САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

На правах рукописи

Ейбоженко Дмитрий Анатольевич

Приближенные методы решения задачи Штейнера на ориентированных графах

Специальность 05.13.11 — Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей.

АВТОРЕФЕРАТ

диссертации на соискание учёной степени кандидата физико-математических наук

Санкт-Петербург
2012

Работа выполнена на кафедре исследования операций математико-механического факультета Санкт-Петербургского государственного университета.

Научный руководитель:	доктор физико-математических наук, профессор Романовский Иосиф Владимирович
Официальные оппоненты:	доктор физико-математических наук, профессор Чирков Михаил Константинович (Санкт-Петербургский государственный университет)
	доктор технических наук, профессор Филиппова Анна Сергеевна (Уфимский государственный авиационный технический университет)
Ведущая организация:	Санкт-Петербургский экономико-математический институт РАН

Защита состоится «___» _______ 2012 г. в ___ часов на заседании совета Д 212.232.51 по защите докторских и кандидатских диссертаций при Санкт-Петербургском государственном университете по адресу: 198504, Санкт-Петербург, Петродворец, Университетский пр., д. 28, математико-механический факультет, ауд. 405.

С диссертацией можно ознакомиться в Научной библиотеке имени М. Горького Санкт-Петербургского государственного университета по адресу: 199034, Санкт-Петербург, Университетская наб., д. 7/9.

Автореферат разослан «___» __________ 2012 года.

Ученый секретарь диссертационного совета
доктор физико-математических наук,
доцент Кривулин Н.К.

Общая характеристика работы

Данная работа посвящена разработке приближенных методов решения задачи Штейнера на ориентированных графах, а также некоторых ее частных постановок, их теоретическому исследованию и обоснованию, программной реализации, оптимизации быстродействия с учетом текущего развития вычислительной техники, а также эмпирическому подтверждению их применимости.

Задача Штейнера на ориентированных графах определяется следующим образом:

Пусть $G(M, N)$ — ориентированный граф с заданной на дугах функцией $d : N \to \mathbb{R}_+$. В M выделена начальная вершина b и множество терминальных вершин E. Требуется найти дерево минимальной длины с корнем в заданной вершине b, содержащее пути от b до любого терминала из E.

Актуальность темы

Задача Штейнера в различных вариациях широко используется в таких передовых областях промышленности, как проектирование и производство интегральных микросхем, в т. ч. и микропроцессоров, телекоммуникации, в особенности при реализации современных технологических систем, таких как выборочное телевещание, интерактивные телеконференции, а также в некоторых областях биологии (филогенетике).

Как известно, задача Штейнера в графовой форме является NP-трудной [6]. Большинство исследований данной задачи ставят целью найти приемлемые с точки зрения быстродействия и точности решения приближенные алгоритмы, эвристики и аппроксимации, с применением широкого спектра подходов: жадных алгоритмов, методов динамического программирования, генетических методов, а также различных аппроксимаций, основанных на линеаризации задачи и релаксации полученных условий. Большой вклад в исследование различных постановок задачи Штейнера внесли А. Зеликовский, А. Иванов, А. Тужилин, M. Zachariasen, S. Arora, M. Hwang.

В то же время, вопрос о существовании более эффективных методов для решения данной задачи как с точки зрения быстродействия, так и точности, остается открытым. Кроме того, во многих практических задачах имеется дополнительная информация о структуре графа, что влечет интерес к разработке методов с большей точностью для таких, более узких, классов задач. Наконец, современное развитие вычислительной техники вызывает интерес к методам, обладающим большей способности к распараллеливанию, т. е. к декомпозиции на отдельные подзадачи, которые могут решаться одновременно, независимо друг от друга.

Цель работы

Целью данной работы является исследование ориентированной задачи Штейнера на графах и некоторых ее специальных постановок, разработка новых приближенных методов и подходов к ее решению и их программная оптимизация.

Методы исследования

При экспериментальных исследованиях точности разработанных алгоритмов и сравнения их с другими распространенными методами были использованы программные реализации их на языке C# 4.0. В качестве тестовой базы использовались задачи из базы задач SteinLib [8], а также сгенерированные автором наборы задач.

Для разработки параллельных реализаций методов была использована библиотека Microsoft Task Parallel Library.

Основные положения, выносимые на защиту

Среди полученных в ходе исследования результатов можно выделить следующие:

1. Разработан и реализован метод k-кластерной оптимизации для задачи Штейнера на ориентированных графах.

2. Разработан и реализован приближенный метод S^* для решения задачи Штейнера на *евклидовых* ориентированных графах.

3. Теоретически доказана полиномиальная вычислительная сложность обоих алгоритмов.

4. Проведены экспериментальные исследования данных методов и показана их практическая ценность.

5. Разработаны и реализованы параллельные версии метода динамического программирования и обоих представленных приближенных методов, проведены экспериментальные сравнения быстродействия параллельных и последовательных версий алгоритмов.

Научная новизна

Все результаты, выносимые на защиту, являются новыми.

Практическая значимость

В результате исследований было показано, что данные приближенные методы могут быть эффективно использованы для решения практических задач, а параллельная реализация наилучшим образом использует получающую все большее распространение в настоящее время многоядерную архитектуру центральных процессоров.

Кроме того, предложенные подходы, использованные в приведенных методах, являются новыми и могут быть исследованы далее для разработки новых приближенных методов.

Апробация работы

Основные результаты работы докладывались на российской конференции по проблемам дискретной оптимизации и исследования операций DOOR-2010, на межвузовской научной конференции по проблемам информатики СПИСОК-2011, международной конференции «Математика, экономика, менеджмент: 100 лет со дня рождения Л. В. Канторовича», а также на семинарах кафедры исследования операций математико-механического факультета СПбГУ.

Публикации по теме работы

Материалы диссертации опубликованы в пяти работах [1,2,3,4,5]. Из них работы [2,3] — в списке журналов, рекомендованных ВАК. Работа [2] выполнена в соавторстве: соискателю принадлежат доработка и реализация ограниченного метода динамического программирования, доказательство точности, проведение вычислительных экспериментов.

Объем и структура работы

Диссертация состоит из введения, пяти глав, заключения и приложения. Полный объем диссертации — 120 страниц текста с 18 рисунками и 2 таблицами. Список литературы содержит 69 наименований на 9 страницах.

Содержание работы

Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, формулируется цель, ставятся задачи работы, сформулированы научная новизна и практическая значимость представляемой работы.

Первая глава посвящена описанию, классификации и анализу основных направлений развития в исследовании задачи Штейнера на ориентированных графах. Внимание уделяется как методам точного, так и приближенного решения, в том числе методам динамического программирования и линейной релаксации, жадным алгоритмам, алгоритмам локальных улучшений, и т. д.

Одним из наиболее широко распространенных методов точного решения является алгоритм, основанный на методе динамического программирования и уравнении Беллмана. Задача Штейнера решается одновременно для любых начальных вершин $m \in M$ и любых возможных подмножеств $E_P \subseteq E$. Обозначим через (i, E_P) — состояние процесса, в котором решается задача Штейнера для начальной точки i и множества терминалов E_P, а через $v(i, E_P)$ — решение этой задачи. Необходимо найти $v(b, E)$.

Пусть процесс находится в состоянии (i, E_P). Считаем что $i \notin E_P$. В этом состоянии можно выбрать следующие решения:

– перейти по какой-либо дуге, начинающейся в i, в другую вершину, скажем, i_1, и решать задачу (i_1, E_P);

– разветвить путь, выбрав разбиение терминального множества $E_P = \bigcup_k E_{Pk}$, и для каждого состояния (i, E_{Pk}) решить такую же задачу.

Таким образом, получаем уравнение Беллмана:

$$v(i, E_P) = \min\{v_{\text{cont}}(i, E_P\{i\}), v_{\text{part}}(i, E_P\{i\})\}, \tag{1}$$

где v_{cont} — наименьшие затраты при наилучшем переходе по какой-либо дуге:

$$v_{\text{cont}}(i, E_P) = \min\{l_j + v(\mathbf{end}\, j, E_P) | \mathbf{beg}\, j = i\}, \tag{2}$$

а v_{part} — наименьшие затраты при наилучшем разбиении терминального множества E_P на два подмножества A и $E_P \setminus A$:

$$v_{\text{part}}(i, E_P) = \min\{v(i, A) + v(i, E_P \setminus A) | A \subset E_P\}. \tag{3}$$

При этом, если E_P состоит из одной вершины, то дерево Штейнера для состояния (i, E_P) — это кратчайший путь между этими вершинами, который можно вычислить по алгоритму Дейкстры.

Другой точный метод использует приведение задачи Штейнера к задаче целочисленного программирования и решения ее с помощью имеющихся в теории линейного программирования методов (метод ветвей и границ, отсечений Гомори и т. д.). Задача, получающаяся после приведения, выглядит следующим образом:

$$\min \quad \sum_{n \in N} c_n x_n, \tag{4}$$

$$\sum_{n \in \delta^-(W)} x_n \geq 1, \quad \forall W \in \rho, \tag{5}$$

$$x_n \in \{0, 1\}, \quad \forall n \in N, \tag{6}$$

а ее релаксация достигается заменой условий (6) на условия:

$$x_n \geq 0, \ \forall n \in N. \tag{7}$$

Здесь $\rho = \{W \subseteq M \setminus \{b\}, W \bigcap E \neq 0\}$, $\bar{W} = M \setminus W$, а $\delta^{-}(W)$ — это множество дуг $[v_i, v_j] \in N$, для которых $v_i \in \bar{W}$ и $v_j \in W$,

Наиболее широко известный приближенный метод — алгоритм Такахаши-Мацуямы [9], суть которого состоит в том, что на каждом шаге к уже имеющемуся дереву с корнем в b, содержащему некоторое подмножество E добавляется новый терминал из E вместе с кратчайшим путем, соединяющим дерево и терминал.

В главе также описывается общая схема жадного сокращения, которую используют многие приближенные алгоритмы [11]. В качестве первого приближения берется минимальное остовное дерево на графе G. Рассматриваются все полные (т. е. такие, все терминалы которых являются листьями) деревья Штейнера из определенного класса K, из них выбирается то дерево T, которое минимизирует некоторую оценочную функцию $f(T)$, и T сокращается (уменьшаются до нуля стоимости дуг минимального остовного дерева, заканчивающихся в терминалах, содержащихся в T). После того, как длина минимального остовного дерево становится равной нулю, из выбранных полных деревьев восстанавливается дерево Штейнера. Разные алгоритмы используют разные классы K (пути, звезды, трехтерминальные графы) и оценочные функции f.

Во **второй главе** представлен разработанный автором k-кластерный приближенный алгоритм для решения задачи Штейнера на ориентированных графах.

Общая идея данного алгоритма заключается в том, что исходный граф разбивается на не более чем k (параметр) непересекающихся подграфов, на каждом из которых ставится отдельная подзадача Штейнера, индуцированная исходной. Подзадачи, содержащие не более k терминалов, решаются с помощью точного метода. Подзадачи с бо́льшим числом терминалов в свою очередь вновь разбиваются на более мелкие подзадачи. Приведем данный алгоритм подробнее:

1. В графе G строится *опорное* дерево, являющееся некоторым деревом Штейнера для заданного множества терминалов. Для построения используется приведенный выше алгоритм Такахаши.

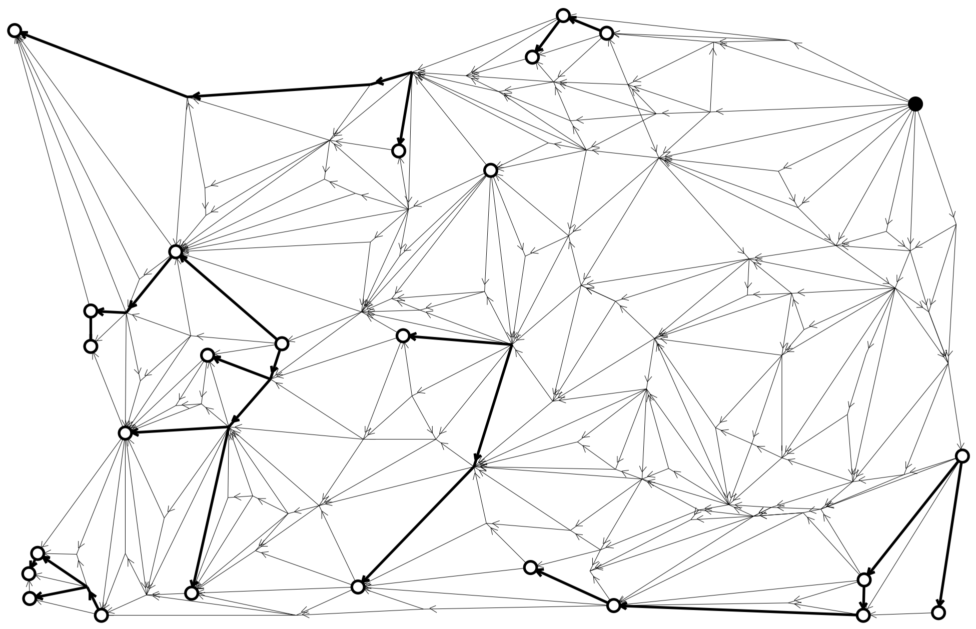

Рисунок 1 – Частичные поддеревья Штейнера

2. На основании опорного дерева строится разбиение на кластеры. Находится самая дальняя от начальной вершины b вершина ветвления опорного дерева, такая, что число терминалов в поддереве, ограниченном ей, больше k (это условие гарантирует, что кластеров будет не более k). Соответствующее ей поддерево удаляется из опорного дерева. Такие поддеревья выделяются до тех пор, пока опорное дерево не становится пустым. Затем на основании выделенных поддеревьев строится множество кластеров $\mathcal{C}$.

3. На каждом из кластеров $C \in \mathcal{C}$ определим *индуцированную задачу Штейнера* следующим образом: индуцированное множество терминалов $E(C) = E \cap M(C)$, начальная вершина $b(C)$ — вершина разветвления, на основании которой был построен кластер C. Если $|E(C)| \leqslant k$, то решаем поставленную задачу на подграфе методом динамического программирования, иначе вновь разбиваем полученный подграф на кластеры.

4. Исходный граф преобразуется с учетом найденных поддеревьев Штейнера: всем дугам, входящим в полученные частичные деревья Штейнера присваивается нулевая длина, терминалами новой задачи становятся начальные вершины кластеров (их менее k), из графа удаляются все дуги, входящие в любые вершины полученных частичных деревьев Штейнера,

кроме начальной. Поставленная задача решается методом динамического программирования.

5. Полученное дерево преобразуется с помощью алгоритма локальных улучшений.

Также в данной главе доказывается теорема о вычислительной сложности этого алгоритма, которая составляет $O(2^k tn \log m)$, где t — число терминалов, n — число дуг, m — число вершин.

В **третьей главе** представлен приближенный алгоритм для решения ориентированной задачи Штейнера на евклидовых графах.

Напомним, граф называется евклидовым, если его вершинам соответствуют некоторые точки в евклидовом пространстве, а длина дуги равна евклидовому расстоянию между точками.

Данный метод основан на идее алгоритма поиска кратчайшего пути A^* [7], который использует некоторую дополнительную эвристическую функцию (например, евклидово расстояние между двумя точками), чтобы оценить расстояние от текущей точки пути до конечной.

Идея представленного алгоритма заключается в том, чтобы значительно уменьшить число рассматриваемых подмножеств терминалов в методе динамического программирования, а именно, сделать их число полиномиальным относительно числа терминалов.

В главе приведен общий приближенный алгоритм решения, основанный на методе динамического программирования, для произвольного подмножества всего множества терминальных подмножеств. Кроме того, представлены три способа построения ограниченного множества терминальных подмножеств:

1. *Наивный метод.* Будем называть множество $J \subseteq E$ *b-допустимым*, если:

$$\nexists m' \in E \setminus J : \ \exists m_1, m_2 \in J : \angle(m_1 b m_2) = \angle(m_1 b m') + \angle(m_2 b m'). \quad (8)$$

Тогда определим множество терминальных подмножеств, получаемое наивным методом построения, как совокупность всевозможных b-допустимых подмножеств E.

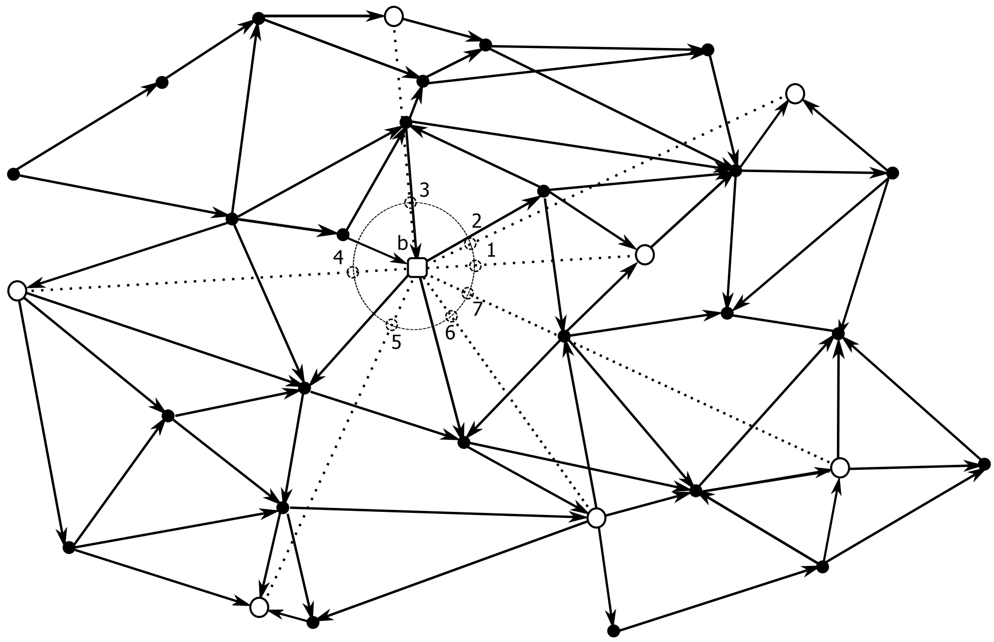

Рисунок 2 – Упорядочивание вершин в наивном методе

2. *Метод «концентрических окружностей».* Данный метод, в отличие от предыдущего, учитывает также расстояние от начальной вершины до терминалов. Здесь все множество терминалов разбивается определенным образом на группы, в зависимости от расстояния до начальной вершины. Далее в рамках каждой из групп терминальные подмножества выбираются аналогично наивному методу.

3. *Обобщенный метод.* Данный метод добавляет в результирующее множество терминальных подмножеств не только те, которые допустимы для начальной вершины, но и допустимые для любых других вершин. Для каждой вершины m ищутся допустимые подмножества на множестве терминалов, достижимых из нее, и добавляются в конечное множество.

Кроме того, в главе доказывается, что для любой задачи алгоритм, основанный на обобщенном методе построения множества терминальных подмножеств, выдаст некоторое решение, а также находится теоретическая оценка вычислительной сложности, и доказывается полиномиальность представленного алгоритма решения.

В **четвертой главе** рассматриваются вычислительные особенности реализации представленных в работе алгоритмов на языке C#, позволяющие уменьшить время решения и объем занимаемой памяти.

Среди заслуживающих упоминания программных особенностей отметим следующие:

- Стуктура хранения информации о графе, упрощающая обход всех инцидентных для произвольной вершины дуг, который активно используется алгоритмом Дейкстры. Дуги графа хранятся в порядке увеличения номера конечной вершины, а в структуре, соответствующей вершине, хранится индекс первой дуги с данной конечной вершиной, а также количество таких дуг.

- Пять различных реализаций приоритетной очереди, использующейся для хранения множества M_1 в алгоритме Дейкстры: в виде сортированных и несортированных списков, ведер, биномиальной кучи и фибоначчиевой кучи. Приведены теоретические оценки вычислительной сложности алгоритма Дейкстры при использовании данных реализаций.

- Система хранения и работы с терминальными подмножествами в виде целых чисел, каждый регистр двоичного представления которых отвечает за наличие в подмножестве определенной вершины. Реализованы операции с множествами для такого представления на основе побитовых логических операторов.

В последнее время основные производители микропроцессоров отказываются от традиционных подходов к увеличению производительности центральных процессоров. Вместо увеличения тактовой частоты и пропускной способности последовательных инструкций, они обращаются к технологиям мультиядерности и гипертрединга (несколько виртуальных потоков на одном ядре). Обе эти возможности уже широко используются в современных процессорах [12].

В главе представлены параллельные версии всех алгоритмов, выполненные с помощью библиотеки Microsoft Task Parallel Library, которые дают возможность значительно уменьшить время решения задач на компьютерах с многоядерными центральными процессорами. В главе приведены особенности реализаций параллельных программ, определены вычисления, выполняемые параллельно, моменты начала решения каждой из подзадач, выявлены необходимые критические секции и блокировки.

В **пятой главе** приведены результаты экспериментальных исследований представленных методов и их различных реализаций, производится их сравнение с другими известными приближенными методами решения задачи Штейнера, а также друг с другом. Для исследований использовались как известные наборы задач из библиотеки SteinLib [8], так и сгенерированые автором наборы задач.

В том числе, произведены следующие эксперименты:

1. Сравнение быстродействия пяти различных реализаций приоритетных очередей в рамках применения алгоритма динамического программирования на наборе из 200 задач с $|M| \in [80, 640]$, $|N| \in [120, 204480]$, $|T| \in [6, 17]$ из библиотеки SteinLib. Наилучшие результаты достигнуты при использовании приоритетных очередей на основе ведерной структуры.

2. Сравнение точности решения алгоритма k-кластерной оптимизации с другими известными алгоритмами (алгоритмом Такахаши, алгоритмом Зеликовского [10]) на наборе из 400 задач с $|M| \in [80, 640]$, $|N| \in [120, 204480]$, $|T| \in [6, 160]$ из библиотеки SteinLib. Коэффициент аппроксимации, достигаемый с помощью k-кластерного метода оказался в среднем равным 1.05, что меньше, чем полученный в алгоритме Такахаши и алгоритме Зеликовского.

3. Сравнение методов генерации множества терминальных подмножеств на наборе из 50 сгенерированных автором задач с $|M| = 160$, $|N| \in [1100, 7700]$, $|T| \in [10, 50]$. Решения, получаемые с помощью наивного метода и метода «концентрических окружностей» на 15% и 40 % хуже в сравнении с решением, полученным обобщенным методом.

4. Сравнение точности решения приближенного алгоритма S^* с вышеприведенными алгоритмами, а также с алгоритмом k-кластерной оптимиза-

ции на сгенерированным автором наборе из 150 задач с $|M| \in [160, 640]$, $|N| \in [2371, 204156]$, $|T| \in [20, 120]$. Точность решений, получаемых методом S^*, оказалась лучшей среди участвовавших в сравнении методов.

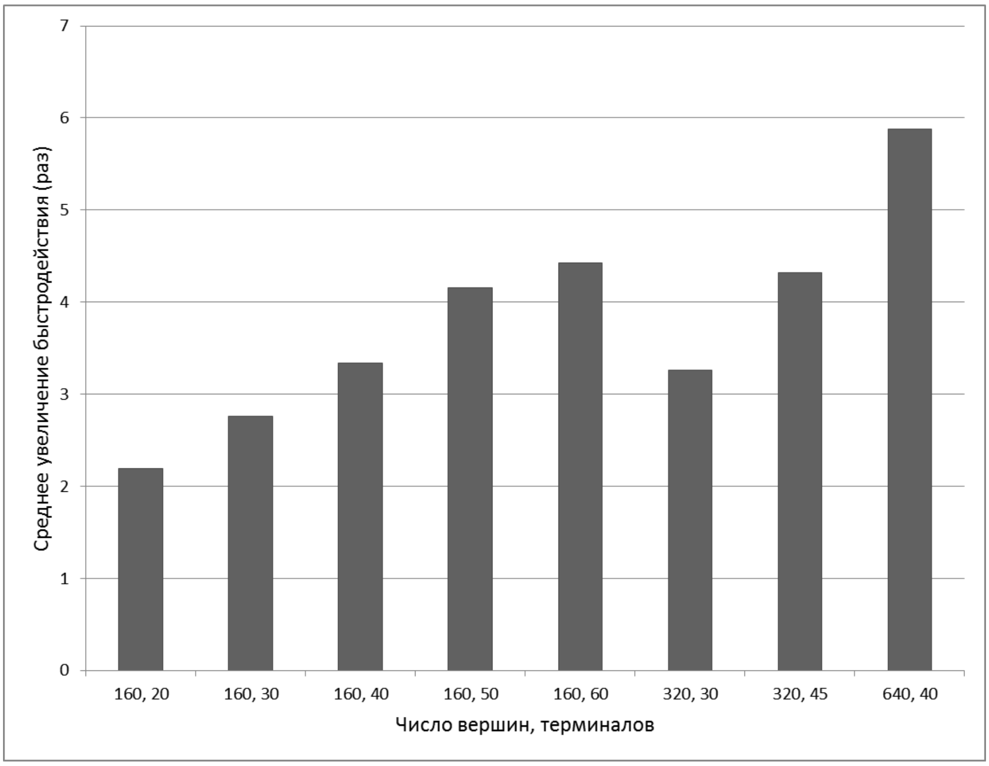

Рисунок 3 – Сравнение быстродействия параллельной и последовательной реализаций метода S^*

5. Проведено сравнение быстродействия последовательных и параллельных версий алгоритмов динамического программирования, k-кластерной оптимизации и приближенного алгоритма S^* на представленных ранее наборах задач. В результате исследований отмечено увеличение быстродействия до 6 раз на восьмиядерном процессоре. Диаграмма, отображающая отношение среднего увеличения быстродействия в зависимости от параметров задачи для метода S^*, представлена на рисунке 3.

В **заключении** по результатам исследования делаются выводы о применимости представленных методов на практике, преимуществах и недостатках в сравнении с другими методами, а также приводятся варианты возможных модификаций и улучшений данных методов с целью улучшения их быстродействия и коэффициента аппроксимации. Делается вывод о целесообразности дальнейшего исследования в данном направлении.

Публикации автора по теме диссертации

[1] *Ейбоженко Д. А.* k-кластерный метод для задачи Штейнера на графах. // Материалы российской конференции «Дискретная оптимизация и исследование операций», Новосибирск: Институт математики им. С. Л. Соболева, 2010, с. 159.

[2] *Романовский И. В., Ейбоженко Д. А.* Модификации метода динамического программирования в задачах Штейнера на ориентированных графах. // Компьютерные инструменты в образовании. Вып. 5, 2010, с. 22-28.

[3] *Ейбоженко Д. А.* Алгоритм k-кластерной оптимизации для задачи Штейнера на ориентированных графах. // Вестник СПбГУ. Сер. 10. Прикладная математика, информатика, процессы управления. Вып. 2, 2011, с. 29-39.

[4] *Ейбоженко Д. А.* Эвристический алгоритм S^* для задачи Штейнера на ориентированных евклидовых графах. // Материалы второй межвузовской научной конференции по проблемам информатики СПИСОК-2011, Санкт-Петербург: СПбГУ, 2011, с. 223-224.

[5] *Ейбоженко Д. А.* Задача Штейнера на ориентированных графах. // Материалы международной конференции «Математика, экономика, менеджмент: 100 лет со дня рождения Л. В. Канторовича» Санкт-Петербург: СПбГУ, 2012, с. 41.

Цитируемая литература

[6] *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. /Пер. с англ. Е.В.Левнера, М. А. Фрумкина, под ред. А. А. Фридмана, М.: Мир, 1982. 416 с.

[7] *Hart P.E., Nilsson N.J., Raphael B.* A formal basis for the heuristic determination of minimum cost paths. // IEEE transactions on Systems Science and Cybernetics, Vol. 4, 1968, P. 100–107.

[8] *SteinLib Testdata Library* — 2009. [Электронный ресурс]. URL: http://steinlib.zib.de/steinlib.php (дата обращения: 27.05.2009)

[9] *Takahashi H., Matsuyama A.* An Approximate Solution for the Steiner Problem In Graphs // Math. Japonica, 1980, Vol. 24, N. 6, P. 573-577.

[10] *Zelikovsky A.* 11/6-approximation algorithm for the Steiner problem on graphs. // Proc. Fourth Czechoslovakian Symposium on Combinatorics, Graphs, and Complexity. 1992, P. 351-354.

[11] *Zelikovsky A.* A Series of Approximation Algorithms for the Acyclic Directed Steiner Tree Problem. // Framework, 1997, P. 1-10.

[12] *H. Sutter* The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software. // Dr. Dobb's Journal, 30(3), March 2005.