача относится к классу NP-полных.

Известны попытки решить OS-задачу за полиномиальное время при определенных ограничениях на исходные данные, в частности, в [2] показано существование алгоритма со сложностью $O(n^2m^3)$, находящего оптимальное решение в случае, когда

$$L > (16m'\log_2 m' + 5m')K_1 ,$$

где $K \geq t_{ij},\ i = 1,\ldots,n,\ j = 1,\ldots,m;\ L = \max_j \sum_{i=1}^{n} t_{ij}$ и $m' = 2^P$ такое, что $2^{P-1} < m \leq 2^P$.

При этих условиях доказано, то длина кратчайшего расписания равна L.

В предлагаемой работе, используя верхнюю оценку хроматической длины взвешенного графа [3], будет построен полиномиальный алгоритм нахождения приближенного решения OS-задачи при любых исходных данных.

Поставим в соответствие OS-задаче взвешенный граф $G(V,E,t)$ следующим образом:

$V = \{a_{ij}\},\ i \in N = \{1,\ldots,n\},\ j \in M = \{1,\ldots,m\}$ - множество вершин;

$E = \{(a_{ij}, a_{ks}) |$ либо $i = k$, либо $j = s\}$ - множество ребер;
$t(a_{ij}) = t_{ij}$ - вес вершины a_{ij}.

Расписанием графа $G(V, E, t)$ (далее обозначаемого G) будем называть множество открытых интервалов $\{\mathcal{I}_{ij}\}$ $i \in N, j \in M$ на числовой оси, удовлетворяющих следующим условиям:

а) $|\mathcal{I}_{ij}| = t_{ij}$,

б) $\mathcal{I}_{ij} \cap \mathcal{I}_{ks} = \emptyset$, если $(a_{ij}, a_{ks}) \in E$.

Длиной расписания будем называть величину $|\bigcup_{i,j} \mathcal{I}_{ij}|$.

При таких обозначениях длина кратчайшего расписания - $\min |\bigcup_{i,j} \mathcal{I}_{ij}|$, где минимум берется по всем расписаниям, по существу является определением хроматической длины [3] взвешенного графа, соответствующего рассматриваемой OS-задаче.

В [3] доказано, что хроматическая длина любого взвешенного графа ($\bar{\chi}(G)$) не превосходит веса самой "тяжелой" окрестности вершин графа ($\bar{\sigma}(G)$). В рассматриваемом здесь случае это означает, что

$$\bar{\chi}(G) \leq \max_{i,j} \Big(\sum_{k=1}^{n} t_{kj} + \sum_{s=1}^{m} t_{is} - t_{ij} \Big) = \bar{\sigma}(G). \qquad (1)$$

Очевидно, что хроматическая длина не меньше, чем вес самой тяжелой клики ($\bar{\varphi}(G)$). Для графа, соответствующего OS-задаче, это означает, что

$$\bar{\chi}(G) \geq \max \Big\{ \max_{i \in N} \sum_{s=1}^{m} t_{is}, \max_{j \in M} \sum_{k=1}^{n} t_{kj} \Big\} = \bar{\varphi}(G). \qquad (2)$$

Из (1) и (2) следует, что

$$\bar{\varphi}(G) \leq \bar{\chi}(G) \leq \bar{\sigma}(G) \leq 2\bar{\varphi}(G). \qquad (3)$$

Значит, конструктивное доказательство неравенства (1) в [3] содержит в себе алгоритм построения расписания OS-задачи, длина которого не более чем в два раза отличается от оптимального.

Рассмотрим этот алгоритм и оценим его сложность. В утверждении 1 из [3] доказано, что для любого независимого множества вершин A графа G существует расписание этого графа, для которого, во-первых, справедливо неравенство (1), во-вторых, выполнение расписания начинается с некоторого тупикового (нерасширяемого) независимого множества работ B, содержащего A. Индукционный шаг в доказательстве утверждения 1 из [3] состоит в уменьшении веса вершин из B на фиксированную величину, при этом уменьшается суммарный вес вершин графа и величина $\bar{\sigma}(G)$. Этот шаг индукции можно положить в основу алгоритма построения приближенного решения OS-задачи.

В процессе реализации расписания OS -задачи операция обработки j -й детали на i -м станке может находиться в одном из трех состояний: операция выполнена, выполняется, еще не начата. В соответствии с этим вершине a_{ij} графа G будем приписывать метки "2", "1" и "0".

Алгоритм состоит в построении последовательности взвешенных графов G^K с убывающим числом вершин. Для каждого G^K строится тупиковое независимое множество B^K вершин, которым будут соответствовать операции, начинающиеся в специально вычисляемые моменты b_K.

Графом G^0 является граф, представляющий исходную OS -задачу - $G(V,E,t)$; $B^0=\{a_{ii}\}$ $i\in N$ (без ограничения общности можно считать, что $m \geqslant n$); $b_0=0$, $J_{ii}=(0,t_{ii})$, $i\in N$, и $t^0(a)=t(a)$, $a\in V$. При этом вершины из множества B^0 получают метку "1", а все остальные - "0".

<u>$K+1$ шаг</u> алгоритма ($K\geqslant 0$) состоит из двух этапов:

<u>этап 1</u> включает в себя последовательное выполнение следующих операций:

1.1. вычисление величин $\Delta=\min\limits_{a\in B^K} t^K(a)$; $b_{K+1}=b_K+\Delta$,

1.2. изменение весов в графе G^K

$$t^{K+1}(a)=\begin{cases} t^K(a)-\Delta, & \text{если} \quad a\in B^K, \\ t^K(a), & \text{если} \quad a\notin B^K. \end{cases}$$

1.3. удаление из G^K вершин, для которых $t^{K+1}(x)=0$. Этим вершинам соответствуют законченные к моменту b_{K+1} операции обработки деталей на станках, поэтому удаляемым вершинам присваивается метка "2". В результате получаем взвешенный граф, который обозначаем G^{K+1}, кроме того, полагаем $A^K=\{x\in B^K \mid t^{K+1}(x)>0\}$.

<u>Этап 2</u> состоит в расширении множества A^K до тупикового независимого множества B^{K+1} в графе.

Для описания алгоритма построения множества B^{K+1} введем следующее разбиение на множестве станков и деталей: j -й детали приписывается метка "1" или "0" в зависимости от того, обрабатывается эта деталь в заданный момент времени на каком-либо станке или нет; аналогично i -му станку приписывается метка "1" или "0" в зависимости от того, обрабатывается на i -м станке какая-либо деталь или нет.

Построение множества B^{K+1} производится в момент b_{K+1}.

Рассмотрим станок с наименьшим номером (x) с меткой "0". Последовательно просматривая детали, ищем деталь y с меткой "0" и такую, что a_{xy} тоже имеет метку "0". Если такой детали нет, то переходим к следующему станку с меткой "0", в противном случае присваиваем станку, детали и вершине

a_{xy} метку "1" и переходим к следующему станку с меткой "0".

Этап 2 заканчивается, когда просмотрены все станки с меткой "0". Множество вершин, получивших метку "1", вместе с множеством A^K образуют тупиковое независимое множество B^{K+1}.

Весь алгоритм прекращает работу, когда все вершины получат метку "2".

Оценим теперь сложность алгоритма. Поскольку на каждом шаге метку "2" получает по меньшей мере одна вершина, то число шагов не превосходит nm. Все независимые множества в G содержат не более $n\,(n \leq m)$ вершин, поэтому для выполнения пп. 1.1, 1.2 и 1.3 и этапа 2 потребуется не более $O(n)$ операций. Значит, общая сложность алгоритма - $O(m \cdot n \cdot \min\{m,n\})$.

Поступила в ред.-изд. отдел
2 апреля 1987 г.

Литература

I. Gonzalez T., Sahni S. Open shop scheduling to minimize finish time //J. Assoc. Comput. Math. - 1976. - V.23.-P. 665-679.

2. Fiala T. An algorithm for the open shop problem. // Math. Oper. Res. - 1983. - V. 8, N.1. - P.100-109.

3. Аксенов В.А. Обобщение некоторых оценок хроматического числа графов // 30 Intern. Wiss. Koll. TH. Ilmenau. - 1985. - Vortragsriehe F.(Graphen und Netzwerke - Teorie und Anwendungen). - P. 3-5.