ТРУДЫ ИНСТИТУТА МАТЕМАТИКИ И МЕХАНИКИ УрО РАН

Том 26 № 2 2020

УДК 519.8

О НЕКОТОРЫХ ЭФФЕКТИВНО РАЗРЕШИМЫХ КЛАССАХ СЕТЕВОЙ ЗАДАЧИ РАЗМЕЩЕНИЯ С ОГРАНИЧЕНИЯМИ НА ПРОПУСКНЫЕ СПОСОБНОСТИ КОММУНИКАЦИЙ[1]

Э. Х. Гимади, О. Ю. Цидулко

В работе исследуется задача размещения в сетях с ограниченными пропускными способностями коммуникаций, в которой требуется разместить предприятия в вершинах заданной сети так, чтобы с минимальными затратами единовременно удовлетворить спросы всех клиентов, находящихся в вершинах сети. Рассматриваются постановки задачи с делимыми спросами, где спрос клиента может быть удовлетворен несколькими предприятиями совместно, и неделимыми спросами, тогда спрос клиента должен быть целиком удовлетворен одним предприятием. В работе показано, что задача с неделимыми спросами NP-трудна, даже если заданная сеть является простым путем, и NP-трудна в сильном смысле, если сеть — дерево. Задача с делимыми спросами слабо NP-трудна на деревьях. Для этой задачи в работе предложен псевдополиномиальный алгоритм решения на графах с древесной шириной, ограниченной константой, а также представлен линейный алгоритм для случая, когда граф сети является простым путем.

Ключевые слова: задача размещения, пропускные способности, делимый спрос, неделимый спрос, NP-трудная задача, древесная ширина, псевдополиномиальный алгоритм, полиномиальный алгоритм.

E. Kh. Gimadi, O. Yu. Tsidulko. On some efficiently solvable classes of the network facility location problem with constraints on the capacities of communication lines.

We study the network facility location problem with constraints on the capacities of communication lines, called Restricted Facility Location Problem (RFLP). It is required to locate facilities at the vertices of a given network graph so as to simultaneously satisfy the demands of the clients, which are also located at the vertices of the network graph, at minimum cost. We consider two statements of the problem: the multiple-allocation RFLP, where the demand of a client can be satisfied jointly by several facilities, and the single-allocation RFLP, where the demand of a client must be entirely satisfied by a single facility. We show that the single-allocation RFLP is NP-hard even if the network is a simple path and strongly NP-hard if the network is a tree. The multiple-allocation RFLP is weakly NP-hard on trees. For this problem we propose a pseudopolynomial algorithm for the case where the network graph has constant treewidth, and show a linear-time algorithm for the case where the network is a simple path.

Keywords: facility location problem, capacities, single-allocation, multiple-allocation, NP-hard problem, treewidth, pseudopolynomial algorithm, polynomial-time algorithm.

MSC: 90-02, 90B80

DOI: 10.21538/0134-4889-2020-26-2-108-124

Введение

Задачи размещения (Facility Location Problems) — это широкий раздел в комбинаторной оптимизации, в котором изучаются задачи о выборе наилучшего расположения одного или нескольких предприятий для эффективного обслуживания спроса заданного множества клиентов. Задачи размещения имеют большое число приложений в логистике, в сфере размещения медицинских объектов, в системах GIS и в сфере телекоммуникаций [20, разд. III], в задачах кластеризации, в компьютерном зрении [21], в биоинформатике [19] и др.

В данной работе рассматриваются сетевые задачи размещения, в которых клиенты и возможные пункты открытия предприятий находятся в вершинах заданного графа сети. Сетевые

[1]Работа выполнена при поддержке РФФИ (проект 18-31-00470) и частичной поддержке Математического Центра в Академгородке (соглашение с Министерством науки и высшего образования Российской Федерации 075-15-2019-1675).

задачи размещения возникают, например, в задачах размещения прокси-серверов в компьютерных сетях, в задачах управления поставками, при планировании путей эвакуации в транспортных сетях [3]. В таких задачах естественным образом возникают ограничения на объем продукции, который может единовременно передаваться по ребрам сети.

Сетевую задачу размещения с ограничениями на пропускные способности коммуникаций в этой работе мы будем сокращенно называть RFLP (от английского названия Restricted Facility Location Problem, предложенного в статье [14]). В задаче RFLP в заданном n-вершинном графе для каждой вершины известен объем клиентского спроса и стоимость открытия предприятия в этой вершине, а для каждого ребра сети известны его пропускная способность и стоимость передачи по нему единицы продукта. Требуется открыть набор предприятий так, чтобы удовлетворить спрос всех клиентов, не нарушая ограничений на пропускные способности ребер, с минимальными суммарными расходами на открытие предприятий и единовременную транспортировку продукта от предприятий к клиентам. Задачу можно рассматривать в постановке *с делимыми спросами*, когда спрос одного клиента может быть обслужен несколькими предприятиями совместно, и в постановке *с неделимыми спросами*, когда спрос клиента должен быть целиком удовлетворен одним предприятием.

В произвольном графе даже сетевая задача размещения с неограниченными пропускными способностями ребер NP-трудна [11], однако, она решается точно за время $O(n)$, если входной граф является простым путем [17], за время $O(n \log n)$, если входной граф является деревом [23], за время $O(n^{k+2})$ [2; 15], если входной граф является частичным k-деревом, или, что эквивалентно [8, теорема 1], имеет древесную ширину, не превосходящую k.

Непосредственно задача RFLP исследовалась в литературе в работах [14;25;26]. Для задачи RFLP с *делимыми спросами* в статье [14] построен алгоритм с трудоемкостью $O(n^3)$ в случае, когда входной граф является простым путем, а в работах [25] и [26] предложены псевдополиномиальные алгоритмы решения этой задачи на деревьях и частичных 2-деревьях. Отметим, что в работе [14] нами также предлагался алгоритм с трудоемкостью $O(n^3)$ для задачи RFLP с *неделимыми спросами* на путевом графе, однако он основывался на неверном предположении о структуре оптимальных решений задачи. Далее мы покажем (см. утверждение 2), что задача RFLP с *неделимыми спросами* на путевом графе NP-трудна.

Среди задач с ограничениями на пропускные способности сети в литературе известны так называемые задачи размещения источников (Source Location) [3;4;24], близкие к задачам RFLP с *делимыми спросами*. Эти задачи мотивированы приложениями передачи данных, где время передачи одного пакета данных должно быть пренебрежимо мало и не заметно для конечного пользователя, поэтому затраты на передачу данных по ребрам сети в этих постановках равны нулю. Кроме нулевых затрат на передачу данных, задачи Source Location отличаются от RFLP с делимыми спросами тем, что в них, как правило, клиенты должны быть обслужены не одновременно, т. е. после того, как спрос некоторого клиента удовлетворен совокупностью открытых предприятий, ребра сети освобождаются, и для следующего клиента можно вновь использовать их полную пропускную способность. Задачи Source Location, в которых спрос всех клиентов должен быть обслужен одновременно, называются задачами Simultaneous Source Location (SSL) [3; 24]. Эти задачи являются частным случаем RFLP с *делимыми спросами*. В работе [3] рассматривалась задача SSL, где стоимость открытия любого предприятия равна 1. Для случая, когда входной граф является деревом, в [3] был предложен алгоритм с трудоемкостью $O(n^3)$, который позднее в работе [24] был улучшен до $O(n)$, и показано, что задача SSL становится NP-трудной на графах с древесной шириной, равной 2. При фиксированном k для задачи SSL на графах с древесной шириной k в [3] построен псевдополиномиальный алгоритм.

К задачам RFLP с *неделимыми спросами* среди известных в литературе наиболее близки задачи о неделимых потоках (Unsplittable Flow Problem, UFP). В этих задачах даны граф, пропускные способности ребер, и m запросов: четверок вида $\{s_i, t_i, d_i, w_i\}$, где s_i — вершина источник, t_i — вершина сток, d_i — объем продукции, который нужно доставить из s_i

в t_i, а w_i — доход от выполнения запроса. Требуется выбрать подмножество запросов с максимальным суммарным доходом, которые можно удовлетворить одновременно. Частным случаем задачи UFP является задача о максимальном числе непересекающихся путей. Задача UFP NP-трудна даже на графе с двумя вершинами и двумя параллельными ребрами [18], и в общем случае APX-трудна [16]. Поэтому большая часть результатов для нее касается приближенных алгоритмов, в том числе для частных случаев с ограниченной структурой графа [1; 10], с ограничениями на функцию доходов [1], с одним источником [13; 18] и др.

В данной работе мы исследуем частные случаи задачи RFLP со структурными ограничениями на граф сети. В разд. 2 мы уточним сложностной статус задач RFLP с произвольными стоимостями открытия предприятий на деревьях и простых путях, поскольку этот вопрос в предыдущих работах по RFLP [14;25;26] не обсуждался. Мы покажем, что задача RFLP с *неделимыми спросами* NP-трудна, даже если входной граф — простой путь, и NP-трудна в сильном смысле на деревьях, а NP-трудность задачи RFLP с *делимыми спросами* на деревьях напрямую следует из аналогичного результата для задачи Source Location [4]. В разд. 3 мы обобщим псевдополиномиальный алгоритм из [3] на случай задачи RFLP с делимыми спросами, произвольными стоимостями открытия предприятий и ненулевыми транспортными расходами на графах с ограниченной древесной шириной. За счет более аккуратных вычислений в самом алгоритме и при оценке числа его шагов полученный алгоритм по сравнению с алгоритмом из [3] имеет трудоемкость, меньшую на два порядка. Наконец, в разд. 4 мы рассмотрим задачу RFLP с делимыми спросами на путевом графе, для которой был известен алгоритм с трудоемкостью $O(n^3)$ [14]. Мы покажем, как находить точные решения этой задачи за время $O(n)$, используя подход к решению задачи с неограниченными пропускными способностями из [17].

Ниже в таблице собраны результаты, касающиеся вычислительной сложности задач RFLP на различных простых типах графов:

Граф G	RFLP с делимыми спросами: $f_i \equiv 1,\ c(e) \equiv 0$		RFLP с делимыми спросами: Общий случай		RFLP с неделимыми спросами: $f_i \equiv 1,\ c(e) \equiv 0$	
простой путь	$O(n)$	[24]	$O(n)$	(теорема 2)	NP-тр.	(утверждение 2)
дерево	$O(n)$	[24]	NP-тр.	(утверждение 1)	NP-тр. с.с.	(утверждение 2)
tw = 2	NP-тр.,	[3]	NP-тр.		NP-тр. с.с.	
tw = $O(1)$	$O(nB^{2\mathrm{tw}})$		$O(nB^{2\mathrm{tw}})$	(теорема 1)	NP-тр. с.с.	

Для задачи RFLP на графе каждого типа либо указана трудоемкость известного алгоритма решения задачи, либо сложностной статус задачи, где "NP-тр." — "NP-трудна"; "NP-тр.с.с." — сокращение от "NP-трудна в сильном смысле"; tw — древесная ширина графа G; f_i — стоимость открытия предприятия в вершине i; $c(e)$ — стоимость передачи единицы продукта по ребру e; B — максимальный объем продукции, передаваемый по ребру.

1. Постановка задачи RFLP

Приведем формальную постановку задачи RFLP. Дан граф сети $G = (V, E)$ со множеством вершин $V = \{1, \dots, n\}$. Для каждой вершины $i \in V$ заданы величина клиентского спроса $b_i \in \mathbb{Z}_{\geq 0}$ и стоимость $f_i \in \mathbb{Z}_{\geq 0} \cup \{\infty\}$ открытия предприятия в вершине i. Для каждого ребра $e \in E$ заданы его пропускная способность $q(e) \in \mathbb{Z}_{\geq 0} \cup \{\infty\}$ и стоимость $c(e) \in \mathbb{Z}_{\geq 0}$ транспортировки единицы продукции по ребру e.

В задаче RFLP требуется найти подмножество вершин $S \subseteq V$, в которых будут открыты предприятия, и план перевозок $x = (x_{ij}^p)$ такие, что

$$\sum_{i \in S} f_i + \sum_{i,j \in V} \sum_{p \in P_{ij}} g_{ij}^p x_{ij}^p \to \min_{S, x_{ij}^p} \tag{1.1}$$

при условиях

$$\sum_{i\in S}\sum_{p\in P_{ij}} x_{ij}^p = b_j, \quad j \in V, \tag{1.2}$$

$$\sum_{(i,j,p)\in Q_e} x_{ij}^p \le q(e), \quad e \in E, \tag{1.3}$$

$$x_{ij}^p \ge 0, \quad i \in S, \quad j \in V, \quad p \in P_{ij}, \tag{1.4}$$

где x_{ij}^p — объем продукта, поставленного предприятием $i \in S$ в вершину $j \in V$ по пути p; $g_{ij}^p = \sum_{e\in p} c(e)$ — стоимость транспортировки единицы продукта по пути p; P_{ij} — множество всех простых путей в графе G, ведущих из вершины i в вершину j; Q_e — множество троек (i, j, p) таких, что путь $p \in P_{ij}$ проходит через ребро e.

Условие (1.2) обеспечивает удовлетворение спроса в вершине $j \in V$, неравенство (1.3) задает ограничения на пропускные способности ребер.

Задачу (1.1)–(1.3) и (1.4) будем называть задачей RFLP *с делимыми спросами*. В ней несколько предприятий могут совместно обеспечивать спрос клиента в вершине j. Задачи *с неделимыми спросами*, где весь спрос клиента должен обеспечиваться ровно одним предприятием, могут моделироваться либо заменой условия (1.4) на условие $x_{ij}^p \in \{0, b_j\}$, $i \in S$, $j \in V$, $p \in P_{ij}$, тогда весь необходимый объем продукции будет доставлен из некоторого предприятия $i \in S$ клиенту j по одному пути, либо добавлением условия $\sum_{p\in P_{ij}} x_{ij}^p \in \{0, b_j\}$, $j \in V$, $i \in S$, тогда единственное предприятие i, обслуживающее клиента j, может доставлять к нему продукцию по нескольким путям. Заметим, что если граф G является деревом или в задаче отсутствуют ограничения (1.3), в оптимальном решении весь неделимый объем продукции будет доставляться от предприятия i к клиенту j по кратчайшему пути.

2. Сложностной статус задачи RFLP

В этом разделе мы обсудим сложностной статус задачи RFLP на деревьях и путевых графах. Доказательство NP-трудности частных случаев задач RFLP легко следует из известных результатов для близких к RFLP задачи размещения источников в сети (Source Location) [4] и задачи о неделимом потоке (Unsplittable Flow Problem) [18].

Утверждение 1. *Задача* RFLP с делимыми спросами *NP-трудна, даже если транспортные затраты равны нулю, а заданный граф транспортной сети является графом-звездой, т. е. деревом, в котором все, кроме одной, вершины — листья.*

Доказательство следует из NP-трудности задачи размещения источников (Source Location) на графе-звезде с одним клиентом, к которой в работе [4, теорема 5.1] сводится известная NP-трудная задача о ранце. Как было сказано во введении, задача Source Location отличается от задачи RFLP тем, что в Source Location разные клиенты обслуживаются не одновременно. Однако, задача Source Location с одним клиентом является частным случаем задачи RFLP, откуда следует требуемое утверждение.

Утверждение 2. *Задача* RFLP с неделимыми спросами *NP-трудна, даже когда заданный граф является простым путем, и NP-трудна в сильном смысле, даже если заданный граф является гусеничным деревом, т. е. деревом, которое превратится в простой путь, если из него одновременно удалить все листья. Трудность задачи в обоих случаях сохраняется, даже если транспортные затраты равны нулю, и стоимости открытия всех предприятий одинаковые.*

Доказательство следует из NP-трудности задачи Unsplittable Flow Problem (UFP) [18, утверждение 3.0.1]. В [18] показано, что задача UFP NP-трудна даже на графах с двумя

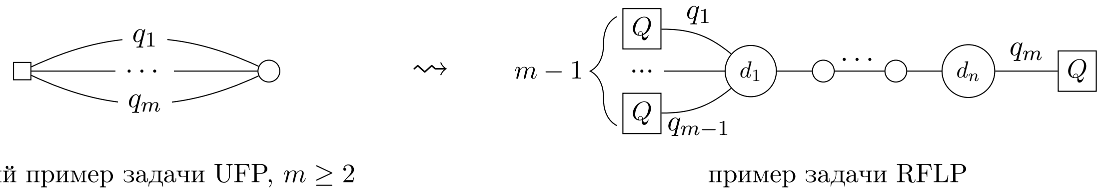

Рис. 1. Сведение сложного примера задачи UFL к задаче RFLP с неделимыми спросами. Квадратами обозначены вершина-источник и с необходимостью открытые предприятия, в круглых вершинах находятся клиенты.

вершинами и двумя параллельными ребрами, поскольку к ней сводится задача о ранце, и NP-трудна в сильном смысле на графах с двумя вершинами и нефиксированным числом параллельных ребер, поскольку к ней сводится задача об упаковке в контейнеры. Покажем, что этот сложный частный случай задачи UFP сводится к RFLP с неделимыми спросами (см. рис. 1).

В рассматриваемом частном случае задачи UFP дан граф G с вершинами s и t, и m параллельными ребрами, каждое из которых имеет пропускную способность q_j, $j = 1, \ldots, m$. Заданы n запросов вида $\{s, t, d_i\}$, $i = 1, \ldots, n$, на доставку неделимого спроса d_i из s в t, т. е. объем продукции d_i должен быть доставлен целиком по одному ребру. Необходимо выяснить, возможно ли выполнить все n запросов одновременно.

В эквивалентном примере задачи RFLP рассмотрим путевой граф G' с вершинами v_i, в которых задан спрос d_i, $i = 1, \ldots, n$, и назначим ребрам пути G' пропускную способность, равную суммарному спросу. Пусть $Q = \max\{q_j \mid 1 \leq j \leq m\} + 1$. Добавим в G' новые вершины $u_1, \ldots, u_m$, спрос в которых равен Q. Добавим в G' ребро $\{v_n, u_m\}$ с пропускной способностью q_m и ребра $\{v_1, u_j\}$ с пропускными способностями q_j, $j = 1, \ldots, m-1$. Заметим, что при $m = 2$ граф G' является простым путем, а при $m > 2$ — гусеничным деревом. Положим стоимость открытия предприятия в любой вершине равной 1, а стоимость транспортировки по каждому ребру в G' равной 0. В любом допустимом решении построенного примера RFLP в каждой вершине u_j, $j = 1, \ldots, m$, будет открыто предприятие, поскольку каждая вершина u_j связана с остальным графом единственным ребром с пропускной способностью q_j, которой недостаточно, чтобы передать в u_j объем продукции Q из других вершин. То есть стоимость оптимального решения в таком примере всегда не меньше m. Если стоимость оптимального решения примера RFLP равна m, то в примере задачи UFP есть допустимое решение, в котором по каждому ребру $j = 1, \ldots, m$ передается такой же объем продукции, как и по ребру, ведущему из u_j в решении задачи RFLP, а если стоимость оптимального решения RFLP больше m, то допустимого решения примера задачи UFP нет. И наоборот.

3. Задача RFLP на графах с ограниченной древесной шириной

В этом разделе мы рассмотрим задачу RFLP с делимыми спросами в случае, когда древесная ширина (treewidth, tw) заданного графа сети ограничена константой.

Понятие древесной ширины и его определение с помощью древесных декомпозиций было предложено Робертсоном и Сеймуром в работе [22]. Древесная декомпозиция дает представление графа в виде “укрупненного” дерева, узлы которого содержат подмножества вершин исходного графа, а ширина декомпозиции — это числовая характеристика, отражающая степень такого “укрупнения”. Древесная ширина графа — это минимальная ширина среди всех возможных его древесных декомпозиций. В частности, древесная ширина дерева равна 1, древесная ширина n-вершинной клики равна $n-1$, а частичные k-деревья эквивалентны графам с древесной шириной не более k [8, теорема 1]. Графы с ограниченной древесной шириной часто возникают в приложениях [8; 9], при этом в таких графах для многих NP-трудных задач существуют эффективные алгоритмы динамического программирования (см., например, обзор [7] и книгу [12, гл. 7]).

Для задачи RFLP с делимыми спросами на графах с древесной шириной 2 в работе [26] был построен алгоритм с трудоемкостью $O(n\,B^4)$, где n — число вершин графа, а величина $B = \min\bigl\{\sum_{k=1}^{n} b_k, \max_{e\in E(G)} q(e)\bigr\}$. В работе [3] исследовался частный случай задачи RFLP с делимыми спросами, нулевыми транспортными затратами и единичными стоимостями открытия предприятий и был предложен алгоритм динамического программирования с трудоемкостью $O(N\,n^2\,\mathrm{tw}\,B^{2\mathrm{tw}+2})$, где N — число узлов в древесной декомпозиции графа.

В этой главе мы обобщим алгоритм динамического программирования из [3] на случай задачи RFLP с делимыми спросами и произвольными стоимостями транспортировки и открытия предприятий. За счет более аккуратных вычислений в самом алгоритме и при оценке числа его операций трудоемкость алгоритма составит $O(n\,\mathrm{tw}^2\,B^{2\mathrm{tw}})$. Если $\mathrm{tw} = O(1)$, алгоритм псевдополиномиален. При $\mathrm{tw} = 2$ его трудоемкость совпадает с трудоемкостью алгоритма из [26].

3.1. Древесная ширина и древесные декомпозиции

Введем формальные определения древесной ширины и древесной декомпозиции графа, которые потребуются нам далее.

О п р е д е л е н и е 1. *Древесная декомпозиция* $\mathbb{T} = (T, \{X_t\}_{t\in V(T)})$ графа $G = (V, E)$ состоит из дерева T, в котором каждому узлу $t \in V(T)$ соответствует подмножество вершин исходного графа $X_t \subseteq V$ такое, что

(1) для каждой вершины $v \in V$ найдется узел t дерева T такой, что $v \in X_t$,

(2) для каждого ребра $\{u, v\} \in E$ найдется узел t дерева T такой, что $\{u, v\} \subseteq X_t$,

(3) для каждой вершины $v \in V$ узлы $t \in V(T)$, для которых $v \in X_t$, образуют связное поддерево в T.

Шириной декомпозиции $\mathbb{T}$ называется величина $w(\mathbb{T}) := \max_{t\in V(T)} |X_t| - 1$; *древесной шириной* $tw(G)$ графа G называется минимальная ширина всех его возможных древесных декомпозиций.

Нахождение древесной ширины и соответствующей древесной декомпозиции графа G является NP-трудной задачей, однако, в работе [6] был предложен алгоритм, находящий древесную декомпозицию ширины $\mathrm{tw}(G)$ с $O(n)$ узлами за время $\mathrm{tw}(G)^{O(\mathrm{tw}(G)^3)}\,n$. Если $\mathrm{tw}(G)$ ограничено константой, нахождение древесной декомпозиции займет линейное время.

При решении комбинаторных задач на графах с ограниченной древесной шириной, алгоритмы динамического программирования строятся по корневому дереву декомпозиции. При этом для упрощения рекуррентных соотношений зачастую удобнее использовать *хорошие древесные декомпозиции*, в которых множества вершин, соответствующие соседним узлам дерева декомпозиции, отличаются друг от друга не более, чем на одну вершину.

О п р е д е л е н и е 2. *Хорошая древесная декомпозиция (nice tree decomposition)* $\mathbb{T}$ — это древесная декомпозиция, в которой есть один выделенный корневой узел r, причем $X_r = \varnothing$, а все остальные ее узлы принадлежат к одному из следующих типов:

- *Лист*: узел t является листом в $\mathbb{T}$ и $X_t = \varnothing$.
- *Узел включения*: узел t в $\mathbb{T}$ с одним сыном t_1 и $X_t = X_{t_1} \cup \{v\}$ для некоторой вершины $v \notin X_{t_1}$.
- *Узел исключения*: узел t в $\mathbb{T}$ с одним сыном t_1 и $X_t = X_{t_1} \setminus \{v\}$ для некоторой вершины $v \in X_{t_1}$.
- *Узел слияния*: узел t в $\mathbb{T}$ с двумя сыновьями t_1 и t_2, где $X_t = X_{t_1} = X_{t_2}$.

Заметим, что в силу условия (3) в определении 1 для каждой вершины v в хорошей древесной декомпозиции есть ровно один узел исключения. Узлов включения для вершины v может быть больше одного, но они не могут быть потомками друг друга. Из произвольной древесной декомпозиции ширины w за время $O(w^2\,n)$ можно получить хорошую древесную декомпозицию ширины не более w с $O(w\,n)$ узлами [12, лемма 7.4].

Согласно определению 1 для каждого узла t в любой древесной декомпозиции ширины w мощность $|X_t| \le w + 1$. В дальнейшем для более аккуратного подсчета числа действий в алгоритме динамического программирования нам потребуется следующее замечание о размере множеств X_t в хорошей древесной декомпозиции.

З а м е ч а н и е 1. Пусть $\mathbb{T}$ — хорошая древесная декомпозиция с шириной w, тогда

(а) если t — *узел исключения*, то $|X_t| \leq w$;

(б) если t — *узел слияния*, то $|X_t| = w + 1$ тогда и только тогда, когда в T для узла t в его левом и правом поддеревьях найдутся *узлы включения* y_1 и y_2 соответственно, такие, что $|X_{y_1}| = |X_{y_2}| = w + 1$ и на путях между t и y_1, t и y_2 нет *узлов исключения.*

Д о к а з а т е л ь с т в о. Покажем (а). Если t — узел исключения, то по определению $X_t = X_{t'} \setminus \{v\}$, где $v \notin X_{t'}$, причем $|X_{t'}| \leq w + 1$, а значит, $|X_t| \leq w$.

Покажем (б) для левого (правого) поддерева узла t. Поскольку любой узел слияния t сохраняет множество X_t таким же, как было у его сыновей, то рекурсивно получим $|X_t| = |X_y|$, где y — ближайший к t узел в левом (правом) поддереве t, который не является узлом слияния.

($\Rightarrow$) Если y — это узел включения такой, что $|X_y| = w + 1$, то $|X_t| = w + 1$.

($\Leftarrow$) Пусть $|X_t| = w+1$. Тогда $|X_y| = |X_t| = w+1$. Если y — это узел исключения, получим противоречие с пунктом (а). Узел y также не может быть листом или корнем, поскольку тогда $|X_y| = 0$. Следовательно, y — узел включения. Замечание 1 доказано.

З а м е ч а н и е 2. Обходом в глубину по дереву T можно для каждого *узла слияния* t, в котором $|X_t| = w + 1$, найти ближайшие *узлы включения* y_1 и y_2 в его левом и правом поддеревьях. Для каждого узла слияния будем дополнительно хранить номера вершин j_1 и j_2, которые добавляются в узлах y_1 и y_2 соответственно.

3.2. Псевдополиномиальный алгоритм

В этом разделе мы приведем алгоритм динамического программирования для решения задачи RFLP с делимыми спросами на графах с древесной шириной, ограниченной константой. Будем считать, что исходный n-вершинный граф G задан вместе с его хорошей древесной декомпозицией ширины $\mathrm{tw}(G)$. В противном случае, как упоминалось в предыдущем разделе, при $\mathrm{tw}(G) = O(1)$, древесную ширину $\mathrm{tw}(G)$ и хорошую древесную декомпозицию с $O(\mathrm{tw}(G)n)$ узлами можно найти за время, линейное по n.

Алгоритм динамического программирования последовательно строит частичные решения в узлах хорошей древесной декомпозиции $\mathbb{T}$ из частичных решений в узлах-потомках. При этом нам будет достаточно рассматривать такие частичные решения, в которых суммарные потоки продукции, проходящие по ребрам или поступающие в вершины исходного графа, обладают следующими свойствами.

Утверждение 3 [26, лемма 1]. *Существует такое оптимальное решение задачи* RFLP *с делимыми спросами, что*

1) *каждый пункт производства не является транзитной вершиной для потоков продукта из других предприятий*;

2) *все потоки продукта, идущие через данное ребро, имеют одинаковое направление.*

Частичные решения. Каждому узлу t декомпозиции $\mathbb{T}$ помимо подмножества вершин X_t исходного графа $G = (V, E)$ будем ставить в соответствие подграф $G_t = (V_t, E_t)$ графа G такой, что

∘ $V_t := \{v \in X_y \mid$ для всех узлов y, являющихся потомками t в $\mathbb{T}\} \setminus X_t$,

∘ $E_t = \{\{u, v\} \in E \mid u, v \in V_t\}$.

Для каждого узла $t \in T$ будем вычислять частичное решение, величину $F_t[z_1, \ldots, z_n]$, равную оптимальному решению задачи RFLP на подграфе G_t, если в каждую вершину $i \in X_t$ поступает дополнительный целочисленный поток продукции z_i. Если $z_i \geq 0$, считаем, что поток поступает снаружи подграфа G_t; если $z_i \leq 0$, считаем, что поток поступает из подграфа G_t. Для вершин $i \notin X_t$ дополнительный поток $z_i = 0$. Если набор $z = (z_1, \ldots, z_n)$ по каким-то причинам недопустим для узла t, то $F_t[z_1, \ldots, z_n] := \infty$.

Рекуррентные соотношения. Покажем, как вычислить величины $F_t[z_1,\dots,z_n]$ для узла t каждого типа: листа, узла включения, узла слияния и узла исключения.

Лист. По определению, если $t \in T$ — лист, то $X_t = \varnothing$ и $F_t[z_1,\dots,z_n] := 0$, где допустим только набор $z = (0,\dots,0)$.

Узел включения. Пусть t_1 — сын узла t в дереве T такой, что $X_t = X_{t_1} \cup \{i\}$, где вершина $i \notin X_{t_1}$. Поскольку $i \notin X_{t_1}$, то согласно пп. (2) и (3) определения 1 $i \notin X_y$ для всех потомков y узла t_1, и, следовательно, вершина i не лежит в подграфе G_t и не имеет ребер, ведущих в вершины из V_t. Тогда дополнительный поток продукта z_i в вершину i не повлияет на решение задачи в подграфе G_t:

$$F_t[z_1,\dots,z_{i-1},z_i,z_{i+1},\dots,z_n] = F_{t_1}[z_1,\dots,z_{i-1},0,z_{i+1},\dots,z_n], \tag{3.1}$$

где допустимыми являются наборы $z \in \mathbb{Z}^n$, в которых $z_j = 0$ для всех $j \notin X_t$, $z_i \in [0,B] \cap \mathbb{Z}$ и $z_j \in [-B,B] \cap \mathbb{Z}$ для $j \in X_{t_1}$.

Узел слияния. Пусть узел t в $\mathbb{T}$ имеет двух сыновей t_1 и t_2, где $X_t = X_{t_1} = X_{t_2}$. В оптимальном решении для подграфа G_t потоки продукции, входящие в вершины X_t, должны оптимально распределяться между подграфами G_{t_1} и G_{t_2}:

$$F_t[z_1,\dots,z_n] = \min_{\alpha}\{F_{t_1}[z_1-\alpha_1,\dots,z_n-\alpha_n] + F_{t_2}[\alpha_1,\dots,\alpha_n]\}, \tag{3.2}$$

где допустимыми являются наборы $z \in \mathbb{Z}^n$, в которых $z_j = 0$ для всех $j \notin X_t$, $z_j \in [-B,B] \cap \mathbb{Z}$ для $j \in X_t$, а минимум берется по всем наборам $\alpha = (\alpha_1,\dots,\alpha_n) \in \mathbb{Z}^n$ таким, что $|\alpha_j| \le B$ и $|z_j - \alpha_j| \le B$ для всех $i = 1,\dots,n$, и $\alpha_j = 0$ для всех вершин $j \notin X_t$. В случае $|X_t| = \mathrm{tw}+1$ будем дополнительно полагать $\alpha_{j_1} = z_k$, $\alpha_{j_2} = 0$, где j_1 и j_2 номера вершин из замечания 2, хранящиеся для узла слияния t.

Поясним дополнительные условия на наборы α при $|X_t| = \mathrm{tw}+1$. В этом случае согласно замечанию 1 ближайшие к t узлы в левом и правом поддеревьях T с корнем t, не являющиеся узлами слияния, есть узлы включения y_1 и y_2, причем $|X_{y_1}| = |X_{y_2}| = \mathrm{tw}+1$. Согласно замечанию 2 нам известны вершины j_1 и j_2, добавленные в этих узлах. Тогда в вершину j_1 в подграфе G_{t_1} достаточно отправить поток величины 0, поскольку,

∘ если y_1 — сын узла t, то для любого потока z_{j_1} в вершину j_1 выполняется (3.1);

∘ если в кратчайшем пути от узла t к узлу y_1 в дереве T есть узел слияния $s \neq t$, за счет перераспределения потоков в (3.2) все допустимые величины потоков для поддерева, не содержащего y_1, тем не менее будут просмотрены.

Аналогичные рассуждения верны для вершины j_2 в подграфе G_{t_2}.

Узел исключения. Пусть узел t в $\mathbb{T}$ имеет одного сына t_1 и $X_t = X_{t_1} \setminus \{i\}$ для вершины $i \in X_{t_1}$. Поскольку $i \notin X_t$, согласно определению 1 все ребра, инцидентные вершине i в графе G, находятся в подграфе G_t или ведут в вершины из X_t. При исключении вершины i принимается решение о том, открывать в ней предприятие или нет, и перераспределяются потоки, идущие через i. Обозначим $V_i := \{j \in X_{t_1} \mid \{i,j\} \in E\}$. Тогда

$$F_t[z_1,\dots,z_{i-1},0,z_{i+1},\dots,z_n] = \min\{Y(z),N(z)\},$$

где

$$Y(z) = \min_{\alpha}\Big\{f_i + \sum_{j\in V_i} c(i,j)|\alpha_j| + F_{t_1}[z_1+\alpha_1,\dots,\underset{i}{0},\dots,z_n+\alpha_n]\Big\},$$

$$N(z) = \min_{\alpha}\Big\{\sum_{j\in V_i} c(i,j)|\alpha_j| + F_{t_1}[z_1-\alpha_1,\dots,\alpha_i,\dots,z_n-\alpha_n]\Big\}. \tag{3.3}$$

где допустимыми являются наборы $z \in \mathbb{Z}^n$, в которых $z_j = 0$ для всех $j \notin X_t$, $z_j \in [-B,B] \cap \mathbb{Z}$ для $j \in X_t$. Минимум в $Y(z)$ берется по всем наборам $\alpha = (\alpha_1,\dots,\alpha_n) \in \mathbb{Z}^n$ таким, что $\alpha_j = 0$ для всех $j \notin V_i$, $0 \le \alpha_j \le \min\{q(i,j),B\}$ для всех $j \in V_i$. Минимум в $N(z)$ берется по всем наборам $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}^n$ таким, что $\alpha_j = 0$ для всех $j \notin V_i \cup \{i\}$, $|\alpha_j| \le \min\{q(i,j), B\}$ для всех $j \in V_i$, и $\alpha_i \in [-B, B]$ такое, что $\sum_{j \in V_i} \alpha_j - b_i \ge \alpha_i$. Если для фиксированного набора z наборов α с требуемыми свойствами нет, то $F_t[z_1, \ldots, z_n] := \infty$.

Здесь $Y(z)$ — это стоимость частичного решения, в котором в вершине i открывается предприятие, и в каждую вершину $j \in X_t$, смежную с i, по ребру $\{i,j\}$ добавляется допустимый поток величины $\alpha_j \ge 0$. Соответственно, $N(z)$ это стоимость частичного решения, в котором в вершине i не открывается предприятие, а из каждой вершины $j \in X_t$, смежной с i, по ребру $\{i,j\}$ в i направляется допустимый поток величины α_j, из поддерева G_t в i направляется дополнительный поток α_i и сумма всех потоков в i позволяет удовлетворить спрос b_i.

Теорема 1. *Для задачи* RFLP *на графах с фиксированной древесной шириной* tw *существует псевдополиномиальный алгоритм решения с трудоемкостью* $O(n\,B^{2\mathrm{tw}})$, *где величина* $B = \min\{\sum_{k=1}^n b_k, \max_{e \in E(G)} q(e)\}$.

Д о к а з а т е л ь с т в о. При заданной хорошей древесной декомпозиции графа ширины tw алгоритм работает следующим образом.

Для каждого узла $t \in T$ вычислим значения $F_t[z_1, \ldots, z_n]$, соответствующие оптимальным частичным решениям в поддереве G_t, для каждого допустимого набора $z = (z_1, \ldots, z_n) \in \mathbb{Z}^n$. Оптимальное значение целевой функции задачи RFLP в исходном графе G будет лежать в $F_r[0, \ldots, 0]$.

Оценим время работы алгоритма. На предварительном шаге за время $O(\mathrm{tw}\, n)$ для всех узлов слияния найдем величины из замечания 2. Оценим время, требующееся для вычисления величин $F_t[z_1, \ldots, z_n]$ для всех допустимых наборов $z = (z_1, \ldots, z_n) \in \mathbb{Z}^n$ в узле каждого типа.

Если t — узел включения, то $|X_t| \le \mathrm{tw} + 1$, и для вычисления всех величин $F_t[z_1, \ldots, z_n]$ согласно (3.1) потребуется время $O(B^{\mathrm{tw}+1})$.

Если t — узел слияния и $|X_t| \le \mathrm{tw}$, то в (3.2) необходимо просмотреть $O(B^{\mathrm{tw}})$ наборов α для каждого из $O(B^{\mathrm{tw}})$ наборов z; если $|X_t| = \mathrm{tw} + 1$, то потребуется просмотреть $O(B^{\mathrm{tw}-1})$ наборов α для каждого из $O(B^{\mathrm{tw}+1})$ наборов z. На проверку допустимости каждой пары наборов α и z потребуется время $O(\mathrm{tw})$. Суммарно обработка узла слияния занимает время $O(\mathrm{tw}\, B^{2\mathrm{tw}})$.

Если t — узел исключения, то $|X_t| \le \mathrm{tw}$ в силу замечания 1, в (3.3) необходимо просмотреть $O(B^{\mathrm{tw}})$ наборов α для каждого из $O(B^{\mathrm{tw}})$ наборов z. На проверку допустимости каждой пары наборов α и z потребуется время $O(\mathrm{tw})$. Суммарно обработка узла исключения занимает время $O(\mathrm{tw} B^{2\mathrm{tw}})$.

Поскольку в $\mathbb{T}$ всего $O(n\,\mathrm{tw})$ узлов, решение исходной задачи восстанавливается обратным ходом по узлам дерева T за время $O(\mathrm{tw}\, n\, B)$, а общее время работы алгоритма динамического программирования при фиксированном $\mathrm{tw} = O(1)$ составляет $O(nB^{2\mathrm{tw}})$.

Теорема 1 доказана.

З а м е ч а н и е 3. Пользуясь сведением из [14, разд. 2], можно за время $O(n)$ свести пример задачи RCFLP (Restricted Capacitated Facility Location Problem), в которой кроме ограничений на пропускные способности ребер (1.3) заданы еще и ограничения на производственные мощности предприятий, на графе G к примеру задачи RFLP на графе G', при этом $\mathrm{tw}(G) = \mathrm{tw}(G')$ и $|V(G')| \le 2|V(G)|$. Таким образом, предложенный выше алгоритм можно использовать для получения решений задачи RCFLP с теми же оценками трудоемкости.

4. Задача RFLP с делимыми спросами на путевом графе

В предыдущих разделах мы показали, что хотя задача RFLP с *делимыми спросами* NP-трудна даже на графах с древесной шириной $\mathrm{tw} = 1$, она может быть решена за псевдополиномиальное время, если $\mathrm{tw} = O(1)$. Для более простого случая задачи RFLP с *делимыми спросами* на путевом графе известен полиномиальный алгоритм с трудоемкостью $O(n^3)$ [14]. В этом разделе мы покажем, что задача RFLP с делимыми спросами на путевом графе может быть решена за время $O(n)$.

Для решения нашей задачи адаптируем алгоритм из работы [17], который решает неограниченную задачу размещения (1.1), (1.2), (1.4) на путевом графе за время $O(n)$. Алгоритм из [17] в свою очередь опирается на следующий результат из работы [27].

Определение 3. Функция $h\colon \{1,\dots,n\}\times\{1,\dots,n\}\to\mathbb{R}$ *вогнутая*, если для любых $1\le i_1<i_2<j_1<j_2\le n$

$$h(i_1,j_1)+h(i_2,j_2)\le h(i_1,j_2)+h(i_2,j_1). \tag{4.1}$$

Утверждение 4 [27, разд. 3–4]. *Существует алгоритм, находящий за время $O(n)$ решение одномерных динамических программ вида*

$$H(j)=\min_{1\le i\le j}\{H(i)+h(i,j)\},\quad j=1,\dots,n,$$

если функция $h(i,j)$ вогнутая, и значение функции $h(i,j)$ для каждой пары $1\le i<j\le n$ можно получить за время $O(1)$.

Далее в разд. 4.1 мы опишем основные идеи алгоритма из [17], в разд. 4.2 покажем, как на основе этих идей получить решение задачи RFLP с делимыми спросами на путевом графе за время $O(n)$.

4.1. Алгоритм для неограниченной задачи размещения на путевом графе

Рассмотрим неограниченную задачу размещения на путевом графе. Пусть $V=\{1,\dots,n\}$ — вершины заданного пути $G=(V,E)$. *Отрезком* $[j,k]$ будем называть простой подпуть в G из вершины j в вершину k. Введем функцию $w(j,k)$, определенную для $1\le j<k\le n$, равную суммарным транспортным затратам на обслуживание всех клиентов отрезка $[j,k]$ из вершины j, и функцию $\bar w(j,k)$, равную суммарным транспортным затратам на обслуживание всех клиентов отрезка $[j,k]$ из вершины k:

$$w(j,k)=\sum_{t=j}^{k}b_t g_{tj},\quad \bar w(j,k)=\sum_{t=j}^{k}b_t g_{tk},\quad 1\le j\le k\le n. \tag{4.2}$$

Для фиксированной пары $j<k$ значение $w(j,k)$ и $\bar w(j,k)$ можно получать за время $O(1)$, если предварительно провести следующие вычисления.

Утверждение 5 [17, разд. 2]. *Пусть $B_j=\sum_{t=1}^{j}b_t$, $C_1=0$ и $C_j=\sum_{t=2}^{j}c_{t-1,t}$, $D_j=\sum_{t=1}^{j}b_tC_t$ для всех $j=1,\dots,n$. Все эти величины можно вычислить рекурсивно за общее время $O(n)$. Тогда*

$$w(j,k)=D_k-D_{j-1}-(B_k-B_j)C_j,\quad \bar w(j,k)=-D_k+D_{j-1}+(B_k-B_j)C_k.$$

Утверждение 6 [17, лемма 1]. *Функции $w(j,k)$ и $\bar w(j,k)$ вогнутые.*

Через $F(j)$ обозначим оптимальное решение подзадачи на вершинах $\{j,j+1,\dots,n\}$ для $1\le j\le n$. Через $G(j)$ обозначим оптимальное решение подзадачи на вершинах $\{j,j+1,\dots,n\}$ такое, что в вершине j открыто предприятие. Тогда верны следующие рекуррентные соотношения [17]: для каждого $j=n,\dots,1$

$$G(j)=f_j+\min_{j<k\le n+1}\{w(j,k)+F(k)\},\quad F(j)=\min_{j\le k\le n}\{\bar w(j,k)+G(k)\}, \tag{4.3}$$

и величина $F(1)$ соответствует оптимуму целевой функции исходной задачи. Обратным ходом можно восстановить решение за время $O(n)$.

Поскольку функции $w(j,k)$ и $\bar w(j,k)$ вогнутые, умея получать их значения за $O(1)$, согласно утверждению 2, с помощью алгоритма из [27] можно вычислить значения $F(j)$ и $G(j)$ для всех j за суммарное время $O(n)$. Учитывая трудоемкость предварительных построений для быстрого вычисления величин $w(j,k)$ и $\bar w(j,k)$, а также трудоемкость обратного хода, суммарное время работы данного алгоритма для неограниченной задачи размещения на путевом графе составляет $O(n)$ [17].

4.2. Линейный алгоритм для задачи RFLP на путевом графе

В этом разделе покажем, как за время $O(n)$ свести задачу RFLP с *делимыми спросами* на путевом графе к решению динамической программы, аналогичной (4.3), с вогнутыми слагаемыми.

Согласно утверждению 3 для задачи RFLP с делимыми спросами существует оптимальное решение, в котором все потоки продукта по ребру имеют одинаковое направление. Легко заметить, что для задачи на путевом графе такое оптимальное решение будет *центрально-связным*, т. е. будет разбивать путь на отрезки, в каждом из которых открыто ровно одно предприятие, полностью обслуживающее всех клиентов этого отрезка, за исключением, быть может, клиентов в крайних вершинах отрезка. Далее для задачи будем искать именно центрально-связные решения.

Для каждой вершины $i = 1, \ldots, n$, в которой $f_i < \infty$, определим *границы возможной области обслуживания*, т. е. самую левую и самую правую вершины $\ell_i \le i \le r_i$ такие, что ограничения на пропускные способности ребер позволяют предприятию, открытому в i, полностью удовлетворить спрос всех клиентов на отрезке $[\ell_i, r_i]$, после чего у него еще останется возможность доставить $\alpha_i \in [0, b_{\ell_i - 1})$ единиц продукта клиенту в вершине $\ell_i - 1$ и $\beta_i \in [0, b_{r_i+1})$ единиц продукта клиенту в вершине $r_i + 1$. Границу области обслуживания ℓ_i (r_i) будем называть *строгой*, если $\alpha_i = 0$ ($\beta_i = 0$), и *нестрогой*, в ином случае.

Лемма 1. *Номера вершин ℓ_i и r_i, а также величины α_i и β_i для всех $i = 1, \ldots, n$ можно найти суммарно за время $O(n)$.*

Д о к а з а т е л ь с т в о. Покажем, как, двигаясь от вершины 1 к вершине n, вычислить величины r_i и β_i за время $O(n)$. Аналогичным способом, двигаясь от n к 1, можно вычислить все величины ℓ_i и α_i.

Заметим, что $r_{i+1} \ge r_i$ для любого $i = 1, \ldots, n-1$, поскольку если ограничения на пропускные способности ребер позволяют из вершины i полностью обслужить отрезок $[i, r_i]$, то и из вершины $i+1$ возможно полностью обслужить отрезок $[i+1, r_i]$. Воспользуемся этим свойством в процедуре поиска величин r_i и β_i. В процедуре последовательно просматриваются вершины i пути G, и для каждого $j > i$ проверяется, возможно ли из вершины i в дополнение к обслуженному отрезку $[i, j-1]$ обслужить клиента в j. Если возможно, процедура переходит к проверке клиента $j+1$, если нет, то величина r_i найдена, и нужно проверять, возможно ли обслужить клиента j из вершины $i+1$.

Приведем формальное описание процедуры. Пусть, как и раньше, $B_j := \sum_{t=1}^{j} b_t$. Для пары вершин $i < j$ через $\mathrm{dif}_{i,j} \ge 0$ обозначим величину, на которую можно увеличить поток продукции из вершины i после того как из i полностью обслужены клиенты отрезка $[i, j-1]$:

$$\mathrm{dif}_{i,j} = \min_{i<j'<j} \{q(\{j'-1, j'\}) - (B_j - B_{j'-1})\} = \min_{i<j'<j} \{q(\{j'-1, j'\}) + B_{j'-1}\} - B_j.$$

В работе [5, разд. 4–5] показано, что для заданного массива $A[1, \ldots, n]$, потратив на предобработку массива время $O(n)$, можно за время $O(1)$ находить номер минимального элемента в подмассиве $A[i, i+1, \ldots, j]$. Используя алгоритм из [5] для массива с элементами $A[j'] = q(\{j'-1, j'\}) + B_{j'-1}$, $j' = 1, \ldots, n$, для каждой пары $i < j$ значение $\mathrm{dif}_{i,j}$ можно вычислить за время $O(1)$.

П р о ц е д у р а в ы ч и с л е н и я r_i и β_i.

1. *Положим* $r_k := n$, $\beta_k := 0$, $B_0 := 0$, $B_k := B_{k-1} + b_k$ для всех $1 \le k \le n$,
   а также $i := 1$, $j := 2$.
2. *Вычислим* элементы массива A и выполним его предобработку алгоритмом из [5].
3. *Пока* $j \le n$ и $i < n$:
4. *Если* $b_j \le \mathrm{dif}_{i,j}$ и $b_j \le q(\{j-1, j\})$, то $j := j+1$.
5. *Иначе*: $r_i := j-1$, $\beta_i := \min\{\mathrm{dif}_{i,j}, q(\{j-1, j\})\}$ и $i := i+1$.

Шаги 1, 2 выполняются за время $O(n)$. На каждой итерации цикла (3–5) либо j, либо i увеличивается на 1, поэтому общее число итераций не превосходит $2n$. Все действия внутри одной итерации цикла требуют $O(1)$ времени. Таким образом, трудоемкость процедуры равна $O(n)$.

Лемма 2. *Если для каждой вершины известны границы областей обслуживания, пример исходной задачи* RFLP *на путевом графе с* n *вершинами можно за время* $O(n)$ *свести к эквивалентному примеру задачи* RFLP *на путевом графе с* $O(n)$ *вершинами, в которой* $\alpha_i = \beta_i = 0$ *для каждой вершины* i.

Доказательство. Для каждой вершины $i = 1, \ldots, n$ выполним следующее. Если $\alpha_i > 0$, уменьшим спрос в вершине $\ell_i - 1$ на α_i и добавим в путевом графе между вершинами ℓ_i и $\ell_i - 1$ вершину v_i со спросом α_i и бесконечной стоимостью открытия предприятия. Стоимость транспортировки единицы товара по ребру $\{v_i, \ell_i\}$ и его пропускную способность положим такими же, как были у ребра $\{\ell_i - 1, \ell_i\}$, а для ребра $\{\ell_i - 1, v_i\}$ положим $c(\{\ell_i - 1, v_i\}) = 0$ и $q(\{\ell_i - 1, v_i\}) = \infty$. Аналогичным образом, если $\beta_i > 0$, уменьшим спрос в вершине $r_i + 1$ на β_i и добавим между вершинами r_i и $r_i + 1$ вершину u_i со спросом β_i и бесконечной стоимостью открытия предприятия. Таким образом, вершины v_i и u_i — это новые строгие границы области обслуживания для вершины i.

В ходе описанной выше процедуры для каждой вершины исходного графа было добавлено не более двух дополнительных вершин, поэтому общее число вершин теперь не превосходит $3n$. Кроме того, поскольку в задаче спросы делимые, очевидно, что для любого допустимого решения исходного примера существует допустимое решение той же стоимости в новом примере, и наоборот.

Лемма 3. *Для задачи* RFLP *с делимыми спросами на путевом графе, в которой для каждой вершины* i *границы возможного интервала обслуживания строгие* ($\alpha_i = \beta_i = 0$)*, существует центрально-связное оптимальное решение, в котором спрос каждого клиента обслуживается ровно одним предприятием.*

Доказательство. Рассмотрим пример задачи RFLP с делимыми спросами на путевом графе, в которой для каждой вершины i выполняется $\alpha_i = \beta_i = 0$. Пусть (S, x) — оптимальное центрально-связное решение этой задачи. Пусть $i_1, i_2 \in S$ — два последовательных предприятия в этом решении такие, что $i_1 < j < i_2$, i_1 обслуживает всех клиентов отрезка $[i_1, j-1]$ и часть $b \in [0, b_j]$ спроса клиента j, а i_2 обслуживает всех клиентов отрезка $[j+1, i_2]$ и часть $(b_j - b)$ спроса клиента j. Тогда суммарные затраты на обслуживание клиента j в этом решении составляют

$$b \sum_{k=i_1+1}^{j} c(\{k-1, k\}) + (b_j - b) \sum_{k=j+1}^{i_2-1} c(\{k-1, k\}),$$

где $0 \leq b \leq b_j$. Это линейная по b функция, следовательно, она достигает своего минимума либо при $b = b_j$, либо при $b = 0$, что соответствует полному обслуживанию спроса клиента j либо из i_1, либо из i_2. Для каждой из вершин i_1 и i_2 по условию леммы 3 границы интервалов обслуживания строгие, и в решении (S, x) каждая из них частично обслуживает j. Следовательно, у каждой из них есть возможность обслужить вершину j полностью. Значит, найдется решение (S, x'), стоимость которого не больше стоимости решения (S, x) и в котором все клиенты отрезка $[i_1, i_2]$ обслуживаются ровно одним предприятием. Проведя аналогичные рассуждения для каждой пары последовательных предприятий $i_1, i_2 \in S$, получим оптимальное центрально-связное решение (S, x''), в котором спрос каждого клиента обслуживается ровно одним предприятием.

Теорема 2. *Для задачи* RFLP *с делимыми спросами на путевом графе существует точный алгоритм, работающий за время* $O(n)$.

Д о к а з а т е л ь с т в о. Центрально-связное оптимальное решение задачи найдем следующим образом. Пусть $G = (V, E)$ — исходный путевой граф с вершинами $V = \{1, \dots, n\}$. За время $T_1 = O(n)$ с помощью процедуры из леммы 1 для каждой вершины $i \in V$ найдем величины $\ell_i, r_i, \alpha_i, \beta_i$. За время $T_2 = O(n)$ с помощью процедуры из леммы 2 построим эквивалентный пример на путевом графе $G' = (V', E')$, где $|V'| = O(n)$ и для каждой вершины $i \in V'$ величины $\alpha_i = \beta_i = 0$. Далее покажем, что решение задачи в графе G' может быть получено алгоритмом динамического программирования, аналогичным (4.3).

Согласно лемме 3 для задачи на графе G' существует центрально-связное оптимальное решение, в котором спрос каждого клиента обслуживается ровно одним предприятием. В таком решении каждая вершина i обслуживает некоторый отрезок $[j'_i, j''_i] \subseteq [\ell_i, r_i]$, причем суммарные транспортные затраты на обслуживание $[j'_i, j''_i]$ из i равны $w(i, j''_i) + \bar{w}(j'_i, i)$, где величины $w(i, j''_i)$ и $\bar{w}(j'_i, i)$ определяются согласно (4.2). В центрально-связном решении задачи RFLP обслуживание клиента $j \notin [\ell_i, r_i]$ из i невозможно, в этом случае транспортные затраты можно считать равными ∞. Введем функции $w'(j, k)$ и $\bar{w}'(j, k)$, определяющиеся через функции из (4.2): для $j, k \in V'$

$$w'(j,k) = \begin{cases} w(j,k), & \text{если } k \le r_j, \\ \infty, & \text{иначе;} \end{cases} \qquad \bar{w}'(j,k) = \begin{cases} \bar{w}(j,k), & \text{если } j \ge \ell_k, \\ \infty, & \text{иначе.} \end{cases} \tag{4.4}$$

Согласно утверждению 5 с помощью предварительных вычислений, требующих $T_3 = O(n)$ времени, для произвольной пары $j < k$ значения $w'(j, k)$ и $\bar{w}'(j, k)$ можно получать за время $O(1)$. Покажем, что функции $w'(j, k)$ и $\bar{w}'(j, k)$ вогнутые. Допустим, на наборе $1 \le j_1 < j_2 < k_1 < k_2 \le |V'|$ для функции $w'(j, k)$ не выполняется условие вогнутости (4.1), т. е.

$$w'(j_1, k_1) + w'(j_2, k_2) > w'(j_1, k_2) + w'(j_2, k_1). \tag{4.5}$$

Если в (4.5) все слагаемые $< \infty$, получим противоречие с утверждением 6 для функции $w(j, k)$. Если $w'(j_1, k_2) = \infty$ или $w'(j_2, k_1) = \infty$, получим противоречие со строгостью неравенства (4.5). Если $w'(j_2, k_2) = \infty$ или $w'(j_1, k_1) = \infty$, то из определения (4.4) следует $w'(j_1, k_2) = \infty$, что снова противоречит строгости неравенства (4.5). Следовательно, функция $w'(j, k)$ вогнутая. Вогнутость $\bar{w}'(j, k)$ доказывается аналогично.

Наконец, в графе G' оптимальное центрально-связное решение задачи RFLP, в котором спрос каждого клиента обслуживается ровно одним предприятием, можно найти по аналогии с (4.3) путем решения динамической программы: для каждого $j = |V'|, \dots, 1$

$$G(j) = f_j + \min_{j<k\le n+1} \{w'(j,k) + F(k)\}, \quad F(j) = \min_{j\le k\le n} \{\bar{w}'(j,k) + G(k)\}. \tag{4.6}$$

Поскольку $w'(j, k)$ и $\bar{w}'(j, k)$ вогнутые и каждое значение для них можно вычислить за время $O(1)$, решение (4.6) может быть найдено за время $T_4 = O(n)$ согласно [17; 27]. Из этого решения за время $T_5 = O(n)$ очевидным образом восстанавливается решение исходной задачи в графе G. Таким образом, общее время работы алгоритма решения исходной задачи составляет $T_1 + \dots + T_5 = O(n)$. Теорема 2 доказана.

Подход из теоремы 2 можно адаптировать для решения задачи p-RFLP с делимыми спросами, в решении которой должно быть открыто не более p предприятий, а также для некоторых частных случаев задачи RFLP с неделимыми спросами на путевом графе.

Следствие 1. *Для задачи p-RFLP с делимыми спросами на n-вершинном путевом графе существует точный алгоритм со временем работы $O(pn)$.*

Д о к а з а т е л ь с т в о. Достаточно в алгоритме из теоремы 2 вместо рекуррентных соотношений (4.3) использовать соотношения из [17] для p-RFLP: для каждого $j = |V'|, \dots, 1$ и каждого $i = 1, \dots, p$

$$G^i(j) = f_j + \min_{j<k\le n+1} \{w'(j,k) + F^{i-1}(k)\}, \quad F^i(j) = \min_{j\le k\le n} \{\bar{w}'(j,k) + G^i(k)\}.$$

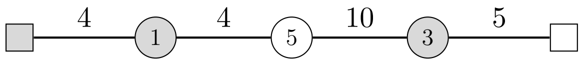

Рис. 2. Пример задачи RFLP с неделимыми спросами, в котором единственное допустимое решение не центрально-связно. Квадратами обозначены вершины, в которых стоимость открытия предприятий меньше ∞, в круглых вершинах указан спрос клиентов, над ребрами указаны их пропускные способности. В решении серое предприятие обслуживает клиентов в серых вершинах, а белое — в белых.

Как было показано в утверждении 2, задача RFLP с неделимыми спросами на путевом графе NP-трудна. Легко заметить, что центрально-связных оптимальных решений для этой задачи может не быть (см. рис. 2). Однако в тех случаях, когда для задачи RFLP с неделимыми спросами существуют центрально-связные оптимальные решения, их можно найти с помощью алгоритма из теоремы 2, в котором пропускается шаг из леммы 2.

Следствие 2. *Задача* RFLP *с неделимыми спросами на путевом графе с n вершинами может быть решена за время $O(n)$, если*

∘ *спросы всех клиентов одинаковые или*

∘ *для всех вершин i границы областей обслуживания строгие ($\alpha_i = \beta_i = 0$) в терминах, введенных для задачи* RFLP *с делимыми спросами.*

Заключение

В работе рассматривались частные случаи задачи RFLP на путевых графах и графах с ограниченной древесной шириной и исследовалась возможность их точного решения. Положительные алгоритмические результаты были получены для задачи RFLP с делимыми спросами. Эта задача слабо NP-трудна даже на простейших типах деревьев, однако, для нее удалось построить псевдополиномиальный алгоритм на графах с ограниченной древесной шириной и линейный алгоритм на путевых графах. С другой стороны, мы показали, что задача RFLP с неделимыми спросами сильно NP-трудна на деревьях и NP-трудна на путевых графах, хотя при некоторых ограничениях на значения пропускных способностей или спросы клиентов RFLP с неделимыми спросами на путевых графах может быть решена за линейное время. Вопрос о существовании псевдополиномиального алгоритма для задачи RFLP с неделимыми спросами на путевом графе в общем случае остается открытым.

СПИСОК ЛИТЕРАТУРЫ

1. **Adamaszek A., Chalermsook P., Ene A., Wiese A.** Submodular unsplittable flow on trees // Math. Program. 2018. Vol. 172. P. 565–589. doi: 10.1007/s10107-017-1218-4 .
2. **Ageev A. A.** A criterion of polynomial-time solvability for the network location problem // Integer Programming and Combinatorial Optimization. Proc. IPCO II Conf. Campus Printing. Pittsburgh: Carnegie Mellon University, 1992. P. 237–245.
3. **Andreev K., Garrod C., Golovin D., Maggs B., Meyerson A.** Simultaneous source location // ACM Trans. Algorithms. 2010. Vol. 6, no. 1. P. 16:1–16:17. doi: 10.1145/1644015.1644031 .
4. **Arata K., Iwata S., Makino K., Fujishige S.** Locating sources to meet flow demands in undirected networks // J. Algorithms. 2002. Vol. 42. P. 54–68. doi: 10.1006/jagm.2001.1203 .
5. **Bender M. A., Farach-Colton M.** The LCA problem revisited // 4th Latin American Symposium on Theoretical Informatics (LATIN). 2000. Lecture Notes in Computer Science; vol. 1776. P. 88–94. doi: 10.1007/10719839_9 .
6. **Bodlaender H. L.** A linear time algorithm for finding tree-decompositions of small treewidth // SIAM J. Computing. 1996. Vol. 25, no. 6. P. 1305–1317. doi: 10.1137/S0097539793251219 .
7. **Bodlaender H. L.** Treewidth: Algorithmic techniques and results // Proc. of the 22nd Internat. Symposium on Mathematical Foundations of Computer Science (MFCS'97). Berlin, Heidelberg: Springer, 1997. P. 19–36. doi: 10.1007/BFb0029946 .

8. **Bodlaender H. L.** A partial k-arboretum of graphs with bounded treewidth // Theoretical Computer Science. 1998. Vol. 209, iss. 1–2. P. 1–45. doi: 10.1016/S0304-3975(97)00228-4 .
9. **Bodlaender H. L.** Treewidth: Characterizations, applications, and computations // Graph-Theoretic Concepts in Computer Science (WG 2006) / ed. F.V. Fomin. Lecture Notes in Computer Science; vol. 4271. 2006. P. 1–14. doi: 10.1007/11917496_1 .
10. **Chekuri C., Ene A., Korula N.** Unsplittable flow in paths and trees and column-restricted packing integer programs // Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX 2009, RANDOM 2009) / eds. I. Dinur et al. 2009. Lecture Notes in Computer Science; vol. 5687. P. 42–55. doi: 10.1007/978-3-642-03685-9_4 .
11. **Cornuéjols G., Nemhauser G. L., Wolsey L. A.** The uncapacitated facility location problem // Discrete location theory / eds. P. B. Mirchandani and R. L. Francis. N Y: Wiley, 1990. P. 119–171.
12. **Cygan M., Fomin F.V., Kowalik L., Lokshtanov D., Marx D., Pilipczuk M., Pilipczuk M., Saurabh S.** Parameterized algorithms. Cham: Springer, 2015. 613 p. doi: 10.1007/978-3-319-21275-3 .
13. **Dinitz Y., Garg N., Goemans M.X.** On the single-source unsplittable flow problem // Proc. 39th Annual Symposium on Foundations of Computer Science. Palo Alto, CA, USA, 1998. P.290–299. doi: 10.1109/SFCS.1998.743461 .
14. **Gimadi E. Kh., Kurochkina A., Tsidulko O.** On exact solvability of the restricted capacitated facility location problem // CEUR-WS. 2017. Vol. 1987. P. 209–216.
15. **Granot D., Skorin-Kapov D.** On some optimization problems on k-trees and partial k-trees // Discrete Appl. Math. 1994. Vol. 48, no 2. P. 129–145. doi: 10.1016/0166-218X(92)00122-3 .
16. **Guruswami V., Khanna S. , Rajaraman R. , Shepherd B., Yannakakis M.** Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems // J. Computer System Sci. 2003. Vol. 67, no. 3, P. 473–496. doi: 10.1016/S0022-0000(03)00066-7 .
17. **Hassin R., Tamir A.** Improved complexity bounds for location problems on the real line // Operations Research Letters. 1991. Vol. 10, no. 7. P. 395–402. doi: 10.1016/0167-6377(91)90041-M .
18. **Kleinberg J.M.** Approximation algorithms for disjoint paths problems // Ph.D. dissertation. M.I.T. 1996. 188 p.
19. **Koivisto M., Manila H., Perola M., Varilo T., Hennah W.,Ekelund J., Lukk M., Peltonen L., Ukkonen E.** An MDL method for finding haplotype blocks and for estimating the strength of block boundaries // Pacific Symposium on Biocomputing, 2003. P. 502–513. doi: 10.1142/9789812776303_0047 .
20. **Laporte G., Nickel S., Saldanha da Gama F.** Location science. Switzerland: Springer Intern. Publ., 2015. 650 p.
21. **Lazic N., Givoni I. E., Frey B. J., Aarabi P.** FLoSS: Facility location for subspace segmentation // IEEE 12th Intern. Conf. on Computer Vision. Kyoto, 2009. P. 825–832. doi: 10.1109/ICCV.2009.5459302 .
22. **Robertson N., Seymour P.D.** Graph minors. II. Algorithmic aspects of tree-width. // J. Algorithms. 1986. Vol 7, no. 3, P. 309–322. doi: 10.1016/0196-6774(86)90023-4 .
23. **Shah R., Farach-Colton M.** Undiscretized dynamic programming: Faster algorithms for facility location and related problems on trees // Proc. of the thirteenth annual ACM-SIAM symposium on Discrete algorithms (SODA). 2002. P. 108–115.
24. **Turner L., Gross D. P., Hamacher H. W., Krumke S. O.** Static and dynamic source locations in undirected networks // TOP. Vol. 23. 2015. P. 619–646. doi: 10.1007/s11750-015-0395-7 .
25. **Вознюк И. П.** Задача размещения на сети с ограниченными пропускными способностями коммуникаций // Дискретный анализ и исследование операций. Сер. 2. 1999. Т. 6, вып. 1. С. 3–11.
26. **Вознюк И. П.** Задача размещения пунктов производства на два-дереве с ограниченными пропускными способностями коммуникаций // Дискретный анализ и исследование операций. Сер. 2. 2000. Т. 7, вып. 1. С. 3–8.
27. **Wilber R.** The concave least-weight subsequence problem revisited // J. Algorithms. 1988. Vol. 9. P. 418–425. doi 10.1016/0196-6774(88)90032-6 .

Поступила 24.03.2020
После доработки 14.05.2020
Принята к публикации 18.05.2020

Гимади Эдуард Хайрутдинович
д.-р. физ.-мат. наук, профессор

главный науч. сотрудник
Институт математики им С. Л. Соболева СО РАН;
Новосибирский государственный университет
г. Новосибирск
e-mail: gimadi@math.nsc.ru

Цидулко Оксана Юрьевна
канд. физ.-мат. наук
науч. сотрудник
Институт математики им С. Л. Соболева СО РАН;
Новосибирский государственный университет
г. Новосибирск
e-mail: tsidulko@math.nsc.ru

REFERENCES

1. Adamaszek A., Chalermsook P., Ene A., Wiese A. Submodular unsplittable flow on trees. *Math. Program.* 2018, vol. 172, pp. 565–589. doi: 10.1007/s10107-017-1218-4 .
2. Ageev A. A. A criterion of polynomial-time solvability for the network location problem. *Integer Programming and Combinatorial Optimization. Proc. IPCO II Conf.* Campus Printing. Pittsburgh: Carnegie Mellon University, 1992, pp. 237–245.
3. Andreev K., Garrod C., Golovin D., Maggs B., Meyerson A. Simultaneous source location *ACM Trans. Algorithms*, 2010, vol. 6, no. 1, pp. 16:1–16:17. doi: 10.1145/1644015.1644031 .
4. Arata K., Iwata S., Makino K., Fujishige S. Locating sources to meet flow demands in undirected networks. *J. Algorithms*, 2002, vol. 42, pp. 54–68. doi: 10.1006/jagm.2001.1203 .
5. Bender M. A., Farach-Colton M. The LCA problem revisited. *4th Latin American Symposium on Theoretical Informatics* (LATIN). 2000, LNCS; vol. 1776, pp. 88–94. doi: 10.1007/10719839_9 .
6. Bodlaender H. L. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. on Computing*, 1996, vol. 25, no. 6, pp. 1305–1317. doi: 10.1137/S0097539793251219 .
7. Bodlaender H. L. Treewidth: Algorithmic techniques and results. In: *Proceedings of the 22nd International Symposium on Mathematical Foundations of Computer Science* (MFCS'97). Berlin, Heidelberg: Springer, 1997, pp. 19–36. doi: 10.1007/BFb0029946
8. Bodlaender H. L. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 1998, vol. 209, no. 1–2, pp. 1–45. doi: 10.1016/S0304-3975(97)00228-4 .
9. Bodlaender H. L. Treewidth: Characterizations, applications, and computations. In: F.V. Fomin (ed.), *Graph-Theoretic Concepts in Computer Science* (WG 2006). 2006, Lecture Notes in Computer Science; vol 4271, pp. 1–14. doi: 10.1007/11917496_1 .
10. Chekuri C., Ene A., Korula N. (2009) Unsplittable flow in paths and trees and column-restricted packing integer programs. In: Dinur I. et al. (eds), *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques* (APPROX 2009, RANDOM 2009). 2009. Lecture Notes in Computer Science; vol. 5687, pp .42–55. doi: 10.1007/978-3-642-03685-9_4 .
11. Cornuéjols G., Nemhauser G. L., Wolsey L. A. The uncapacitated facility location problem. *Discrete location theory.* Mirchandani P. B. and Francis R. L. (eds). N Y: Wiley, 1990, pp. 119–171.
12. Cygan M., Fomin F.V., Kowalik L., Lokshtanov D., Marx D., Pilipczuk M., Pilipczuk M., Saurabh S. *Parameterized algorithms.* Cham: Springer, 2015, 613 p. doi: 10.1007/978-3-319-21275-3 .
13. Dinitz Y., Garg N., Goemans M.X. On the single-source unsplittable flow problem. *Proc. 39th Annual Symposium on Foundations of Computer Science.* Palo Alto, CA, USA, 1998, pp. 290–299. doi: 10.1109/SFCS.1998.743461 .
14. Gimadi E. Kh., Kurochkina A., Tsidulko O. On exact solvability of the restricted capacitated facility location problem. *CEUR-WS*, 2017, vol. 1987, pp. 209–216.
15. Granot D., Skorin-Kapov D. On some optimization problems on k-trees and partial k-trees. *Discrete Appl. Math.*, 1994, vol. 48, no. 2, pp. 129–145. doi: 10.1016/0166-218X(92)00122-3 .
16. Guruswami V., Khanna S., Rajaraman R., Shepherd B., Yannakakis M. Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems. *J. Computer and System Sciences*, 2003, vol. 67, no. 3, pp. 473–496. doi: 10.1016/S0022-0000(03)00066-7 .

17. Hassin R., Tamir A. Improved complexity bounds for location problems on the real line. *Operations Research Letters*, 1991, vol. 10, no. 7, pp. 395–402. doi: 10.1016/0167-6377(91)90041-M .
18. Kleinberg J.M. *Approximation algorithms for disjoint paths problems.* Ph.D. dissertation, M.I.T., 1996, 188 p.
19. Koivisto M., Manila H., Perola M., Varilo T., Hennah W., Ekelund J., Lukk M., Peltonen L., Ukkonen E. An MDL method for finding haplotype blocks and for estimating the strength of block boundaries. In: *Pacific Symposium on Biocomputing*, 2003, pp. 502–513. doi: 10.1142/9789812776303_0047 .
20. Laporte G., Nickel S., Saldanha da Gama F. *Location science*. Switzerland: Springer Intern. Publ., 2015, 650 p.
21. Lazic N., Givoni I. E., Frey B. J., Aarabi P. FLoSS: Facility location for subspace segmentation. *IEEE 12th Intern. Conf. on Computer Vision*, Kyoto, 2009, pp. 825–832. doi: 10.1109/ICCV.2009.5459302 .
22. Robertson N., Seymour P.D. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 1986, vol. 7, no. 3, pp. 309–322. doi: 10.1016/0196-6774(86)90023-4 .
23. Shah R., Farach-Colton M. Undiscretized dynamic programming: Faster algorithms for facility location and related problems on trees. *Proc. of the thirteenth annual ACM-SIAM symposium on Discrete algorithms* (SODA), 2002, pp. 108–115.
24. Turner L., Gross D. P., Hamacher H. W., Krumke S. O. Static and dynamic source locations in undirected networks. *TOP.*, 2015, vol. 23, pp. 619–646. doi: 10.1007/s11750-015-0395-7 .
25. Voznyuk I.P. The location problem on networks with bounded communication capacities. *Diskretn. Anal. Issled. Oper.*, Ser. 2, 1999, vol. 6, no. 1, pp. 3–11 (in Russian).
26. Voznyuk I.P. The plant location problem on a two-tree with bounded communication capacities. *Diskretn. Anal. Issled. Oper.*, Ser. 2, 2000, vol. 7, no. 1, pp. 3–8 (in Russian).
27. Wilber R. The concave least-weight subsequence problem revisited. *J. Algorithms*, 1988, vol. 9. pp. 418–425. doi 10.1016/0196-6774(88)90032-6 .

Received March 24, 2020
Revised May 14, 2020
Accepted May 18, 2020

Funding Agency: This work was supported by the Russian Foundation for Basic Research (project no. 18-31-00470) and partially supported by Mathematical Center in Akademgorodok (agreement with Ministry of Science and High Education of the Russian Federation no. 075-15-2019-1675).

Edward Khairutdinovich Gimadi, Dr. Phys.-Math. Sci., Prof., Sobolev Institute of Mathematics; Novosibirsk State University, Novosibirsk, 630990 Russia, e-mail: gimadi@math.nsc.ru .

Oxana Yurievna Tsidulko, Cand. Sci. (Phys.-Math.), Sobolev Institute of Mathematics; Novosibirsk State University, Novosibirsk, 630990 Russia, e-mail: tsidulko@math.nsc.ru

Cite this article as: E. Kh. Gimadi, O. Yu. Tsidulko. On some efficiently solvable classes of the network facility location problem with constraints on the capacities of communication lines. *Trudy Instituta Matematiki i Mekhaniki URO RAN*, 2020, vol. 26, no. 2, pp. 108–124 .