Задача об оптимальном расписании

Batyr Ishanov

26 ноября 2022 г.

1 Постановка задачи

Имеется множество работ J и множество машин M. Также задана функция $p : J \times M \to R_+$. Значение p_{ij} означает время выполнение i-й работы на j-й машине.

Требуется построить распределение работ по машинам так, чтобы все работы были выполнены и чтобы конечное время выполнения всех работ было минимально.

Рассмотрим частный случай задачи, когда производительности всех машин одинаковы, то есть $p_{ij} = p_i$.

Язык $\{(J, m, k) |$ на m одинаковых машинах выполнятся все задачи из J за время $\leq k\}$ будем называть PARALLEL-SCHEDULING.

2 NP-полнота

Доказательство **NP**-полноты состоит из двух пунктов: док-ва принадлежности к **NP** и док-ва **NP**-сложности.

2.1 Принадлежность к NP

Утв. *PARALLEL-SCHEDULING* $\in$ **NP**
Док-во. Сертификат - распределение задач по машинам, его проверка выполняется не более чем за полиномиальное время (суммируем время задач для каждой машины и проверяем, что оно $\leq k$). ■

2.2 NP-сложность

SUBSET-SUM$= \{(n_1, ..., n_l, N) \mid \exists \alpha \in \{0,1\}^l \sum \alpha_i n_i = N\}$
Доказывать **NP**-сложность будем через сведение SUBSET-SUM к нашему языку.

Утв. *SUBSET-SUM - **NP**-полный*
Док-во. Доказательство возьмем отсюда. ■

Утв. *SUBSET-SUM $\leq_p$ PARALLEL-SCHEDULING*
Док-во. Приведем функцию сводимости f. Пользоваться будем идеей, что SUBSET-SUM является частным случаем PARALLEL-SCHEDULING для 2 машин. Опишем f для 3х случаев:

1. $\sum_i n_i = 2N$
   Тогда $(n_1, ..., n_l, N) \longmapsto ((n_1, ..., n_l), 2, N)$. Если у нас есть подмножество $\{n_{i_1}, ..., n_{i_m}\}$ такое, что $\sum_{j=1}^{m} n_{i_j} = N$, то мы может дать эти задачи одной машине, а все остальные - другой. Каждая машина справится за N времени, значит, и суммарное время будет N. Если же такого подмножества нет, то какой-то машине достанется задач на $\geq N + 1$ времени, а значит, и суммарное время будет $\geq N + 1$.

2. $\sum_i n_i < 2N$

   Тогда $(n_1, ..., n_l, N) \longmapsto ((n_1, ..., n_l, 2N - \sum_i n_i), 2, N)$. Заметим, что сумма длительностей всех задач все так же $2N$, поэтому рассуждение работает точно так же, как в предыдущем пункте.

3. $\sum_i n_i > 2N$

   Тогда $(n_1, ..., n_l, N) \longmapsto ((n_1, ..., n_l, \sum_i n_i - 2N), 2, \sum_i n_i - N)$. Если у нас есть подмножество $\{n_{i_1}, ..., n_{i_m}\}$ такое, что $\sum_{j=1}^{m} n_{i_j} = N$, то мы может дать задачи $\{n_{i_1}, ..., n_{i_m}, \sum_i n_i - 2N)\}$ одной машине, а все остальные - другой. Если же такого подмножества нет, то какой-то машине достанется задач на $\geq \sum_i n_i - N + 1$ времени, а значит, и суммарное время будет $\geq \sum_i n_i - N + 1$. ■

3 Алгоритм вычисления

Список литературы

[Gra69] L.R. Graham. Bounds on multiprocessing timing anomalies. pages 423–425, 1969.

[IFM] Neerc IFMO. NP-полнота задачи о сумме подмножества.

[WS11] David P. Williamson and David B. Shmoys. The design of approximation Algorithms. pages 39–43, 2011.