UDC 519.863

Algorithm For Solving Weber Problem For a N-Sequentially Connected Chain

Roman E. Shangin

South Ural state university, Russia
454080, Cheliabinsk, 76 ave. Lenina, department VMI
PhD student
E-mail: shanginre@gmail.com

Abstract. Here is set the algorithm, which reasonably solves Veber problem for n-sequentially connected chain and finite set of points of location. Comparison of action period of a given algorithm and a model of integer linear programming, which was realized in IBM ILOG CPLEX, is carried out.

Keywords: Weber problem; n-sequentially connected chain; dynamic programming; exact algorithm.

Введение. Рассматривается задача Вебера в дискретной постановке [1] для неориентированной n-последовательносвязной цепи и конечного множества позиций размещения. Приведем математическую формулировку исследуемой задачи для произвольного неориентированного графа.

Пусть $G=(J,E)$ – неориентированный граф без петель и кратных ребер, где $J=\{i\}$ – множество вершин графа, $E=\{(i,j): i, j \in J\}$ – множество ребер графа G. Пусть V – конечное множество позиций, предназначенных для размещения вершин графа G. Размещением вершин графа G назовем однозначное отображение $\pi : J \to V$, то есть вершина $i \in J$ размещается в позицию $\vartheta_i \in V$, причем в одну позицию возможно размещение нескольких вершин графа.

Стоимость размещения вершины $i \in J$ в множестве позиций V задается функцией $p: J \times V \to R^+$, где соответственно $p(i, \vartheta_i)$ – функция стоимости размещения вершины $i \in J$ в позиции $\vartheta_i \in V$. Стоимость размещения ребра $(i,j) \in E$ на V^2 определяется функцией $c: E \times V^2 \to R^+$, где $c((i,j), \vartheta_i, \vartheta_j)$ – функция стоимости 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.

Необходимо разместить вершины графа G в позициях множества V таким образом, чтобы стоимость размещения графа G была минимальной. Математическая формулировка задачи в терминах отображений имеет вид

$$F(\pi) = \sum_{\{i,j\} \in E} c((i,j), \pi(i), \pi(j)) + \sum_{i \in J} p(i, \pi(i)) \to \min_{\pi}. \qquad (1)$$

Задача Вебера в данной постановке, в общем случае, является NP-трудной [1]. Задача Вебера исследовалась в различных постановках, в том числе для непрерывной области размещения [2], в многокритериальной постановке [3] и др. Известны полиномиально разрешимые частные случаи задачи Вебера (1).

В настоящей работе рассматривается частный случай задачи (1), когда структура связей между размещаемыми объектами задается графом, имеющим вид n-последовательносвязной цепи. Рассматриваемый частный случай имеет большую практическую значимость, так как структура многих производственно-технологических процессов в топливно-энергетическом, металлургическом, машиностроительном и др. комплексах может быть представлена n-последова-тельносвязной цепью.

Определение и свойства *n*-последовательносвязной цепи

Пусть $N(j)$ – множество вершин графа $G=(J,E)$, смежных с вершиной j. Пусть $\varphi(G)$ – плотность графа G. Будем полагать, что на множестве J введена нумерация и каждая вершина отождествлена с присвоенным ей номером.

Определение. Связный граф $G=(J,E)$ называется n-последовательно-связной цепью (n-sequentially connected chain), если на множестве его вершин можно задать такую нумерацию, что для любой вершины графа G с номером j, имеет место равенство

$$N(j)=\{(j-n),...,(j-1),(j+1),...,(j+n)\}\cap\{1,2,...,|J|\}:n=\varphi(G)-1.$$

Отметим, свойства n-последовательносвязной цепи подробно рассмотрены в [4], в частности доказано, что n-последовательносвязная цепь является триангулированным графом и представляет собой частный случай k-дерева.

Алгоритм решения задачи Вебера для *n*-последовательносвязной цепи

Обозначим тройкой (G,V,F) рассматриваемую задачу Вебера (1), где $G=(J,E)$ – n-последовательносвязная цепь (далее n-пс цепь), V – конечное множество позиций размещения и F – функция стоимости размещения графа G. Предлагается квазиполиномиальный алгоритм **ScChWPA** (Sequentially connected Chain Weber Problem Algorithm), основанный на динамическом программировании (ДП), находящий оптимальное решение задачи (G,V,F).

Идея алгоритма **ScChWPA** заключается в следующем. На множестве вершин n-пс цепи G задается нумерация. Каждая вершина графа отождествляется с присвоенным ей порядковым номером. Процесс решения задачи (G,V,F) разбивается на $(|J|-n)+2$ шагов процесса ДП.

Введем следующие обозначения. Пусть $G(i)=(J(i),E(i))$ – подграф графа G, индуцированный вершинами, номера которых принадлежат множеству $\{1,2,...,i+n\}$. Пусть K_i – клика подграфа $G(i)$ размера n, номера вершин которой принадлежат множеству $\{i+1,i+2,...,i+n\}$. Пусть $\pi(K_i)=\{\pi'(K_i)\}$ – множество всех однозначных отображений вершин клики K_i в множестве V.

Обозначим $V(i)=\{V(\pi'(K_i)):\pi'(K_i)\in\pi(K_i)\}$ – множество состояний процесса ДП на шаге i, где под состоянием $V(\pi'(K_i))$ понимается оптимальное размещение вершин подграфа $G(i)$ в множестве позиций V, когда размещение вершин его клики K_i в множестве V равно $\pi'(K_i)$. Определяется функция Беллмана $f_i(\cdot)$ для процесса ДП решения задачи (G,V,F). Значение функции $f_i(V(\pi'(K_i)))$, вычисленное на шаге i процесса ДП для некоторого состояния $V(\pi'(K_i))\in V(i)$, есть стоимость оптимального размещения подграфа $G(i)$ в множестве позиций V, когда размещение вершин клики K_i в V равно $\pi'(K_i)$.

АЛГОРИТМ **ScChWPA.**

Шаг 0 (начальный) процесса ДП.

Шаг 0.1. Определить подграф $G(0)$; определить клику K_0;

Шаг 0.2. Определить множество $\pi(K_0)=\{\pi'(K_0)\}$;

Шаг 0.3. Определить множество $V(0)=\{V(\pi'(K_0)):\pi'(K_0)\in\pi(K_0)\}$.

Шаг 0.4. Для каждого состояния $V(\pi'(K_0)) \in V(0)$ вычислить соответствующее значение функции Беллмана $f_0(V(\pi'(K_0)))$ по формуле

$$f_0(V(\pi'(K_0))) = \sum_{\substack{j\in K_0,\\ \vartheta_j \in \pi'(K_0)}} p(j,\vartheta_j) + \sum_{\substack{j,s\in K_0,\\ \vartheta_j,\vartheta_s \in \pi'(K_0)}} c((j,s),\vartheta_j,\vartheta_s).$$

Перейти на шаг 1 процесса ДП.

Шаг $i : 1 \le i \le (|J| - n)$ процесса ДП.

Шаг i.1. Определить подграф $G(i)$; определить клику K_i.

Шаг i.2. Определить множество $\pi(K_i) = \{\pi'(K_i)\}$;

Шаг i.3. Для каждого $\vartheta_i \in V$ и любых $\pi'(K_i) \in \pi(K_i)$ определить множество $T(\pi'(K_i),\vartheta_i)$ оптимального размещения вершин подграфа $G(i)$ в множестве позиций V, когда его вершина с номером i размещена в позицию ϑ_i, а размещение вершин клики $K_i \in J(i)$ в V равно $\pi'(K_i)$:

$$T(\pi'(K_i),\vartheta_i) = V(\pi'(K_{i-1})) \cup \{\vartheta_{i+n}\} : \pi'(K_{i-1}) = \vartheta_i \cup (\pi'(K_i)/\{\vartheta_{i+n}\}),$$

где $i \in K_{i-1}$ и $i+n \in K_i$, причем $\vartheta_i \in \pi'(K_{i-1})$ и $\vartheta_{i+n} \in \pi'(K_i)$.

Шаг i.4. Для каждого множества размещения $T(\pi'(K_i),\vartheta_i)$, определенного на шаге i.3, вычислить значение функции $R(T(\pi'(K_i),\vartheta_i))$ стоимости соответствующего размещения по формуле

$$R(T(\pi'(K_i),\vartheta_i)) = p(i+n,\vartheta_{i+n}) + \\ + \sum_{\substack{j\in K_{i-1},\\ \vartheta_j \in \pi'(K_{i-1})}} c((i+n,j),\vartheta_{i+n},\vartheta_j) + f_{i-1}(V(\pi'(K_{i-1}))),$$

где размещение $\pi'(K_{i-1}) = \vartheta_i \cup (\pi'(K_i)/\{\vartheta_{i+n}\})$ и $\vartheta_i \in \pi'(K_{i-1})$, $\vartheta_{i+n} \in \pi'(K_i)$.

Шаг i.5. Определить множество $V(i) = \{V(\pi'(K_i)) : \pi'(K_i) \in \pi(K_i)\}$ состояний процесса ДП на шаге i, где некоторое состояние $V(\pi'(K_i)) \in V(i)$ определяется согласно формуле

$$V(\pi'(K_i)) = \arg\min_{\vartheta_i \in V}\{R(T(\pi'(K_i),\vartheta_i))\}. \qquad (2)$$

Шаг i.6. Для каждого состояния $V(\pi'(K_i)) \in V(i)$ вычислить значение функции Беллмана $f_i(V(\pi'(K_i)))$ по формуле

$$f_i(V(\pi'(K_i))) = \min_{\vartheta_i \in V}\{R(T(\pi'(K_i),\vartheta_i))\}.$$

Перейти на шаг $i+1$ процесса ДП.

Шаг $(|J| - n) + 1$ (конечный) процесса ДП. Найти оптимальное размещение π^* вершин графа G в множестве позиций V по формуле

$$\pi^* = \arg\min_{\pi'(K_{|J|-n}) \in \pi(K_{|J|-n})}\{V(\pi'(K_{|J|-n}))\}. \qquad (3)$$

Стоп.

Вычислительная сложность предложенного алгоритма **ScChWPA** не превосходит $O(|V|^{n+1} \cdot (|J| - n))$ операций. Пространственная сложность алгоритма равна $O(|V|^{n+2})$ памяти.

Теорема. Алгоритм **ScChWPA** находит точное решение задачи Вебера (G,V,F), где G – n-пс цепь, V – конечное множество позиций размещения.

Алгоритм **ScChWPA**, находящий точное решение задачи (G,V,F), является квазиполиномиальным, поскольку при некотором фиксированном значении входного параметра n, он оказывается полиномиальным.

Вычислительный эксперимент

Алгоритм **ScChWPA** был реализован на ЭВМ. Проведен 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) задачи Вебера алгоритмом ветвей и границ с ограничением по времени работы).

Для проведения эксперимента был случайным образом с равномерным распределением сгенерирован класс задач, состоящий из серий, каждая из которых включала 30 задач одинаковой размерности.

Результаты вычислительного эксперимента приведены в таблице, где $\bar{t}$ – среднее время работы предложенного алгоритма (сек.); $\bar{f}$ – среднее время работы модели ЦЛП (сек.), реализованной в IBM ILOG CPLEX.

Таблица.

Результаты вычислительного эксперимента

Величина n	Размерность задачи					
		J=5,V=5	J=10,V=10	J=20,V=20	J=40,V=40	J=100,V=100
n=1	$\bar{t}$	0,0016	0,0044	0,0233	0,2379	3,3854
	$\bar{f}$	0,1421	0,5342	13,1563	–	–
n=2	$\bar{t}$	0,0213	0,0264	0,5964	9,5264	374,5622
	$\bar{f}$	0,1525	0,5719	12,4813	–	–
n=3	$\bar{t}$	0,0931	0,3526	11,9642	383,2657	–
	$\bar{f}$	0,1424	0,6391	17,5722	–	–
n=4	$\bar{t}$	0,1762	3,2752	39,5721	–	–
	$\bar{f}$	0,1121	0,9877	17,3413	–	–
n=5	$\bar{t}$	0,2651	39,4758	–	–	–
	$\bar{f}$	0,1499	1,0012	17,4887	–	–

«–» решение не удалось получить за приемлемое время (1000 сек.)

Для задач Вебера для n-пс цепи размерности $|J|=40, |V|=40$ и выше не удалось получить решение с помощью модели ЦЛП, реализованной в среде IBM ILOG CPLEX, за приемлемое время для любых значениях параметра n, притом, что среднее время решения задач Вебера такой размерности для 1(2)-пс цепи с помощью предлагаемого алгоритма не превысило одной (десяти) секунд соответственно.

Применение алгоритма **ScChWPA** для решения задачи Вебера для n-пс цепи при $n \geq 4$ нецелесообразно, так как время работы данного алгоритма превосходит время работы модели ЦЛП в среде IBM ILOG CPLEX. Установлено: чем меньше величина n и чем больше количество вершин размещаемого графа, тем алгоритм **ScChWPA** более эффективен по сравнению с алгоритмом ветвей и границ.

Примечания:

1. Panyukov A.V., Pelzwerger B.V. Polynomial Algorithms to Finite Veber Problem for a Tree Network / A. V. Panyukov, B. V. Pelzwerger // Journal of Computational and Applied Mathematics. 1991. Vol 35. P. 291-296.

2. Трубин В.А. Эффективный алгоритм для задачи Вебера с прямоугольной метрикой / В. А. Трубин // Кибернетика. 1978. № 6. С. 67-70.
3. Zabudsky G.G., Filimonov D. V. An algorithm for minimax location problem on tree with maximal distances / G. G. Zabudsky, D. V. Filimonov // Proc. of the Second International Workshop "Discrete Optimization Methods in Production and Logistics" (DOM2004). Omsk-Irkutsk. 2004. P. 81-85.
4. Шангин Р.Э. О некоторых свойствах n-последовательносвязной цепи / Р.Э. Шангин // Вестник ЮУрГУ. Серия Вычислительная математика и информатика. 2012. №2. С. 130-138.

УДК 519.863

Алгоритм для решения задачи Вебера для N-последовательносвязной цепи

Роман Эдуардович Шангин

Южно-Уральский государственный университет, Россия
54080, г. Челябинск., пр. Ленина, 76, факультет ВМИ
Аспирант
E-mail: shanginre@gmail.com

Аннотация. Предлагается алгоритм, находящий точное решение задачи Вебера для n-последовательносвязной цепи и конечного множества позиций размещения. Проведен вычислительный эксперимент по анализу эффективности предложенного алгоритма в сравнении с пакетом IBM ILOG CPLEX.

Ключевые слова: задача Вебера; n-последовательносвязная цепь; динамическое программирование; точный алгоритм.