

АЛГОРИТМ ТОЧНОГО РЕШЕНИЯ ДИСКРЕТНОЙ ЗАДАЧИ ВЕБЕРА
ДЛЯ ПРОСТОГО ЦИКЛА

Р. Э. Шангин

*Южно-Уральский государственный университет, г. Челябинск, Россия***E-mail:** shanginre@gmail.com

Предлагается полиномиальный алгоритм, находящий точное решение задачи Вебера в дискретной постановке для простого цикла и конечного множества позиций размещения, основанный на динамическом программировании. Проведен вычислительный эксперимент по анализу эффективности предложенного алгоритма в сравнении с пакетом IBM ILOG CPLEX.

Ключевые слова: задача Вебера, простой цикл, динамическое программирование, точный алгоритм.

Введение

Рассматривается дискретная задача Вебера [1, 2] для случая, когда размещаемый граф имеет вид простого цикла. Приведём математическую формулировку задачи.

Пусть $G = (J, E)$ — простой цикл, где J — множество вершин (размещаемые объекты); E — множество рёбер графа G (связи между размещаемыми объектами). Пусть V — конечное множество позиций (точек), предназначенных для размещения вершин графа G . Размещением вершин графа G назовём однозначное отображение $\pi : J \rightarrow V$, то есть вершина $i \in J$ размещается в позицию $\vartheta_i \in V$. Множество всех однозначных отображений множества вершин J в множество позиций V обозначим через $\Pi = \{\pi : J \rightarrow V\}$.

Стоимость размещения вершины $i \in J$ в множестве позиций V задаётся функцией $p : J \times V \rightarrow \mathbb{R}^+$, где $p(i, \vartheta_i)$ — стоимость размещения вершины $i \in J$ в позиции $\vartheta_i \in V$. Стоимость размещения ребра $[i, j] \in E$ на V^2 определяется функцией $c : E \times V^2 \rightarrow \mathbb{R}^+$, где $c([i, j], \vartheta_i, \vartheta_j)$ — стоимость размещения ребра $[i, j] \in E$ на V^2 при размещении его концевых вершин $i, j \in J$ в позициях $\vartheta_i, \vartheta_j \in V$ соответственно. Отметим, что зависимость между стоимостью размещения ребра $[i, j]$ и стоимостью размещения его концевых вершин i, j отсутствует.

Необходимо разместить вершины графа G в позициях множества V таким образом, чтобы суммарная стоимость размещения вершин и рёбер графа G была минимальной:

$$F(\pi) = \sum_{[i,j] \in E} c([i,j], \pi(i), \pi(j)) + \sum_{i \in J} p(i, \pi(i)) \rightarrow \min_{\pi \in \Pi}. \quad (1)$$

Заметим, что в дискретной постановке задачи Вебера размещаемые объекты рассматриваются как точки, а структура области, в которой производится размещение, является дискретной, то есть для размещения объектов указывается конечное количество позиций, причём ограничения на размещение объектов отсутствуют.

В общем случае дискретная задача Вебера является NP-трудной [3] и представляет собой релаксацию квадратичной задачи о назначениях [4], где условие инъективности

отображения множества вершин J графа G в конечное множество позиций размещения V снимается, то есть в дискретной задаче Вебера в одну позицию возможно размещение нескольких вершин графа [5–7].

Задача Вебера исследовалась в различных постановках, в том числе для непрерывной области размещения [8], в многокритериальной постановке [9] и др. Известны полиномиально разрешимые частные случаи. Для решения задачи Вебера на древовидной сети в непрерывной постановке, то есть когда допускается размещение объектов на дугах, разработаны полиномиальные алгоритмы [10]. В [11] предложен полиномиальный алгоритм решения минимаксной задачи Вебера на дереве. Предложен полиномиальный алгоритм для решения задачи Вебера для корневого дерева и конечного множества позиций размещения [6].

1. Точный алгоритм для решения дискретной задачи Вебера для цикла

Обозначим тройкой (G, V, F) рассматриваемую дискретную задачу Вебера (1), где $G = (J, E)$ — простой цикл, V — конечное множество позиций размещения и F — функция стоимости размещения графа G . Предлагается полиномиальный алгоритм CyWPA (Cycle Weber Problem Algorithm), основанный на динамическом программировании (ДП) и находящий оптимальное решение задачи (G, V, F) .

Введём следующие обозначения. Пусть $s \in J$ — произвольная вершина цикла G и $T = (I, W)$ — подграф графа G , индуцированный множеством вершин $J \setminus \{s\}$. Пусть $N = |I|$ — мощность множества вершин цепи T и $i_N \in I$ — висячая вершина цепи T . Выбор вершины i_N в качестве корня дерева $T = (I, W)$ индуцирует на множестве вершин I отношение частичного порядка

$$L = \{(i, j) : i, j \in I, j \text{ принадлежит цепи в } T \text{ между } i \text{ и } i_N\}. \quad (2)$$

В дальнейшем будем считать, что $I = \{1, 2, \dots, N\}$ и выполняется условие $(l, m) \in L \Rightarrow l < m$, где $l, m \in I$. Обозначим G_i — подграф графа G , индуцированный множеством вершин $\{s, 1, 2, \dots, i-1, i\}$. На рис. 1 представлены цепь T и подграф G_3 в цикле G .

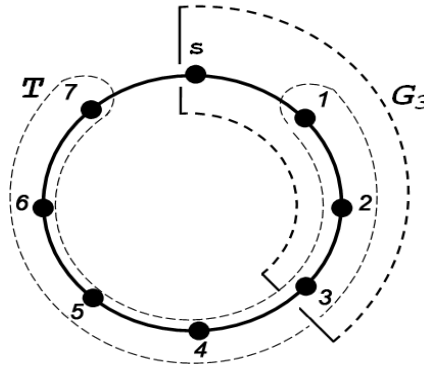


Рис. 1. Цепь T и подграф G_3 в простом цикле G

Положим, что $G(\vartheta_s)$, $\vartheta_s \in V$, есть граф G , в котором вершина s размещена в позицию ϑ_s . Исходную задачу (G, V, F) разобьём на ряд подзадач $(G(\vartheta_s), V, F)$ для любых $\vartheta_s \in V$, где тройка $(G(\vartheta_s), V, F)$ есть дискретная задача Вебера. Решение каждой подзадачи $(G(\vartheta_s), V, F)$, в свою очередь, разбивается на $N + 1$ шагов процесса ДП. Значение функции Беллмана $f_i(\vartheta_i)$, вычисленное на шаге i процесса ДП решения подзадачи $(G(\vartheta_s), V, F)$ для некоторого состояния $\vartheta_i \in V$, есть стоимость оптимального размещения подграфа G_i в множестве позиций V , когда размещение вершин s и i в множестве V

равно ϑ_s и ϑ_i соответственно. На шаге i процесса ДП решения подзадачи $(G(\vartheta_s), V, F)$ для каждого состояния $\vartheta_i \in V$ определяется также множество $V(\vartheta_i)$ оптимального размещения вершин подграфа G_i в множестве позиций V , когда размещение вершин s и i в множестве V равно ϑ_s и ϑ_i соответственно.

Алгоритм СуWPA

Э т а п 0. Выбрать произвольную вершину $s \in J$. Определить подграф $T = (I, W)$ и на множестве его вершин задать отношение порядка (2). Для каждого $\vartheta_s \in V$ определить граф $G(\vartheta_s)$.

Э т а п 1. Для каждого $\vartheta_s \in V$ решить подзадачу $(G(\vartheta_s), V, F)$.

Шаг 1 процесса ДП. Для каждого состояния $\vartheta_1 \in V$ вычислить значение функции Беллмана

$$f_1(\vartheta_1) = p(1, \vartheta_1) + c([1, s], \vartheta_1, \vartheta_s) + p(s, \vartheta_s)$$

и определить множество размещения

$$V(\vartheta_1) = \{\vartheta_1\} \cup \{\vartheta_s\}.$$

Шаг i , $i = 2, 3, \dots, N - 1$, процесса ДП. Для каждого $\vartheta_i \in V$ и любого $\vartheta_{i-1} \in V$ вычислить значение функции $R(\vartheta_i, \vartheta_{i-1})$ стоимости оптимального размещения подграфа G_i в множестве позиций V , когда вершины s, i и $i - 1$ размещены в позициях ϑ_s, ϑ_i и ϑ_{i-1} соответственно:

$$R(\vartheta_i, \vartheta_{i-1}) = p(i, \vartheta_i) + c([i, i - 1], \vartheta_i, \vartheta_{i-1}) + f_{i-1}(\vartheta_{i-1}).$$

Для каждого состояния $\vartheta_i \in V$ вычислить значение функции Беллмана

$$f_i(\vartheta_i) = \min_{\vartheta_{i-1} \in V} \{R(\vartheta_i, \vartheta_{i-1})\} \quad (3)$$

и определить множество размещения

$$V(\vartheta_i) = \{\vartheta_i\} \cup V(\vartheta_{i-1}^*), \quad \vartheta_{i-1}^* = \arg \min_{\vartheta_{i-1} \in V} \{R(\vartheta_i, \vartheta_{i-1})\}. \quad (4)$$

Шаг N процесса ДП. Для каждого $\vartheta_{i_N} \in V$ и любого $\vartheta_{i_N-1} \in V$ вычислить значение функции $R(\vartheta_{i_N}, \vartheta_{i_N-1})$ по формуле

$$R(\vartheta_{i_N}, \vartheta_{i_N-1}) = p(i_N, \vartheta_{i_N}) + c([i_N, i_N - 1], \vartheta_{i_N}, \vartheta_{i_N-1}) + f_{N-1}(\vartheta_{i_N-1}) + c([i_N, s], \vartheta_{i_N}, \vartheta_s).$$

Для каждого состояния $\vartheta_{i_N} \in V_N$ вычислить значение функции Беллмана $f_N(\vartheta_{i_N})$ и определить множество $V(\vartheta_{i_N})$ по формулам (3) и (4) соответственно.

Шаг $N + 1$ процесса ДП. Определить оптимальное размещение $\pi_{G(\vartheta_s)}$ вершин графа $G(\vartheta_s)$ и его стоимость $F_{G(\vartheta_s)}$ по формулам

$$\pi_{G(\vartheta_s)} = V(\vartheta_{i_N}^*), \quad \vartheta_{i_N}^* = \arg \min_{\vartheta_{i_N} \in V} \{f_N(\vartheta_{i_N})\}, \quad (5)$$

$$F_{G(\vartheta_s)} = \min_{\vartheta_{i_N} \in V} \{f_N(\vartheta_{i_N})\}.$$

Э т а п 2. После того, как оптимальное решение подзадач $(G(\vartheta_s), V, F)$ для каждого $\vartheta_s \in V$ найдено, определить оптимальное решение π_G^* исходной задачи (G, V, F) :

$$\pi_G^* = \pi_{G(\vartheta_s^*)}, \quad \vartheta_s^* = \arg \min_{\vartheta_s \in V} \{F_{G(\vartheta_s)}\}. \quad (6)$$

Стоп.

Теорема 1. Алгоритм СуWPA находит точное решение задачи Вебера (G, V, F) , где $G = (J, E)$ — простой цикл; V — конечное множество позиций размещения.

Доказательство. Докажем, что подзадача $(G(\vartheta_s), V, F)$ при каждом фиксированном размещении $\vartheta_s \in V$ решается алгоритмом СуWPA оптимально. Очевидно, на шаге 2 процесса ДП для любых $\vartheta_s, \vartheta_2 \in V$ алгоритм находит оптимальное размещение вершины 1 относительно заданного размещения ϑ_s, ϑ_2 вершин s и 2, причём в данном случае это достигается путем полного перебора всех способов размещения ϑ_1 вершины 1 в множестве позиций V . На шаге 3 процесса ДП для любых $\vartheta_s, \vartheta_3 \in V$ алгоритм находит оптимальное размещение вершин 1 и 2 относительно заданного размещения ϑ_s, ϑ_3 вершин s и 3, поскольку для вершины 1 оптимальное размещение определено на предыдущем шаге 1 процесса ДП для каждого заданного размещения ϑ_2 вершины 2, а оптимальное размещение вершины 2, согласно формуле (4), находится перебором способов размещения ϑ_2 в множестве позиций V с целью минимизации стоимости размещения подграфа G_3 , когда размещение вершин s и 3 в V равно ϑ_s и ϑ_3 соответственно.

Аналогично, на последующих шагах $i = 4, 5, \dots, N$ динамического процесса для любых $\vartheta_s, \vartheta_i \in V$ алгоритм находит оптимальное размещение вершин $1, 2, \dots, i - 1$ относительно заданного размещения ϑ_s, ϑ_i вершин s и i .

Таким образом, рекуррентно определяя оптимальное размещение вершин подграфа G_i на шагах $i = 2, 3, \dots, N$, алгоритм на шаге $N + 1$ процесса ДП, согласно формуле (5), зная оптимальное размещение вершин $1, 2, \dots, N - 1$ относительно любых заданных размещений ϑ_s и ϑ_{i_N} , находит оптимальное размещение графа $G(\vartheta_s)$.

Поскольку размещение графа $G(\vartheta_s)$ для любых $\vartheta_s \in V$ найдено оптимально, то алгоритм на этапе 2, осуществив полный перебор способов размещения ϑ_s вершины s в множестве позиций V согласно формуле (6), находит оптимальное решение исходной задачи (G, V, F) . ■

Теорема 2. Вычислительная сложность алгоритма СуWPA не превосходит $O(|V|^3(|J| - 2))$ операций. Пространственная сложность равна $O(|V|^2)$ памяти.

Доказательство. Определим оценку числа операций, требуемых для решения одной подзадачи $(G(\vartheta_s), V, F)$, $\vartheta_s \in V$. На шаге 1 процесса ДП необходимо выполнить $O(|V|)$ операций, так как $O(|V|)$ операций требуется для вычисления значений функции Беллмана $f_1(\cdot)$ и $O(|V|)$ операций — для определения множеств размещения $V(\cdot)$ вершин подграфа G_1 . На каждом из последующих шагов $i = 2, 3, \dots, N$ процесса ДП требуется выполнить $O(|V|^2)$ операций, поскольку $O(|V|^2)$ операций необходимо для вычисления значений функции $R(\cdot)$, столько же — для вычисления значений функции Беллмана $f_i(\cdot)$ и столько же — для определения множеств размещения $V(\cdot)$ вершин подграфа G_i . На шаге $N + 1$ требуется выполнить $O(|V|)$ операций, так как столько операций необходимо для определения множества $\pi_{G(\vartheta_s)}$ оптимального размещения вершин графа $G(\vartheta_s)$ и для вычисления значения функции $F_{G(\vartheta_s)}$ стоимости такого оптимального размещения. В соответствии с этим для решения одной подзадачи $(G(\vartheta_s), V, F)$ требуется $O(|V|^2(|J| - 2))$ вычислительных операций, где

$$O(|V|^2(|J| - 2)) = O(|V|) + O(|V|^2)(|J| - 2) + O(|V|).$$

Заметим, что для определения множества π_G^* оптимального размещения вершин графа G требуется $O(|V|)$ операций. Исходя из того, что количество решаемых подзадач $(G(\vartheta_s), V, F)$ равно $|V|$, а для решения одной подзадачи требуется $O(|V|^2 \cdot (|J| - 2))$

операций, оценка вычислительной сложности алгоритма CyWPA равна $O(|V|^3 \cdot (|J| - 2))$ операций.

Определим оценку пространственной сложности алгоритма. Положим, что для хранения множества $V(\cdot)$ требуется $O(|V|)$ памяти. При решении любой подзадачи $(G(\vartheta_s), V, F)$ на шаге 1 процесса ДП требуется $O(|V|^2)$ памяти, так как $O(|V|^2)$ памяти требуется для хранения множеств $V(\vartheta_1)$, $\vartheta_1 \in V$, и $O(|V|)$ памяти — для хранения массива $f_1(\cdot)$, где для хранения одного значения функции $f_i(\cdot)$ требуется $O(1)$ памяти. На каждом из последующих шагов $i = 2, 3, \dots, N$ процесса ДП требуется $O(|V|^2)$ памяти, поскольку $O(|V|^2)$ памяти требуется для хранения множеств $V(\vartheta_{i-1})$, $\vartheta_{i-1} \in V$, и массива $f_{i-1}(\cdot)$, полученных на предыдущем шаге $i - 1$ процесса ДП; $O(|V|^2)$ памяти требуется для хранения массива $R(\vartheta_i, \vartheta_{i-1})$, $\vartheta_i, \vartheta_{i-1} \in V$, и $O(|V|^2)$ памяти — для хранения множеств $V(\vartheta_i)$, $\vartheta_i \in V$, и массива $f_i(\cdot)$. На шаге $N + 1$ требуется $O(|V|^2)$ памяти, так как столько памяти требуется для хранения множеств $V(\vartheta_N)$, $\vartheta_N \in V$, и массива $f_N(\cdot)$, полученных на предыдущем шаге N процесса ДП, и $O(|V|)$ памяти требуется для хранения множества $\pi_{G(\vartheta_s)}$ оптимального размещения вершин графа $G(\vartheta_s)$. Заметим, что на каждом шаге $i = 2, 3, \dots, N$ процесса ДП для определения множеств $V(\vartheta_i)$, $\vartheta_i \in V$, и массива $f_i(\cdot)$ требуются только соответствующие множества $V(\vartheta_{i-1})$, $\vartheta_{i-1} \in V$, и массив $f_{i-1}(\cdot)$, полученные на предыдущем шаге $i - 1$, поэтому необходимость хранения в памяти на шаге $i = 2, 3, \dots, N$ массивов $V(\vartheta_j)$, $\vartheta_j \in V$, и $f_j(\cdot)$, определённых на всех предшествующих шагах $j = i - 2, i - 3, \dots$ процесса ДП, отпадает.

На этапе 2 для определения множества π_G^* оптимального размещения вершин графа G требуется $O(|V|^2)$ памяти, поскольку $O(|V|)$ памяти требуется для хранения одного множества размещения $\pi_{G(\vartheta_s)}$, а количество хранимых в памяти множеств $\pi_{G(\vartheta_s)}$ на этапе 2 равно $|V|$. В соответствии с этим оценка пространственной сложности предложенного алгоритма CyWPA равна $O(|V|^2)$ памяти. ■

2. Экспериментальное исследование эффективности алгоритма CyWPA

Алгоритм CyWPA реализован на ЭВМ в среде MATLAB. Проведён вычислительный эксперимент по анализу его эффективности. Для оценки эффективности алгоритма использовался программный пакет IBM ILOG CPLEX Optimization Studio 12.2 (решение модели целочисленного линейного программирования (ЦЛП) дискретной задачи Вебера алгоритмом ветвей и границ с ограничением по времени работы).

Для проведения эксперимента был случайным образом с равномерным распределением сгенерирован класс тестовых задач, состоящий из серий, каждая из которых включает 30 задач одинаковой размерности. Вычисления проводились на компьютере с процессором Intel Pentium 1.86 GHz.

Результаты вычислительного эксперимента приведены в таблице, где $\bar{t}_{\text{alg}}$ — среднее время работы предложенного алгоритма, с; $\bar{t}_{\text{ILP}}$ — среднее время работы модели ЦЛП, реализованной в среде IBM ILOG CPLEX Optimization Studio 12.2, с.

Результаты эксперимента

$ J $	$ V $	$\bar{t}_{\text{alg}}$	$\bar{t}_{\text{ILP}}$
5	5	0,0167	0,0071
10	10	0,0854	0,6927
20	20	0,9139	15,3793
40	40	9,2132	—
100	100	386,8167	—

«—» — решение не удалось получить за приемлемое время (1000 с)

Для дискретной задачи Вебера для простого цикла $G = (J, E)$ и конечного множества позиций размещения V размерности $|J| = 40$, $|V| = 40$ и выше не удалось получить решение с помощью пакета IBM ILOG CPLEX за приемлемое время (1000 с); среднее время решения задачи Вебера такой размерности с помощью предлагаемого алгоритма CyWPA не превысило 10 с. Для задачи Вебера размерности $|J| = 20$, $|V| = 20$ среднее время работы предлагаемого алгоритма не превысило 1 с, тогда как среднее время работы пакета IBM ILOG CPLEX составило 16 с. Стоит отметить, что среднее время решения задачи Вебера размерности $|J| = 5$, $|V| = 5$ и ниже с помощью предложенного алгоритма CyWPA значительно превзошло среднее время работы пакета IBM ILOG CPLEX.

Заключение

Предложен полиномиальный алгоритм, основанный на динамическом программировании, находящий оптимальное решение дискретной задачи Вебера, когда размещаемый граф имеет вид простого цикла. Определены оценки вычислительной и пространственной сложности предложенного алгоритма.

Проведён вычислительный эксперимент по анализу эффективности предложенного алгоритма в сравнении с пакетом IBM ILOG CPLEX Optimization Studio 12.2. Среднее время решения дискретной задачи Вебера размерности $|J| = 100$, $|V| = 100$ для простого цикла и конечного множества позиций размещения с помощью предложенного алгоритма CyWPA не превысило 400 с, в то время как с помощью пакета IBM ILOG CPLEX 12.2 не удалось получить решение задачи размерности $|J| = 40$, $|V| = 40$ за 1000 с. Из результатов эксперимента следует, что применение предложенного алгоритма перспективно для решения дискретной задачи Вебера для простого цикла средней и большой размерности.

ЛИТЕРАТУРА

1. Шангин Р. Э. Исследование эффективности приближенных алгоритмов решения одного частного случая задачи Вебера // Экономика, статистика и информатика. Вестник УМО. 2012. № 1. С. 163–168.
2. Панюков А. В., Пельцвергер Б. Ф. Оптимальное размещение дерева в конечном множестве // Журн. вычисл. математики и матем. физики. 1988. Т. 28. № 2. С. 618–620.
3. Панюков А. В. Модели и методы решения задач построения и идентификации геометрического размещения: дис. ... д-ра физ.-мат. наук. Челябинск, 1999. 260 с.
4. Сергеев С. А. Квадратичная задача о назначениях I // Автоматика и телемеханика. 1999. № 8. С. 127–147.
5. Панюков А. В., Пельцвергер Б. Ф., Шафир А. Ю. Оптимальное размещение точек ветвления транспортной сети на цифровой модели местности // Автоматика и телемеханика. 1990. № 9. С. 153–162.
6. Panyukov A. V. and Pelzwerger B. V. Polynomial algorithms to finite Weber problem for a tree network // J. Comput. Appl. Math. 1991. V. 35. P. 291–296.
7. Шангин Р. Э. Разработка и анализ алгоритмов для задачи Вебера // Проблемы оптимизации и экономические приложения. Омск, 2012. С. 121–122.
8. Трубин В. А. Эффективный алгоритм для задачи Вебера с прямоугольной метрикой // Кибернетика. 1978. № 6. С. 67–70.
9. Zabudsky G. G. and Filimonov D. V. An algorithm for minimax location problem on tree with maximal distances // Proc. Second Intern. Workshop “Discrete Optimization Methods in Production and Logistics” (DOM2004). Omsk, 2004. P. 81–85.

10. Забудский Г. Г., Филимонов Д. В. О минимаксной и минисуммной задачах размещения на сетях // Труды XII Байкальской Междунар. конф. «Методы оптимизации и их приложения». Омск, 2001. С. 150–155.
11. Филимонов Д. В. Решение дискретной минимаксной задачи размещения на древовидной сети // Материалы ежегодного научного семинара аспирантов. Омск, 2003. С. 58–61.