ДЕТЕРМИНИРОВАННЫЕ АЛГОРИТМЫ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ВЕБЕРА ДЛЯ N-ПОСЛЕДОВАТЕЛЬНОСВЯЗНОЙ ЦЕПИ

В работе вводятся понятия n-последовательносвязного графа, а также одного его частного случая: n-последовательносвязной цепи. Проводится анализ свойств n-последовательносвязной цепи.

Рассматривается задача Вебера для n-последовательносвязной неориентированной цепи $G = (J, E)$ и конечного множества точек V:

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

где $\varphi : J \to V$ – однозначное отображение из множества J в V; J – множество объектов размещения; $E = \{(i,j) : i,j \in J\}$ – множество ребер графа G; $p : J \times V \to R^+ : \ p(i, \vartheta_i)$ – функция стоимости размещения вершины $i \in J$ в точке $\vartheta_i \in V$; $c : A \times V^2 \to R^+ : \ c(\{i,j\}, \vartheta_i, \vartheta_j) : (i,j) \in E, \vartheta_i, \vartheta_j \in V$ – функция стоимости размещения ребра графа G на V^2.

Задача Вебера в данной постановке, в общем случае, является NP-трудной [1] и представляет собой частный случай квадратичной задачи о назначениях (КЗН), где условие инъективности отображения φ из множества J вершин графа в множество V точек размещения снимается, т.е. в одну точку возможно размещение нескольких вершин графа [2].

Необходимо отметить, что задача Вебера исследовалась в различных постановках, в том числе для непрерывной области размещения [3], в многокритериальной постановке [4] и др.

В работе предложен квазиполиномиальный алгоритм, корректно решающий задачу Вебера для n-последовательносвязной цепи и конечного множества точек размещения.

На классе задач, сгенерированном случайным образом, проведено сравнение времени работы предложенного алгоритма и модели целочисленного линейного программирования (ЦЛП), реализованной в среде IBM ILOG CPLEX 12.3.

1.1 Определения и свойства n-последовательносвязных графов

Вводится понятие n-последовательносвязного графа. Пусть $T_G(j)$ – множество вершин графа G, смежных с вершиной j.

Определение

Граф $G = (J, E)$ называется n-последовательносвязным (n-sequentially connected graph), если на множестве его вершин можно задать такое отношение порядка, что для любой вершины графа G с номером j, номера вершин из множества $T_G(j)$ принадлежат множеству $\{(j-n), .., (j-1), (j+1), .., (j+n)\}$.

Далее приводится понятие одного частного случая n-последовательносвязного графа: n-последовательносвязной цепи.

Определение

Граф $G = (J, E)$ называется n-последовательносвязной цепью (n-sequentially connected chain), если на множестве его вершин можно задать такое отношение порядка, что для любой вершины графа G с номером j, множество

$$T_G(j) \subseteq \{(j-n), .., (j-1), (j+1), .., (j+n)\},$$

притом, что существуют такие две вершины $i, k : i, k \in J$, что

$$(\forall j \in \{i, k\})\ |T_G(j)| = n.$$

В дальнейшем такие вершины i и k будем именовать крайними вершинами n-последовательносвязной цепи.

На рисунке 1 для различных n представлены n-последовательносвязные цепи.

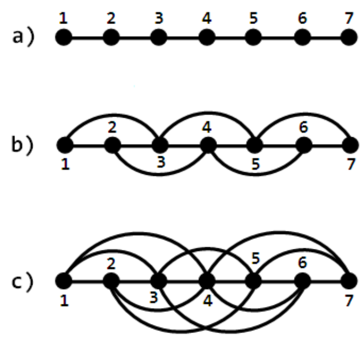

Рисунок 1 – n-последовательносвязные неориентированные цепи: a) 1-последовательносвязная цепь; b) 2-последовательносвязная цепь; c) 3-последовательносвязная цепь.

Т.е. 1-последовательносвязная цепь представляет собой обыкновенную цепь, т.к. для любых j $|T_G(j)| \leq 2$ и каждая вершина такой цепи связана не более чем с одной предшествующей вершиной. В 2(3)-последовательносвязной цепи каждая вершина связана не более чем с двумя (тремя) предыдущими вершинами соответственно.

Свойства n-последовательносвязной цепи $G = (J, E)$

1. Число ребер графа G равно:

$$|E| = n \cdot \left(|J| - \left(\frac{n+1}{2}\right)\right);$$

2. Размер максимальной клики графа G равен $n + 1$, причем существует $|J| - n$ таких максимальных клик;

3. Число вершинной связности графа G равно n, т.е. любая n-последовательносвязная цепь является n-связным графом;

4. Хроматическое число χ_G графа G равно $n + 1$;

5. Цикломатическое число C_G графа G равно:

$$C_G = \left(|J| - \frac{n}{2} - 1\right) \cdot (n - 1).$$

1.2 Точный квазиполиномиальный алгоритм для решения задачи Вебера для n-последовательносвязной цепи

Рассматривается задача (G, V, F), где $G = (J, E)$ – n-последовательносвязная цепь и V – конечное множество точек размещения. Предлагается квазиполиномиальный алгоритм $\mathbf{ScChVPA}^{(n)}$ (Sequentially connected Chain Veber Problem Algorithm), использующий идею динамического программирования (ДП), корректно решающий задачу (G, V, F).

Идея алгоритма $\mathbf{ScChVPA}^{(n)}$ заключается в следующем. На множестве вершин графа G задается отношение порядка. Каждая вершина графа отождествляется с присвоенным ей порядковым номером. Процесс решения исходной задачи (G, V, F) разбивается на $|J|$ шагов процесса ДП. Переход от одного шага динамического процесса к другому рассматривается как последовательный перебор вершин графа G в соответствии с их нумерацией.

Для каждого шага i процесса ДП определяется множество $J_i = \{1, .., i\}$ номеров вершин, размещаемых на шаге i. Так для нулевого (исходного) шага $J_0 = \emptyset$ – ни одна вершина не размещается, на первом шаге $J_1 = \{1\}$ – размещается только вершина с номером 1, на $|J|$-ом (конечном) шаге $J_{|J|} = \{1, .., |J|\}$ – размещаются все вершины графа G.

Также для каждого шага i процесса ДП определяется множество $K_i = \{i, i-1, .., i-n+1\} : K_i \subseteq J_i$ номеров вершин графа G, для которых на шаге i хранятся все возможные комбинации их размещений на множестве V. Множество K_i определяется согласно следующей рекуррентной формуле

$$(\forall i \in \{1, 2, ..., |J|\}) \qquad K_i = \{i\} \cup K_{i-1} \setminus \{i-n\}.$$

Т.е. для каждого шага $i = n, n+1, ..., |J|$ процесса ДП $|K_i| = n$, а для любого шага $j = 1, 2, ..., n-1$ процесса ДП $|K_j| = j$, причем $K_0 = \emptyset$.

Стоит заметить, что множество K_{i-1} есть ничто иное, как множество номеров предшествующих вершин, смежных с вершиной i, т.е. $K_{i-1} = T'_G(i)$.

Под состоянием на шаге i процесса ДП понимается множество $V_{(\vartheta_i, .., \vartheta_{i-n+1})}$ размещений вершин, номера которых принадлежат множеству J_i, при условии, что вершины с номерами из множества $\{i, i-1, .., i-n+1\} = K_i$ размещены в точках $\{\vartheta_i, \vartheta_{i-1}, ..., \vartheta_{i-n+1}\} \subseteq V$ соответственно.

$$(\forall \vartheta_i, \vartheta_{i-1}, ..., \vartheta_{i-n+1} \in V) \qquad V_{(\vartheta_i, .., \vartheta_{i-n+1})} = \{\vartheta_i, \vartheta_{i-1}, ..., \vartheta_{i-n+1}, \vartheta_{i-n}, ..., \vartheta_1\}.$$

Определяется функция Беллмана для динамического процесса. Краевые условия:

$$f_0(\cdot) = 0, \; f_{|J|+1}(\cdot) = F(\varphi^*_G),$$

где φ^*_G – оптимальное размещение вершин графа G на множестве V.

На шаге i процесса ДП для некоторого состояния $V_{(\vartheta_i, .., \vartheta_{i-n+1})}$ рекуррентная функция Беллмана имеет вид

$$f_i(V_{(\vartheta_i, .., \vartheta_{i-n+1})}) = p(i, \vartheta_i) +$$

$$+ \min_{\vartheta_{i-n} \in V} \left\{ \sum_{j \in J_{i-1}, \vartheta_j \in V_{(\vartheta_{i-1}, .., \vartheta_{i-n})}} c([i, j], \vartheta_i, \vartheta_j) + f_{i-1}(V_{(\vartheta_{i-1}, .., \vartheta_{i-n})}) \right\}.$$

Т.е. согласно формуле (.) для некоторого состояния $V_{(\vartheta_i,..,\vartheta_{i-n+1})}$ на шаге i процесса ДП осуществляется перебор вариантов размещения $\vartheta_{i-n} \in V$ вершины графа G с номером $i-n \in K_{i-1}$ с целью минимизации стоимости размещения всех вершин из множества J_i, при условии, что вершины из множества $K_i = \{i, i-1, .., i-n+1\}$ размещены соответственно в точках $\{\vartheta_i, \vartheta_{i-1}, ..., \vartheta_{i-n+1}\} \subseteq V$.

Множество размещений $V_{(\vartheta_i,..,\vartheta_{i-n+1})}$ определяется в соответствии с формулой

$$V_{(\vartheta_i,..,\vartheta_{i-n+1})} = \{\vartheta_i\} \cup V_{\vartheta_{i-1},..,\vartheta^*_{i-n}} : \vartheta^*_{i-n} =$$

$$= \arg \min_{\vartheta_{i-n} \in V} \left\{ \sum_{j \in J_{i-1}, \vartheta_j \in V_{(\vartheta_{i-1},..,\vartheta_{i-n})}} c([i,j], \vartheta_i, \vartheta_j) + f_{i-1}(V_{(\vartheta_{i-1},..,\vartheta_{i-n})}) \right\}.$$

Далее для шагов $t = i+1, i+2, ..., |J|$ рекуррентно вычисляются функции Беллмана $f_t(V_{(\vartheta_t,..,\vartheta_{t-n+1})})$ и определяются соответствующие множества размещений $V_{(\vartheta_t,..,\vartheta_{t-n+1})}$ для любых $\vartheta_t, .., \vartheta_{t-n+1} \in V$ согласно формулам () и ().

После завершения $|J|$-го (конечного) шага процесса ДП алгоритм **ScChVPA**$^{(n)}$ получает оптимальное размещение φ^*_G вершин графа G и значение функции стоимости $F(\varphi^*_G)$ оптимального размещения в соответствии с формулами

$$F(\varphi^*_G) = \min_{\vartheta_{|J|},..,\vartheta_{|J|-n+1} \in V} \left\{ f_{|J|}(V_{(\vartheta_{|J|},..,\vartheta_{|J|-n+1})}) \right\};$$

$$\varphi^*_G = V_{\vartheta^*_{|J|},..,\vartheta^*_{|J|-n+1}} : \vartheta^*_{|J|}, .., \vartheta^*_{|J|-n+1} = \arg \min_{\vartheta_{|J|},..,\vartheta_{|J|-n+1} \in V} \left\{ f_{|J|}(V_{\vartheta_{|J|},..,\vartheta_{|J|-n+1}}) \right\}.$$

Оценка пространственной сложности алгоритма **ScChVPA**$^{(n)}$ равна $O(|J| \cdot |V|^{n+1})$, так как для хранения одного множества $V_{(\vartheta_i,..,\vartheta_{i-n+1})}$ требуется $O(|J| \cdot |V|)$ памяти. Оценка вычислительной сложности равна $O(|J| \cdot |V|^{n+1})$ операций.

Теорема 1. Алгоритм **ScChVPA**$^{(n)}$ корректно решает задачу Вебера (G, V, F), где G – n-последовательносвязная цепь.

Доказательство. Разобьем исходную задачу (G, V, F), где G – n-последовательносвязная цепь на последовательность подзадач $(G^i_{(\vartheta_i,..,\vartheta_{i-n+1})}, V, F) : \vartheta_i, .., \vartheta_{i-n+1} \in V$ для $i = 1, 2, ..., |J|$, где $G^i_{(\vartheta_i,..,\vartheta_{i-n+1})}$ – подграф графа G с множеством вершин $J^i_{(\vartheta_i,..,\vartheta_{i-n+1})} = \{1, ..., i\}$ и множеством ребер $E^i_{(\vartheta_i,..,\vartheta_{i-n+1})} = \{(j,l) : j, l \in J_i, (j,l) \in E\}$, при условии, что для любых $j \in \{i, i-1, ..., i-n+1\}$ $\varphi(j) = \vartheta_j$.

Некоторая подзадача $(G^i_{(\vartheta_i,..,\vartheta_{i-n+1})}, V, F)$, решаемая на шаге i процесса ДП заключается в оптимальном размещении вершин графа G с номерами из множества $\{1, 2, ..., i\} = J_i$ в точках множества V, при условии, что вершины из множества $\{i, i-1, .., i-n+1\} = K_i$ размещены соответственно в точках $\{\vartheta_i, \vartheta_{i-1}, ..., \vartheta_{i-n+1}\} \subseteq V$.

Очевидно, что на шаге $n+1$ процесса ДП некоторое множество $V_{(\vartheta_{n+1},..,\vartheta_2)}$ представляет собой оптимальное решение подзадачи $(G^{n+1}_{(\vartheta_{n+1},..,\vartheta_2)}, V, F)$ для любых $\vartheta_{n+1}, .., \vartheta_2 \in V$, т.к.

$$V_{(\vartheta_{n+1},..,\vartheta_2)} = \{\vartheta_{n+1}\} \cup V_{\vartheta_n,..,\vartheta^*_1} : \vartheta^*_1 =$$

$$= \arg \min_{\vartheta_1 \in V} \left\{ \sum_{j \in J_n, \vartheta_j \in V_{(\vartheta_n,..,\vartheta_1)}} c([n+1,j], \vartheta_{n+1}, \vartheta_j) + f_n(V_{(\vartheta_n,..,\vartheta_1)}) \right\},$$

и $T''_G(1) = K_{n+1}$, где для любых $j \in J$ множество $T''_G(j) = T_G(j) \setminus \{j-n, ..., i-1\}$.

Естественно, на последующих шагах $i = 2, 3, ..., |J|$ процесса ДП алгоритм **ScChVPA**$^{(n)}$ для любых $\vartheta_i, .., \vartheta_{i-n+1} \in V$ оптимально решает некоторую подзадачу $(G^i_{(\vartheta_i,..,\vartheta_{i-n+1})}, V, F)$ зная оптимальные решения $V_{(\vartheta_{i-1},..,\vartheta_{i-n})}$ соответствующих подзадач $(G^i_{(\vartheta_i,..,\vartheta_{i-n+1})}, V, F)$ для всех $\vartheta_{i-n} \in V$, полученные на предыдущем шаге $i-1$ процесса ДП, т.к множество $V_{(\vartheta_i,..,\vartheta_{i-n+1})}$ определяется в соответствии с рекуррентной формулой () и $T''_G(i-n) = K_i$.

После завершения $|J|$-го (конечного) шага процесса ДП, алгоритм **ScChVPA**$^{(n)}$ получает оптимальное решение φ^*_G исходной задачи (G, V, F), так как подзадачи $(G^{|J|}_{(\vartheta_{|J|},..,\vartheta_{|J|-n+1})}, V, F)$ для любых $\vartheta_{|J|}, .., \vartheta_{|J|-n+1} \in V$ решены оптимально и

$$\varphi^*_G = V_{\vartheta^*_{|J|},..,\vartheta^*_{|J|-n+1}} : \vartheta^*_{|J|}, .., \vartheta^*_{|J|-n+1} = \arg \min_{\vartheta_{|J|},..,\vartheta_{|J|-n+1} \in V} \left\{ f_{|J|}(V_{\vartheta_{|J|},..,\vartheta_{|J|-n+1}}) \right\}.$$

Теорема 1 доказана.

1.3 Вычислительный эксперимент

Алгоритм **ScChVPA**$^{(n)}$ был реализован на ЭВМ. Проведен вычислительный эксперимент с целью сравнение времени работы предложенного алгоритма и модели целочисленного линейного программирования (ЦЛП), реализованной в среде IBM ILOG CPLEX 12.3.

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

Результаты вычислительного эксперимента приведены в таблице 1, где $\bar{t}_{alg}$ – среднее время работы предложенного алгоритма, сек.; $\bar{t}_{CLP}$ – среднее время работы модели ЦЛП, сек.

Таблица 1 – Результаты вычислительного эксперимента

Величина $\boldsymbol{n}$		**Размерность задачи**				
		$\lvert J\rvert, \lvert V\rvert = 5$	$\lvert J\rvert, \lvert V\rvert = 10$	$\lvert J\rvert, \lvert V\rvert = 20$	$\lvert J\rvert, \lvert V\rvert = 40$	$\lvert J\rvert, \lvert V\rvert = 100$
$\boldsymbol{n}$=1	$\bar{t}_{alg}$	0,0006	0,0044	0,0233	0,2379	3,3854
	$\bar{t}_{CLP}$	0,1421	1,5342	123,1563	–	–
$\boldsymbol{n}$=2	$\bar{t}_{alg}$	0,0021	0,0264	0,5964	9,5264	374,5622
	$\bar{t}_{CLP}$	0,7525	2,5719	252,4813	–	–
$\boldsymbol{n}$=3	$\bar{t}_{alg}$	0,0231	0,3526	11,9642	383,2657	–
	$\bar{t}_{CLP}$	0,8421	2,8391	282,5722	–	–
$\boldsymbol{n}$=4	$\bar{t}_{alg}$	0,0562	3,2752	239,5721	–	–
	$\bar{t}_{CLP}$	1,1121	2,9877	312,3413	–	–
$\boldsymbol{n}$=5	$\bar{t}_{alg}$	0,2351	39,4758	–	–	–
	$\bar{t}_{CLP}$	1,1499	3,0012	317,4887	–	–

«–» – решение не удалось получить за приемлемое время.

Для задач Вебера для n-последовательносвязной цепи размерности $|J| = 40, |V| = 40$ и выше не удалось получить решение с помощью модели ЦЛП за приемлемое время, притом, что среднее время решения задач Вебера такой размерности для 1(2)-последовательносвязных цепей с помощью алгоритма **ScChVPA**$^{(n)}$ не превысило одной(десяти) секунд соответственно. Применение алгоритма **ScChVPA**$^{(n)}$ для решения задачи размещения n-последовательносвязной цепи при $n \geq 5$ нецелесообразно, так как время работы данного алгоритма существенно превосходит время работы модели ЦЛП, реализованной в среде IBM ILOG CPLEX 12.3.

Заключение

В работе введены понятия n-последовательносвязного графа, а также одного его частного случая: n-последовательносвязной цепи. Проведен анализ свойств n-последовательносвязной цепи. Предложен квазиполиномиальный детерминированный алгоритм $\mathbf{ScChVPA}^{(n)}$, корректно решающий задачу Вебера для n-последовательносвязной цепи и конечного множества точек размещения. Доказана теорема об оптимальности решения задачи Вебера для n-последовательносвязной цепи алгоритмом $\mathbf{ScChVPA}^{(n)}$.

Проведен вычислительный эксперимент с целью сравнение времени работы предложенного алгоритма и модели целочисленного линейного программирования (ЦЛП), реализованной в среде IBM ILOG CPLEX 12.3. Исходя из результатов вычислительного эксперимента следует, что алгоритм $\mathbf{ScChVPA}^{(n)}$ целесообразней всего использовать для решения задач Вебера для n-последовательносвязной цепи при $n \leq 3$.

ЛИТЕРАТУРА

1. Panyukov A. V. Polynomial algorithms to finite Veber problem for a tree network / A. V. Panyukov, B.V. Pelzwerger // Journal of Computational and Applied Mathematics, Vol. 35 (1991). P. 291-296. (North-Holland)
2. Шангин Р.Э. Исследование эффективности приближенных алгоритмов решения одного частного случая задачи Вебера // Экономика, статистика и информатика. Вестник УМО.– Москва, 2012.– №1.–С.163-169.
3. Picard J.C. A cut approach to the rectilinear distance facility location problem / J.C. Picard, D.H. Ratli // Oper. Res. - 1978. - Vol. 26, № 3. - P. 422-433.
4. Zabudsky G.G. 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.

Шангин Роман Эдуардович, Южно-Уральский государственный университет, пр. им. В.И. Ленина, 76, Челябинск, 454080, Россия. E-mail: shanginre@gmail.com