

Обзор по задаче 1

18 апреля 2022 г.

Оглавление

Содержание

1	Введение	3
2	Критерии обзора	4
3	Конструктивные алгоритмы	5
3.1	Алгоритмы, основанные на нахождении максимального потока в сети	5
3.2	Алгоритмы, основанные на методе динамического программирования	5

1 Введение

Целями данного обзора являются: рассмотрение различных алгоритмов, решающих задачу построения многопроцессорного расписания минимальной длительности и выделение классов алгоритмов с учетом математических постановок задач из области исследования 1 и классов исходных данных для задач из области исследования 1.

Рассматриваемая задача построения многопроцессорного расписания минимальной длительности принадлежит к классу NP-hard, так как является частным случаем классической NP-hard задачи построения списочного расписания [Sha81]. Для задач из класса NP-hard не существует полиномиальных алгоритмов, оптимально решающих задачу. Однако, для некоторых задач из NP-hard удастся разработать псевдополиномиальные алгоритмы. На практике часто используют приближенные полиномиальные алгоритмы либо алгоритмы, специализированные под конкретные классы входных данных.

Алгоритмы построения многопроцессорных расписаний можно разделить на два класса, различающихся по используемым методам построения расписаний:

1. *Конструктивные алгоритмы.* Алгоритмы этого класса начинают процесс построения расписания с пустого расписания, не содержащего работ, а затем на каждом шаге добавляют работы в расписание.
2. *Итерационные алгоритмы.* Такие алгоритмы работают с полным расписанием (содержащим все работы). На каждом шаге такие алгоритмы изменяют расписание (при этом расписание остаётся полным), последовательно приближаясь к оптимальному решению.

Можно выделить следующие классы конструктивных алгоритмов:

- алгоритмы, основанные на нахождении максимального потока в транспортной сети;
- алгоритмы, основанные на методе ветвей и границ;
- алгоритмы, основанные на методе динамического программирования;
- алгоритмы, построенные на жадной схеме.

Можно выделить следующие классы итерационных алгоритмов:

- генетический алгоритм
- дифференциальная эволюция
- алгоритм имитации отжига
- алгоритм муравьиных колоний

Далее будут сформулированы критерии обзора, и согласно этим критериям приведены описания всех классов алгоритмов, представленных выше.

2 Критерии обзора

Ниже приведены критерии, по которым будут рассматриваться и сравниваться алгоритмы.

1. Отличие решаемой алгоритмом задач из области исследований
2. Сложность адаптации алгоритма к решаемой задаче.
3. Алгоритм итерационный или конструктивный.
4. Алгоритм является рандомизированным или детерминированным.
5. Порядок сложности алгоритма, возможность масштабируемости алгоритма на данные большой размерности.
6. На данных какой размерности протестирован алгоритм. Качество полученного результата. Время работы на протестированных данных.

3 Конструктивные алгоритмы

3.1 Алгоритмы, основанные на нахождении максимального потока в сети

Алгоритмы построения многопроцессорных расписаний, основанные на нахождении максимального потока в транспортной сети [Gol88] предполагают сведение задачи построения расписания к задаче поиска максимального потока в сети. На основе набора рабочих интервалов, процессоров, и ограничений исходной задачи (например, директивных сроков) строится сеть. Затем, для построенной сети решается задача поиска максимального потока, после чего, по построенному потоку в сети восстанавливается многопроцессорное расписание. Данный метод построения хорошо описан в литературе [S77], однако общим свойством задач, для которого он может применяться, является допустимость прерывания работ. В данной работе рассматривается задача построения расписаний, не допускающая прерывания выполнения рабочих интервалов, поэтому алгоритмы, основанные на нахождении максимального потока в сети неприменимы.

3.2 Алгоритмы, основанные на методе динамического программирования

Алгоритмы, основанные на методе динамического программирования [Bel66], требуют введения двух операций:

1. вложение решаемой задачи в семейство более простых задач;
2. нахождение возвратного соотношения, связывающего оптимальные значения этих подзадач.

Данные алгоритмы позволяют получить глобальный оптимум, но при этом для большинства задач комбинаторной оптимизации имеют не полиномиальную сложность. В частности, для NP-трудных задач построения расписаний [Hel62] сложность алгоритмов – факториальная функция от размера входа (число процессов, число рабочих интервалов и мощность множества исходного отношения частичного порядка). Более того, нахождение возвратного соотношения, связывающего оптимальные значения подзадач требует наличия аналитической модели вычислительной системы. Таким образом, данный алгоритм оказывается неприменим по Критерию 4 *Порядок сложности алгоритма*.

Список литературы

- [Hel62] Karp R. M Held M. «A Dynamic Programming Approach to Sequencing Problems». В: *Journal of the Society for Industrial and Applied Mathematics* 10 (1962).

- [Bel66] R. Bellman. «Dynamic programming». B: *Science* 153 (1966).
- [S77] Stone H. S. «Multiprocessor Scheduling with the Aid of Network Flow Algorithms». B: *IEEE Transactions on Software Engineering* SE-3.1 (1977).
- [Sha81] Tushkina T. A. Shakhbazyan K. V. «A survey of scheduling methods for multiprocessor systems». B: 1981.
- [Gol88] Tarjan R. E. Goldberg A. V. «A new approach to the maximum-flow problem». B: (1988).