

УДК 519.863

ДЕТЕРМИНИРОВАННЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ ВЕБЕРА ДЛЯ n -ПОСЛЕДОВАТЕЛЬНОСВЯЗНОЙ ЦЕПИ

Р. Э. Шангин

Аннотация. Вводится класс n -последовательностно связанных цепей. Предлагается алгоритм, находящий точное решение задачи Вебера для n -последовательностно связанной цепи и конечного множества позиций размещения, основанный на динамическом программировании. Дан теоретический анализ предложенного алгоритма. На классе задач, сгенерированном случайным образом, проведён вычислительный эксперимент по анализу эффективности предложенного алгоритма в сравнении с пакетом IBM ILOG CPLEX.

Ключевые слова: задача Вебера, n -последовательностно связанная цепь, динамическое программирование, точный алгоритм.

Введение

Рассматривается задача Вебера в дискретной постановке [5, 15] для неориентированной n -последовательностно связанной цепи и конечного множества позиций размещения. Приведём математическую формулировку исследуемой задачи для произвольного неориентированного графа [6].

Пусть $G = (J, E)$ — неориентированный граф без петель и кратных рёбер, J и $E = \{(i, j) \mid i, j \in J\}$ — множества вершин и рёбер графа G , а V — конечное множество позиций (точек), предназначенных для размещения его вершин. *Размещением вершин графа G* назовём однозначное отображение $\pi : J \rightarrow V$, т.е. вершина $i \in J$ размещается в позицию $\vartheta_i \in V$.

Стоимость размещения вершины $i \in J$ в позиции $\vartheta_i \in V$ задаётся функцией $p(i, \vartheta_i)$. Стоимость размещения ребра $(i, j) \in E$ на V^2 определяется функцией $c : E \times V^2 \rightarrow R^+$, 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 так, что суммарная стоимость размещения вершин и рёбер графа G ми-

нимальна. Математическая формулировка задачи в терминах отображений имеет вид

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

В данной постановке задачи Вебера размещаемые объекты рассматриваются как материальные точки, а структура области размещения дискретна, т. е. для размещения объектов указывается конечное количество позиций, причём ограничения на размещение объектов отсутствуют. Стоит заметить, что в рассматриваемой постановке задачи Вебера (1) учитывается стоимость размещения вершин графа в позиции, а в классической — не учитывается.

Задача Вебера в данной постановке в общем случае NP-трудна [4] и представляет собой релаксацию квадратичной задачи о назначениях [2, 7], где условие инъективности отображения π из множества вершин графа J в конечное множество позиций размещения V снимается, т. е. в одну позицию возможно размещение нескольких вершин графа [5, 6, 10].

Задача Вебера исследовалась в различных постановках, в том числе для непрерывной области размещения [8], в многокритериальной постановке [16] и др. Несмотря на NP-трудность задачи Вебера известны полиномиально разрешимые частные случаи. Для её решения на древовидной сети в непрерывной постановке, т. е. когда допускается размещение объектов на дугах, разработаны полиномиальные алгоритмы [3]. В [9] предложен полиномиальный алгоритм решения минимаксной задачи Вебера на дереве. Предложен полиномиальный алгоритм для решения задачи Вебера для корневого дерева и конечного множества мест размещения [15].

В настоящей работе рассматривается частный случай задачи (1), когда структура связей между размещаемыми объектами задаётся графом, имеющим вид n -последовательностно-связной 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 -последовательностно-связной цепью [11]. Отметим, что предлагаемый алгоритм анонсирован в [13].

1. Определение и свойства n -последовательностно-связной цепи

Пусть $N(j)$ — множество вершин графа $G = (J, E)$, смежных с вершиной j . Пусть $\varphi(G)$ — плотность графа G . Будем полагать, что на мно-

жестве вершин J введена нумерация и каждая вершина отождествлена с присвоенным ей номером.

Определение 1. Связный граф $G = (J, E)$ называется n -последовательносвязной цепью, если на множестве его вершин можно задать нумерацию такую, что

$$N(j) = \{(j-n), \dots, (j-1), (j+1), \dots, (j+n)\} \cap \{1, 2, 3, \dots, |J|\} : n = \varphi(G) - 1$$

для любой вершины графа G с номером j .

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

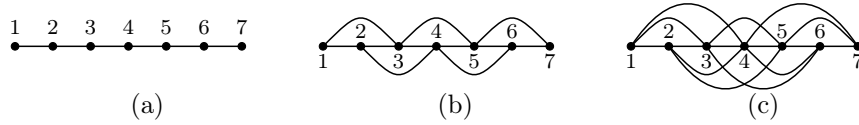


Рис. 1. Неориентированные n -последовательносвязные цепи:
(а) 1-, (б) 2- и (с) 3-последовательносвязная цепь

1-Последовательносвязная цепь представляет собой простую цепь, так как $|N(j)| \leq 2$ для любых j . Заметим, что каждая вершина такой цепи связана с одной предшествующей вершиной, где под предшествующей вершиной понимается вершина с меньшим номером. В 2- (3-)последовательносвязной цепи каждая вершина связана не более чем с двумя (тремя) предшествующими вершинами. Имеет место следующее конструктивное определение n -последовательносвязной цепи.

Определение 2. Связный неориентированный граф G называется n -последовательносвязной цепью, если его построение возможно осуществить рекурсивно по правилам: полный граф из $n + 1$ вершин есть n -последовательносвязная цепь G , n -последовательносвязная цепь с $i + 1$ вершинами получается из n -последовательносвязной цепи G с i вершинами путём добавления в неё новой вершины с номером $i + 1$ и n рёбер так, что новая вершина смежна со всеми вершинами с номерами $(i - n) + 1, (i - n) + 2, \dots, i$.

Отметим, что свойства n -последовательносвязной цепи подробно рассмотрены в [11], в частности, доказано, что n -последовательносвязная цепь является триангулированным (хордальным) графом [14] и представляет собой частный случай k -дерева [1].

Отношение класса n -последовательносвязных цепей к другим классам графов представлено графически на рис. 2.

Класс n -последовательнoсвязных цепей принадлежит классам совершенных и триангулированных графов, причём n -последовательнoсвязные цепи с числом вершин $|J| = n + 1$ принадлежат классу полных графов, а с числом вершин $|J| = n + 2$ — классу расщепляемых графов.



Рис. 2. Отношения классов графов

2. Точный алгоритм решения задачи Вебера для n -последовательнoсвязной цепи

Через (G, V, F) обозначим рассматриваемую задачу Вебера (1), в которой $G = (J, E)$ — n -последовательнoсвязная цепь, V — конечное множество позиций размещения и F — функция стоимости размещения графа G . Предлагается детерминированный 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 G задаётся нумерация согласно определению 1. Каждая вершина графа отождествляется с присвоенным ей порядковым номером. Процесс решения задачи (G, V, F) разбивается на $|J| - n + 2$ этапов процесса ДП. Переход от одного этапа к другому по сути состоит в последовательном переборе вершин G в соответствии с их нумерацией.

Введём следующие обозначения. Пусть $G_i = (J_i, E_i)$ — подграф графа G , индуцированный вершинами с номерами $1, 2, \dots, i + n$, $0 \leq i \leq |J| - n$. Заметим, что подграф $G_{(|J|-n)}$ равен графу G , где под равенством графов понимается совпадение их множеств вершин и рёбер, в данном случае $J_{(|J|-n)} = J$ и $E_{(|J|-n)} = E$. Пусть K_i — клика подграфа G_i размера n с вершинами $\{i + 1, i + 2, \dots, i + n\}$ (рис. 3).

Пусть $\pi(K_i) = \{\pi'(K_i)\}$ — множество всех однозначных отображений вершин клики K_i в множество позиций V , где $\pi'(K_i) = \{\vartheta_{i+1}, \vartheta_{i+2}, \dots,$

ϑ_{i+n} — способ размещения вершин клики K_i в множестве позиций V . Отметим, что $|\pi(K_i)| = |V|^n$ для любых i , $0 \leq i \leq |J| - n$.

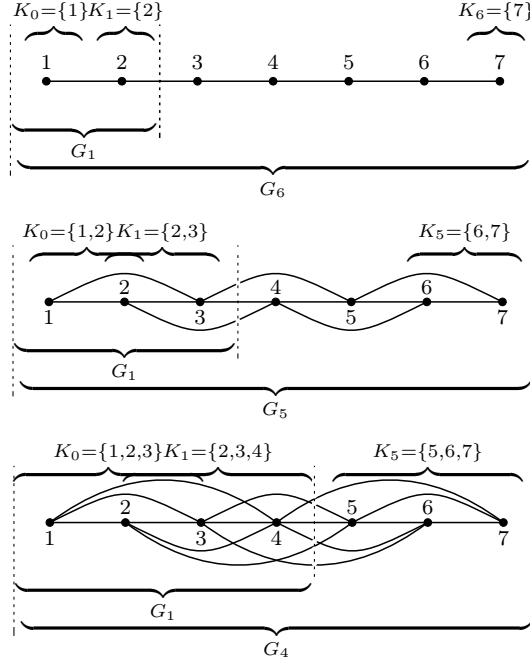


Рис. 3. Подграфы G_i и клики K_i в n -последовательносвязных цепях

Пусть $\pi(K_i) = \{\pi'(K_i)\}$ — множество всех однозначных отображений вершин клики K_i в множество позиций V , где $\pi'(K_i) = \{\vartheta_{i+1}, \vartheta_{i+2}, \dots, \vartheta_{i+n}\}$ — способ размещения вершин клики K_i в множестве позиций V . Отметим, что $|\pi(K_i)| = |V|^n$ для любых i , $0 \leq i \leq |J| - n$.

Обозначим через $V_i = \{V(\pi'(K_i)) \mid \pi'(K_i) \in \pi(K_i)\}$ множество состояний процесса ДП на этапе i , где под $V(\pi'(K_i))$ 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 этапе i для состояния $V(\pi'(K_i)) \in V_i$, равно стоимости оптимального размещения подграфа G_i в множестве позиций V , когда $\pi'(K_i)$ — размещение вершин его клики K_i в множестве V .

АЛГОРИТМ ScChWPA.

ЭТАП 0 (НАЧАЛЬНЫЙ) ПРОЦЕССА ДП.

ШАГ 1. Определить подграф G_0 ; определить клику K_0 .

ШАГ 2. Определить множество $\pi(K_0) = \{\pi'(K_0)\}$ всех однозначных отображений вершин клики K_0 в множество позиций V .

ШАГ 3. Определить множество состояний

$$V_0 = \{V(\pi'(K_0)) \mid \pi'(K_0) \in \pi(K_0)\}.$$

Отметим, что на этапе 0 процесса ДП $V(\pi'(K_0)) = \pi'(K_0)$ для любых $\pi'(K_0) \in \pi(K_0)$.

ШАГ 4. Для каждого состояния $V(\pi'(K_0)) \in V_0$ вычислить значение функции Беллмана $f_0(V(\pi'(K_0)))$:

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

Перейти на этап 1 процесса ДП.

ЭТАП i , $1 \leq i \leq |J| - n$, ПРОЦЕССА ДП.

ШАГ 1. Определить подграф G_i и клику K_i .

ШАГ 2. Определить множество $\pi(K_i) = \{\pi'(K_i)\}$.

ШАГ 3. Для каждого $\vartheta_i \in V$ и любых $\pi'(K_i) \in \pi(K_i)$ определить $T(\pi'(K_i), \vartheta_i)$ — множество 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 $K_i \subset J_i$ в V равно $\pi'(K_i)$:

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

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

ШАГ 4. Для каждого множества размещения $T(\pi'(K_i), \vartheta_i)$ вычислить значение функции стоимости $R(T(\pi'(K_i), \vartheta_i))$ соответствующего размещения:

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

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

ШАГ 5. Определить множество состояний

$$V_i = \{V(\pi'(K_i)) \mid \pi'(K_i) \in \pi(K_i)\}$$

процесса ДП на этапе i , где состояние $V(\pi'(K_i)) \in V_i$ определяется согласно формуле

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

ШАГ 6. Для каждого состояния $V(\pi'(K_i)) \in V_i$ вычислить значение функции Беллмана $f_i(V(\pi'(K_i)))$:

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

Перейти на этап $i + 1$ процесса ДП.

ЭТАП $|J| - n + 1$ (КОНЕЧНЫЙ) ПРОЦЕССА ДП. Найти оптимальное размещение π_G^* вершин графа G в множестве позиций V :

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

Теорема 1. Алгоритм ScChWPA находит точное решение задачи Вебера (G, V, F) , где $G = (J, E)$ — n -последовательностьсвязная цепь, V — конечное множество позиций размещения.

ДОКАЗАТЕЛЬСТВО. Разобьём исходную задачу (G, V, F) на последовательность задач (G_i, V, F) для $i = 0, 1, 2, \dots, |J| - n$, где задача (G_i, V, F) заключается в оптимальном размещении вершин подграфа G_i в множестве позиций V . Отметим, что задача $(G_{(|J|-n)}, V, F)$ соответствует исходной задаче (G, V, F) , так как $G_{(|J|-n)} = G$. Каждая задача (G_i, V, F) , в свою очередь, разбивается на ряд подзадач $(G(\pi'(K_i)), V, F)$ для любых $\pi'(K_i) \in \pi(K_i)$, где подзадача $(G(\pi'(K_i)), V, F)$ заключается в оптимальном размещении вершин подграфа G_i в множестве позиций V , когда размещение вершин его клики K_i размера n в V равно $\pi'(K_i)$. Отметим, что множество $V(\pi'(K_i)) \in V_i$ является оптимальным решением соответствующей подзадачи $(G(\pi'(K_i)), V, F)$. Естественно, для нахождения оптимального решения задачи (G_i, V, F) необходимо знать для каждого $\pi'(K_i) \in \pi(K_i)$ оптимальное решение подзадачи $(G(\pi'(K_i)), V, F)$.

Очевидно, что на начальном этапе 0 процесса ДП каждое множество $V(\pi'(K_0)) \in V_0$ является оптимальным решением соответствующей подзадачи $(G(\pi'(K_0)), V, F)$, так как при решении любой подзадачи $(G(\pi'(K_0)), V, F)$ на этапе 0 размещение всех вершин подграфа G_0 полагается заданным и равно $\pi'(K_0)$.

Ясно, что на этапе 1 процесса ДП каждая тривиальная подзадача $(G(\pi'(K_1)), V, F)$ решается оптимально, где $V(\pi'(K_1)) = \{\vartheta_1^*, \vartheta_2, \dots, \vartheta_{n+1}\}$, поскольку размещение $\{\vartheta_2, \dots, \vartheta_{n+1}\}$ вершин клики K_1 при решении подзадачи $(G(\pi'(K_1)), V, F)$ полагается равным $\pi'(K_1)$, а размещение ϑ_1^* вершины 1 при каждом заданном размещении $\{\vartheta_2, \dots, \vartheta_{n+1}\} = \pi'(K_1)$ вершин клики K_1 определено оптимально, так как множество $N(1)$ вершин, смежных с вершиной 1, равно K_1 , а оптимальное размещение ϑ_1^* вершины 1 согласно (2) находится перебором способов размещения ϑ_1 вершины 1 в множестве позиций V с целью минимизации стоимости размещения подграфа G_1 , когда размещение вершин его клики K_1 в V равно $\pi'(K_1)$.

На последующих этапах $i = 2, 3, \dots, |J| - n$ процесса ДП каждая подзадача $(G(\pi'(K_i)), V, F)$ также решается оптимально, при этом

$$V(\pi'(K_i)) = \{\vartheta_1^*, \vartheta_2^*, \dots, \vartheta_{i-1}^*, \vartheta_i^*, \vartheta_{i+1}, \dots, \vartheta_{i+n-1}, \vartheta_{i+n}\},$$

поскольку для вершин $1, 2, \dots, i - 1$ оптимальное размещение $\{\vartheta_1^*, \vartheta_2^*, \dots, \vartheta_{i-1}^*\}$ определено на этапе $i - 1$ процесса ДП для каждого заданного размещения $\{\vartheta_i, \vartheta_{i+1}, \dots, \vartheta_{i+n-1}\} = \pi'(K_{i-1})$ вершин клики K_{i-1} , а оптимальное размещение ϑ_i^* вершины i согласно (2) находится перебором способов размещения ϑ_i в множестве позиций V с целью минимизации стоимости размещения подграфа G_i , когда размещение вершин его клики K_i в V равно $\pi'(K_i) = \{\vartheta_{i+1}, \dots, \vartheta_{i+n-1}, \vartheta_{i+n}\}$ при том, что множество вершин $N(i)$, смежных с вершиной i , равно $K_i \cup \{1, 2, \dots, i - 1\}$.

Таким образом, рекуррентно решая подзадачи $(G(\pi'(K_i)), V, F)$ на этапах $i = 2, 3, \dots, |J| - n$, алгоритм на этапе $|J| - n$ процесса ДП для каждого $\pi'(K_{|J|-n}) \in \pi(K_{|J|-n})$ находит оптимальное решение соответствующей подзадачи $(G(\pi'(K_{|J|-n})), V, F)$. Поскольку задача $(G_{(|J|-n)}, V, F)$ соответствует исходной задаче (G, V, F) , согласно (3) алгоритм на конечном этапе $|J| - n + 1$ процесса ДП находит оптимальное решение π_G^* исходной задачи (G, V, F) , зная оптимальное решение $V(\pi'(K_{|J|-n}))$ каждой подзадачи $(G(\pi'(K_{|J|-n})), V, F)$. Теорема 1 доказана.

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

Доказательство. Определим оценку вычислительной сложности предложенного алгоритма. Очевидно, что на этапе 0 процесса ДП для решения одной подзадачи $(G(\pi'(K_0)), V, F)$ требуется $O(1)$ операций, поскольку оптимальное размещение $V(\pi'(K_0))$ всех вершин подграфа G_0

при решении каждой подзадачи $(G(\pi'(K_0)), V, F)$ полагается заданным и равно $\pi'(K_0)$. На этапе i , $1 \leq i \leq |J| - n$, процесса ДП для решения одной подзадачи $(G(\pi'(K_i)), V, F)$ требуется $O(|V|)$ операций, так как для определения одного множества $V(\pi'(K_i))$, являющегося оптимальным решением соответствующей задачи, необходимо найти $|V|$ значений функции $R(T(\pi'(K_i), \vartheta_i))$ для каждого $\vartheta_i \in V$, а также найти значение функции Беллмана $f_i(V(\pi'(K_i)))$, представляющее собой минимум из $R(T(\pi'(K_i), \vartheta_i))$ при $\vartheta_i \in V$.

Количество подзадач $(G(\pi'(K_i)), V, F)$, решаемых на каждом этапе $i = 0, 1, 2, \dots, |J| - n$ процесса ДП, равно $|\pi(K_i)| = O(|V|^n)$, поскольку для оптимального решения любой задачи (G_i, V, F) необходимо знать оптимальные решения подзадач $(G(\pi'(K_i)), V, F)$ для всех $\pi'(K_i) \in \pi(K_i)$, а для нахождения их оптимального решения на этапе i , $1 \leq i \leq |J| - n$, процесса ДП, необходимо знать оптимальное решение каждой подзадачи $(G(\pi'(K_{i-1})), V, F)$, найденное на предыдущем этапе $i - 1$ процесса ДП.

Исходя из этого, суммарное количество операций, осуществляемых алгоритмом на этапе 0 процесса ДП, равно $O(|V|^n) = O(1) \cdot O(|V|^n)$, а суммарное количество операций на этапе $i = 1, 2, \dots, |J| - n$ процесса ДП не превосходит $O(|V|^{n+1}) = O(|V|) \cdot O(|V|^n)$. Заметим, что на конечном этапе $|J| - n + 1$ процесса ДП суммарное количество операций равно $O(|V|^n)$, поскольку для нахождения оптимального решения π_G^* исходной задачи (G, V, F) необходимо осуществить перебор способов размещения $V(\pi'(K_{|J|-n}))$, принадлежащих множеству состояний $V_{(|J|-n)}$, с целью нахождения размещения с наименьшей стоимостью.

В соответствии с этим оценка вычислительной сложности алгоритма ScChWPA равна $O(|V|^{n+1} \cdot (|J| - n))$, так как справедливо равенство

$$O(|V|^{n+1} \cdot (|J| - n)) = O(|V|^n) + (|V|^{n+1}) \cdot (|J| - n) + O(|V|^n).$$

Определим оценку пространственной сложности алгоритма. Считаем, что для хранения каждого множества $V(\cdot)$ и $T(\cdot)$ требуется $O(|V|)$ памяти. На этапе 0 процесса ДП для определения одного множества $V(\pi'(K_0)) \in V_0$ требуется $O(|V|)$ памяти, поскольку на данном этапе $V(\pi'(K_0)) = \pi'(K_0)$ для любых $\pi'(K_0) \in \pi(K_0)$, и $O(|V|)$ памяти требуется для хранения самого множества $V(\pi'(K_0))$. На этапе i , $1 \leq i \leq |J| - n$, процесса ДП для определения одного множества $V(\pi'(K_i))$ требуется $O(|V|^2)$ памяти, так как $O(|V|)$ памяти требуется для хранения самого множества $V(\pi'(K_i))$, $O(|V|^2)$ — для хранения $|V|$ множеств $T(\pi'(K_i), \vartheta_i)$ и $O(|V|^2)$ — для хранения $|V|$ множеств $V(\pi'(K_{i-1}))$, используемых для определения соответствующих $T(\cdot)$ на этапе i .

Поскольку количество подзадач $(G(\pi'(K_i)), V, F)$, решаемых на каждом этапе $i = 0, 1, 2, \dots, |J| - n$ процесса ДП, равно $|\pi(K_i)| = O(|V|^n)$, на этапе 0 процесса ДП требуется $O(|V|^{n+1}) = O(|V|) \cdot O(|V|^n)$ памяти, а на каждом этапе i , $1 \leq i \leq |J| - n$, требуется $O(|V|^{n+2})$ памяти, так как $O(|V|^{n+1})$ памяти необходимо для хранения массива $V(\cdot)$ и $O(|V|^{n+2})$ памяти — для массива $T(\cdot)$, т. е.

$$O(|V|^{n+2}) = O(|V|) \cdot O(|V|^n) + O(|V|^2) \cdot O(|V|^n).$$

На конечном этапе $|J| - n + 1$ процесса ДП требуется $O(|V|^{n+1})$ памяти, поскольку для нахождения оптимального решения π_G^* задачи (G, V, F) необходимо знать способы оптимального размещения $V(\pi'(K_{|J|-n})) \in V_{|J|-n}$, где $|\pi(K_{|J|-n})| = O(|V|^n)$, а для хранения каждого множества $V(\cdot)$ требуется $O(|V|)$ памяти.

Исходя из того, что на каждом этапе $i = 1, 2, \dots, |J| - n$ процесса ДП для определения множеств $V(\pi'(K_i))$ требуются только соответствующие множества $V(\pi'(K_{i-1}))$, полученные на предыдущем этапе $i - 1$, необходимость хранения в памяти на этапе $i = 2, 3, \dots, |J| - n$ массивов $V(\cdot)$ и $T(\cdot)$, определённых на предыдущих этапах $i = i - 2, \dots, 0$, отпадает. В соответствии с этим оценка пространственной сложности предложенного алгоритма ScChWPA равна $O(|V|^{n+2})$. Теорема 2 доказана.

3. Теоретическое и экспериментальное исследование эффективности алгоритма ScChWPA

Предложенный алгоритм ScChWPA, находящий точное решение задачи (G, V, F) , является квазиполиномиальным, поскольку при некотором фиксированном значении входного параметра n он полиномиален.

Экспоненциальное возрастание пространственной и вычислительной сложности алгоритма ScChWPA вследствие увеличения параметра n обусловливается следующей причиной: для нахождения оптимального решения задачи (G, V, F) на любом этапе $i = 1, 2, \dots, |J| - n$ процесса ДП необходимо вычислять значения функций $R(T(\pi'(K_i), \vartheta_i))$ и $f_i(V(\pi'(K_i)))$, а также хранить в памяти соответствующие множества $T(\pi'(K_i), \vartheta_i)$ и $V(\pi'(K_i))$ для всех $\pi'(K_i) \in \pi(K_i)$, так как для решения любой задачи (G_i, V, F) необходимо знать оптимальное решение подзадачи $(G(\pi'(K_i)), V, F)$ для каждого $\pi'(K_i) \in \pi(K_i)$, а для нахождения оптимального решения таких подзадач $(G(\pi'(K_i)), V, F)$ на этапе $i = 1, 2, \dots, |J| - n$ необходимо знать оптимальное решение каждой подзадачи $(G(\pi'(K_{i-1})), V, F)$, найденное на предыдущем этапе $i - 1$.

Вследствие экспоненциального возрастания сложности предложенного алгоритма очевидно, что его применение перспективно при сравнительно малых значениях величины параметра n , причём чем меньше величина n и больше количество вершин размещаемого графа, тем он более эффективен по сравнению с точными алгоритмами, являющимися вариациями полного перебора с отсеком заведомо бесперспективных подмножеств допустимых решений, например, алгоритма ветвей и границ, так как вычислительная сложность алгоритма ScChWPA ограничивается полиномом $O(|V|^{n+1} \cdot (|J| - n))$, в то время как вычислительная сложность таких точных алгоритмов в общем случае экспоненциальна. Очевидно, что чем ближе величина параметра n к числу вершин размещаемого графа, тем предлагаемый алгоритм менее эффективен по сравнению с такими алгоритмами, являющимися вариациями полного перебора. Например, при решении задачи Вебера для n -последовательностьсвязной цепи $G = (J, E)$ с количеством вершин $|J| = n + 1$ время работы предложенного алгоритма ScChWPA сравнимо с временем работы алгоритма, использующего идею тривиального последовательного перебора допустимых решений.

Алгоритм ScChWPA был реализован на ЭВМ. 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) задачи Вебера в дискретной постановке 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. Вычисления проводились на ПК с процессором Intel Pentium 1.86 GHz.

Результаты вычислительного эксперимента приведены в табл. 1, где \bar{t}_{alg} — среднее время работы предложенного алгоритма в секундах; \bar{t}_{CLP} — среднее время работы модели ЦЛП в секундах.

Для задач Вебера для n -последовательностьсвязной цепи $G = (J, E)$ и конечного множества позиций размещения V размерности $|J| = 40$, $|V| = 40$ и выше не удалось получить решения с помощью модели ЦЛП за приемлемое время для любых значений параметра n , причём среднее время решения задач Вебера такой размерности для 1- и 2-последовательностьсвязной цепи с помощью предлагаемого алгоритма ScChWPA не превысило одной и десяти секунд соответственно. Отметим, что среднее время решения задач Вебера размерности $|J| = 100, |V| = 100$ для

1-последовательностьсвязной цепи с помощью алгоритма ScChWPA не превысило четырёх секунд.

Т а б л и ц а 1

Результаты вычислительного эксперимента

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

Применение алгоритма ScChWPA для решения задачи Вебера для n -последовательностьсвязной цепи при $n \geq 4$ нецелесообразно, так как время работы данного алгоритма превосходит время работы модели ЦЛП, реализованной в среде IBM ILOG CPLEX 12.2.

ЛИТЕРАТУРА

1. Быкова В. В. Вычислительные аспекты древовидной ширины графа // Прикл. дискрет. математика. — 2011. — № 3. — С. 65–79.
2. Забудский Г. Г., Лагздин А. Ю. Полиномиальные алгоритмы решения минимаксной квадратичной задачи о назначениях на сетях // Дискрет. анализ и исслед. операций. — 2011. Т. 18, № 4. — С. 49–65.
3. Забудский Г. Г., Филимонов Д. В. О минимакской и минисуммной задачах размещения на сетях // Тр. XII Байкальской междунар. конф. «Методы оптимизации и их приложения» (Иркутск, 24 июня–1 июля 2001 г.). — Иркутск: Изд-во ИГУ, 2001. — С. 150–155.
4. Панюков А. В. Модели и методы решения задач построения и идентификации геометрического размещения. Дисс. ... д-ра физ.-мат. наук. — Челябинск: Южноуральский ун-т, 1999. — 260 с.
5. Панюков А. В., Пельцвергер Б. Ф. Оптимальное размещение дерева в конечном множестве // Журн. вычисл. математики и мат. физики. — 1988. — Т. 28. — С. 618–620.

6. 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.
7. Сергеев С. И. Квадратичная задача о назначениях. I // Автоматика и телемеханика. — 1999. — № 8. — С. 127–147.
8. Трубин В. А. Эффективный алгоритм для задачи Вебера с прямоугольной метрикой // Кибернетика. — 1978. — № 6. — С. 67–70.
9. Филимонов Д. В. Решение дискретной минимаксной задачи размещения на древовидной сети // Мат. ежегод. науч. семинара аспирантов и студентов-выпускников. — Омск: Изд-во Омск. гос. ун-та, 2003. — С. 58–61.
10. Шангин Р. Э. Исследование эффективности приближенных алгоритмов решения одного частного случая задачи Вебера // Экономика, статистика и информатика. Вестн. УМО. — 2012. — № 1. — С. 163–169.
11. Шангин Р. Э. О некоторых свойствах n -последовательностно-связной цепи // Вестн. ЮУрГУ. Сер. Вычисл. математика и информатика. — 2013. — Т. 2, № 1. — С. 106–113.
12. Шангин Р. Э. Квазиполиномиальный алгоритм для решения задачи Вебера для n -последовательностно-связного цикла // Обзорение прикл. и пром. математики. — 2012. — Т. 19, вып. 4. — С. 122–124.
13. Шангин Р. Э. Разработка и анализ алгоритмов для задачи Вебера // Тез. докл. междунар. конф. «Проблемы оптимизации и экономические приложения» (Омск, 2–6 июля 2012 г.) — Омск: Изд-во Омск. гос. ун-та, 2012. — С. 121.
14. McKee T. A. On the chordality of a graph // J. Graph Theory. — 1993. — Vol. 17. — P. 221–232.
15. Panyukov A. V., Pelzwerger B. V. Polynomial algorithms to finite Weber problem for a tree network // J. Comput. Appl. Math. — 1991. — Vol. 35. — P. 291–296.
16. Zabudsky G. G., Filimonov D. V. An algorithm for minimax location problem on a tree with maximal distances // Proc. 2nd Int. Workshop Discrete Optimization Methods in Production and Logistics (DOM2004). — Омск; Irkutsk: ISU Press, 2004. — P. 81–85.

Шангин Роман Эдуардович,
e-mail: shanginre@gmail.com

Статья поступила
15 ноября 2012 г.

Переработанный вариант —
19 марта 2013 г.