УДК 004.421
DOI: 10.15587/1729-4061.2015.56247

РАЗРАБОТКА МЕТОДА МОНИТОРИНГА РАСПРЕДЕЛЕННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ОПРЕДЕЛЕНИЯ КРАТЧАЙШИХ ПУТЕЙ И КРАТЧАЙШИХ ГАМИЛЬТОНОВЫХ ЦИКЛОВ В ГРАФЕ

С. В. Листровой
Доктор технических наук, профессор
Кафедра специализированных компьютерных систем
Украинский государственный университет железнодорожного транспорта
пл. Фейербаха, 7, г. Харьков, Украина, 61050
E-mail: om1@ya.ru
С. В. Минухин
Кандидат технических наук, профессор
Кафедра информационных систем
Харьковский национальный экономический университет им. Семена Кузнеца
пр. Ленина, 9-а, г. Харьков, Украина, 61166
E-mail: minukhin.sv@gmail.com
Е. С. Листровая
Кандидат технических наук, доцент
Кафедра экономики и маркетинга
Национальный аэрокосмический университет им. Н. Е. Жуковского,
ул. Чкалова, 17, г. Харьков, Украина, 61070
E-mail: listravkina@gmail.com

Розглянуто метод моніторингу розподіленої обчислювальної системи, в основу якого покладено принцип мінімізації часу опитування віддаленими агентами об'єктів моніторингу, в які входять ресурси (вузли кластера), завдання та комунікаційні канали зв'язку. Показано, що час опитування на основі агентів віддаленого доступу визначається послідовністю їх запуску на ресурсах розподіленої системи

Ключові слова: розподілена обчислювальна система, моніторинг, агент, віддалений доступ, затримка, граф

Рассмотрен метод мониторинга распределенной вычислительной системы, в основу которого положен принцип минимизации времени опроса удаленными агентами объектов мониторинга, в которые входят ресурсы (узлы кластера), задания и коммуникационные каналы связи. Показано, что время опроса на основе агентов удаленного доступа определяется последовательностью их запуска на ресурсах распределенной системы

Ключевые слова: распределенная вычислительная система, мониторинг, агент, удаленный доступ, задержка, граф

1. Введение

Решение задач мониторинга распределенных вычислительных систем (РВС) является важной с точки зрения обеспечения требуемой при обработке потоков заданий производительности и пропускной способности. Следует отметить, что наиболее распространенные системы мониторинга Nagios [1], Icinga [2] используют программные расширения (агенты), устанавливаемые на объектах мониторинга для их удаленного запуска и реализующие различные сервисы. Для выполнения сервисов на узлах РВС требуется установить плагин NPRE (Nagios Remote Plugin Executor), инициализирующий работу программных агентов. Процедура работы вызываемого удаленного сервиса включает: инициализацию командой запуска, осуществляемой агентами NRPE на серверах и узлах РВС; запуск и выполнение сервиса; получение результатов работы сервиса; передачу полученных данных на управляющий узел (базу данных). Для комплексной оценки состояния программно-аппаратных средств, коммуникационных компонент и выполняемых заданий РВС следует использовать сервисы удаленного доступа, непосредственно связанные с решением задач планирования распределенных вычислений. К ним относятся [3, 4]: доступность и уровень загрузки узлов (в том числе многоядерных процессоров) кластеров; доступность и пропускная способность коммуникационных каналов (включая коммуникационные каналы кластеров); количество и доступность свободных узлов кластеров; состояние выполняемых заданий на узлах; доступность, текущая производительность и

загрузка узлов, используемых для хранения данных мониторинга. При наличии большого количества необходимых для качественного мониторинга объектов РВС резко увеличивается нагрузка на коммуникационную сеть, ограниченную ее пропускной способностью. Информация о состоянии объектов мониторинга является фоновой по отношению к основной – пользовательской и служебной (управляющей) информации, объемы которых непосредственно влияют на уровень обеспечения качества обслуживания пользователей – время обслуживания запросов и приложений, суммарное запаздывание, стоимость вычислений и т. д.

Для минимизации времени задержки на инициализацию и выполнение удаленных агентов предлагается метод оптимизации последовательности опроса узлов РВС, использующий представление РВС в виде неполносвязного графа, в котором требуется минимизировать задержку путем решения задач нахождения кратчайшего пути и кратчайшего гамильтонова цикла.

2. Анализ литературных данных и постановка проблемы

Модель реализации множеств указанных сервисов Rem_Serv можно представить в следующем виде:

$$\text{Rem\_Serv}: AN \times AU \times ANet \times AJ \times ADB \rightarrow \{State_1, State_2, State_3, State_4\},$$

где AN – множество сервисов определения доступности узлов; AU – множество сервисов определения загрузки (использования) узлов; AN – множество сервисов определения состояния коммуникационных каналов; AJ – множество сервисов определения состояния выполняемых заданий; ADB – множество сервисов определения состояния БД; $State_i$, $i=\overline{1,4}$ – множество состояний объекта мониторинга, определяемых {OK, WARNING, CRITICAL, UNKNOWN}. Учитывая, что мощности множеств рассмотренных сервисов для мониторинга составляют [1] $|AN|=2$, $|AU|=3$, $|ANet|=3$, $|AJ|=4$, $|ADB|=6$, а контролировать системному администратору и пользователям виртуальных организаций РВС необходимо, например, только два состояния объектов мониторинга WARNING и CRITICAL, минимальное количество сообщений, полученных от участвующих только в одном опросе удаленных агентов, составит 2^{18}. Учитывая, что размер одного сообщения составляет порядка 4 Кб, такой объем служебной информации приводит к значительной загрузке коммуникационных каналов, что приводит к необходимости минимизации временных задержек работы удаленных сервисов для мониторинга вычислительных кластеров и их узлов РВС, влияющих на пропускную способность каналов связи.

Для получения математической модели для оптимизации последовательности опроса объектов мониторинга РВС используем схему на рис. 1, показывающую технологию работы сервисов удаленного доступа.

На рис. 1 использованы следующие обозначения: R_i, $i=\overline{1,m}$, – ресурсы (вычислительные кластеры) РВС; S_i, $i=\overline{1,m}$, – серверы ресурсов; R_{ij}, $i=\overline{1,m}; j=\overline{1,N_m}$ – узлы кластеров; M_i – маршрутизаторы; УУ – управляющий узел, с которого инициируется запуск удаленных агентов.

Обозначим время передачи запроса на запуск удаленного агента как t_{ij}, определяемое по формуле:

$$t_{ij} = \frac{l_{ij}}{p_{ij}} x_{ij},$$

где l_{ij} – размер инициализирующего пакета, передаваемого между ресурсами R_i и R_j, Байт; p_{ij} – пропускная способность коммуникационного канала между ресурсами R_i и R_j (Байт/сек); $x_{ij}=1$, если ресурсы R_i и R_j соединены коммуникационным каналом, 0 – в противном случае.

Тогда временная задержка пакетов для запуска агентов на всех кластерах РВС составит:

$$T^1_{delay} = \sum_{i=1}^{m} \sum_{j=1}^{m} t_{ij}.$$

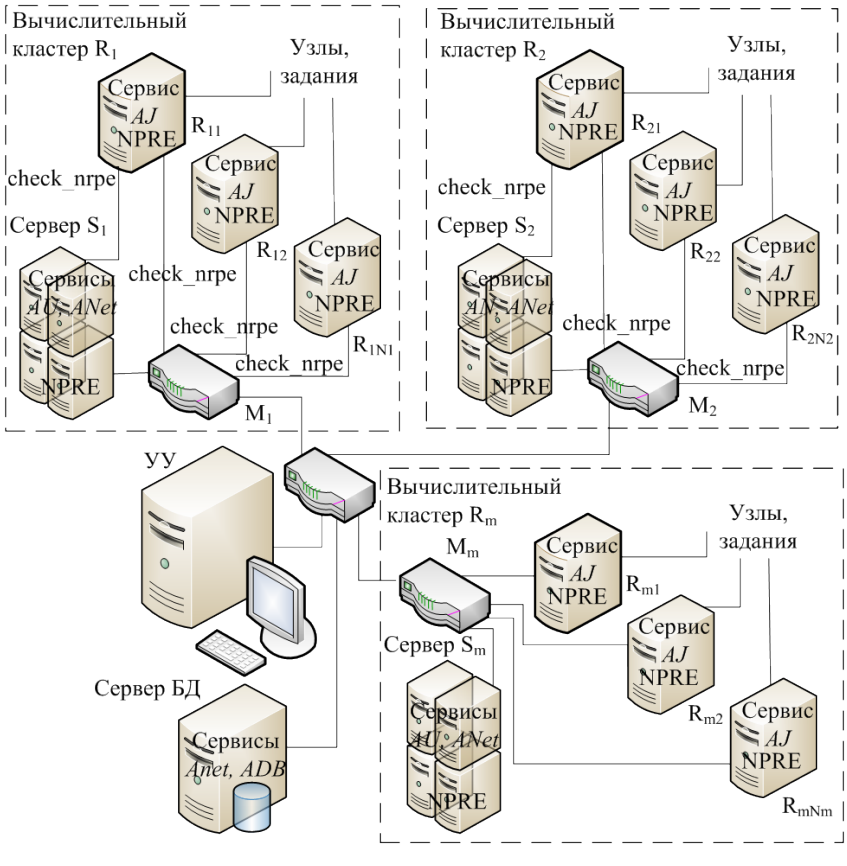

Рис. 1. Схема работы сервисов удаленного доступа в системе мониторинга РВС

После запуска команды check_nrpe на серверах S_i происходит инициализация и запуск удаленных сервисов на узлах вычислительных кластеров РВС. Временная задержка, определяющая запуск сервисов на узлах кластера, составит:

$$T_{delay\_Node} = \sum_{k=1}^{Rem\_Serv} \sum_{l=1}^{Nm} t_{kl}.$$

Временная задержка, определяющая запуск агентов на всех кластерах РВС определится по формуле:

$$T^2_{delay} = \sum_{i=1}^{m} \sum_{k=1}^{Rem\_Serv} \sum_{l=1}^{Nm} t_{ikl}.$$

Таким образом, общая временная задержка запуска удаленных агентов для мониторинга вычислительных кластеров РВС составит:

$$T_{delay} = \sum_{i=1}^{m} \sum_{j=1}^{m} t_{ij} + \sum_{i=1}^{m} \sum_{k=1}^{Rem\_Serv} \sum_{l=1}^{Nm} t_{ikl}.$$

Учитывая, что топология коммуникационных каналов РВС не является полносвязной, то есть не все ресурсы РВС соединены между собой, далее предлагается метод минимизации суммарной задержки: $T_{delay} \to \min$. При этом предполагается, что на интервалах планирования пропускная способность коммуникационных сетей между вычислительными кластерами РВС является постоянной.

Метод решения проблемы

Решение задачи минимизации суммарной временной задержки запуска удаленных агентов на узлах РВС предлагается решить в два этапа.

Этап 1. Построение для всех опрашиваемых узлов РВС дерева кратчайших путей к остальным узлам сети, которую будем представлять в виде графа G, вершинам которого будут соответствовать узлы сети РВС, ребрам – коммуникационные каналы между узлами сети РВС. При этом длины этих ребер будут определять временную задержку передачи данных по каналам связи, объединяющим выделенные узлы в единую сеть, длины путей в дереве кратчайших путей – минимальное время доставки данных (запросов) из некоторого узла i в узел j данной сети.

Этап 2. Рассматривается полный граф из опрашиваемых узлов РВС, в котором ребрам присвоены веса t_{ij}, определяющие минимальную задержку передачи запроса от некоторого узла i к узлу j рассматриваемой сети, равную длине кратчайшего пути в дереве кратчайших путей, построенном на этапе 1. Далее для данного графа определяется обход всех этих узлов за минимальное время, то есть определяется кратчайший гамильтонов цикл. Поскольку обе задачи должны решаться в реальном времени, для реализации предлагаемых алгоритмов на этапе 1 предлагается разработать параллельную реализацию алгоритма Дейкстры на основе рангового подхода [5], этапе 2 – решить задачу определения кратчайшего обхода всех улов сети на основе решения системы квадратичных уравнений.

При решении задачи определения путей возможны случаи, когда граф, к которому сведена исходная задача, имеет положительные и отрицательные веса ребер. В литературе выделяют следующие случаи: в графе все ребра имеют положительные веса; в графе имеются ребра с отрицательными весами, но нет циклов отрицательной длины; в графе имеются циклы отрицательной длины. При решении этих задач используются следующие методы: индексные (пометок), матричные и аппаратные. Все эти методы используются для решения двух типов задач: нахождения кратчайшего пути между заданной парой вершин и нахождения кратчайших путей между всеми парами вершин в графе. Для решения задачи поиска кратчайшего пути между парой вершин лучшими являются индексные и аппаратные методы, для решения задач определения кратчайших путей между всеми парами вершин в графе – матричные методы. Одним из наиболее эффективных алгоритмов определения кратчайших путей в графе с положительными весами ребер является алгоритм Дейкстры, использование которого в графах с отрицательными весами ребер приводит к экспоненциальному росту временной сложности [5]. В случае, когда в графе есть циклы отрицательной длины, задача относится к классу *NP*-полных задач. Даже перечисление всех алгоритмов определения кратчайших путей не является возможным, так как количество публикаций по методам определения кратчайших путей сейчас уже превысило 200 [6]. Задача поиска гамильтонова цикла минимального веса в полном графе с целыми неотрицательными весами рёбер известна как задача коммивояжёра (ЗК). К известным методам ее решения следует отнести точные алгоритмы, полученные в работах [7, 8]. В них для решения ЗК предложен алгоритм динамического программирования, имеющий временную сложность $O(2^n)$ и использующий память M (M – максимальный вес ребра). Лучший алгоритм с полиномиальной памятью с временной сложностью $O(4^n n^{\log n})$ получен в работах [9, 10]. В работе [11] показано, что ЗК может быть решена за время $O(2^{2n-t} n^{\log(n-t)})$ и памятью $O(2^t)$ для любого $t=n, n/2, n/4, \ldots$ В работах [12–15] предложен алгоритм для решения ЗК с временной сложностью $O(2^n)$ и памятью $O(M)$. Остается открытым вопрос о существовании алгоритма с временной сложностью $O(2^n)=2^n$ poly(n; log M) и полиномиальной памятью [16]. В работе [17] предложен алгоритм для решения ЗК в кубических графах со временем работы 1.260^n, для графов степени 4 со временем работы 1.890^n. Авторы работы [18] улучшили оценку для кубических графов до 1.51^n, в [19] разработан алгоритм для графов степени 4 со временем работы 1.733^n и экспоненциальной памятью. Актуальность использования ЗК в различных приложениях подтверждается разработкой в последнее время эвристических алгоритмов – муравьиного алгоритма [20, 21] и его модификаций [22, 23] с временной сложностью в худшем случае $O(n^3)$. В работе [24] предложен точный алгоритм решения ЗК с временной сложностью в интервале ($O(n^4)$, $O(n^3 2n)$). В работе [25] для оптимизации стоимости перевозок предложен метод решения транспортной задачи в вероятностной постановке, интерпретируемый для решения данной задачи как минимизация стоимости доставки сообщений в сетях с различной пропускной способностью.

Таким образом, проведенный анализ показывает, что получение точных решений ЗК связано с большими временными затратами. В связи с этим, целью этапа 2 является разработка метода решения ЗК с малой временной сложностью и малой погрешностью, позволяющего использовать его в системе мониторинга РВС в режиме реального времени.

3. Цели и задачи исследования

Целью исследования является решение задачи запуска удаленных агентов для системы мониторинга РВС путем оптимизации порядка их запуска на основе решения задач определения кратчайшего пути и кратчайшего гамильтонова цикла в графе, которые обеспечат повышение пропускной способности и качества обслуживания пользователей РВС.

Для достижения поставленной цели предлагается решить следующие задачи:

– разработать модель мониторинга РВС на основе сервисов удаленного доступа, обеспечивающую минимизацию времени их запуска;

– разработать эффективные методы и алгоритмы построения кратчайших путей, использующие представление коммуникационных каналов РВС в виде графа;

– разработать эффективные алгоритмы построения кратчайших гамильтоновых циклов в произвольном графе.

4. Методы решения задачи определения кратчайшего пути на основе рангового подхода

Для решения поставленной задачи воспользуемся представлением исходного графа G в виде симметричного дерева путей предложенного в работах [26, 27]. Пусть все возможные состояния некоторой системы определяется графом $G(V, E)$ с n вершинами, где вершины соответствуют возможным состояниям системы. Перейдем к пространству с $(n-1)^2$ состояниями. Для этого каждому из n состояний поставим в соответствие еще $(n-1)$ состояние, характеризующее способ достижения состояния из множества $\{1, 2, ..., n\}$. При этом в качестве добавляемых состояний определим ранг пути в графе $G(V, E)$: из вершины s графа $G(V, E)$ в произвольную вершину j можно попасть:

– путем ранга r=1, используя одно ребро;

– путем ранга r=2, используя 2 ребра и т. д.;

– путем ранга $r=n-1$, используя $n-1$ ребро.

Такое пространство состояний можно представить в виде стянутого дерева путей D (рис. 2).

Исходя из стянутого дерева путей, для произвольной вершины j множество путей, ведущих в эту вершину из некоторой вершины s, можно представить в следующем виде:

$$m_s(j) = m_{sj}^{r=1} \cup m_{sj}^{r=2} \cup .... \cup m_{sj}^{r=n-1}; j = \overline{(1, n-1)},$$

где $m_{sj}^r = \{\mu_{sj}^r\}$ подмножества путей из произвольной вершины s в некоторую вершину j графа $G(V, E)$, ранга r. Следует отметить, что дерево всех путей D может строиться и от конкретной вершины i графа: в этом случае вершина $s=i$ и i-я горизонтальная линейка исключаются в D. Таким образом, используя граф D и введя правила формирования путей следующего ранга можно из произвольной вершины s поэтапно строить пути $\{\mu_{sj}^r\}$ произвольного ранга вплоть до ранга $r=n-1$. В работе рассматривается взвешенный граф, ребрам которого присвоены веса l_{ij}, где i, j – смежные вершины, которые задаются матрицей длин $L=\{l_{ij}\}$. Если $\mu_{st}=(s, \nu_1, \nu_2, ..., \nu_k, t)$ – некоторый путь из вершины s в t, то его длина определяется как сумма длин ребер, образующих данный путь. Величину длины пути между вершинами i и j обозначим $d(i, j)$, величину кратчайшего пути между вершинами i и j – $P(i, j)$.

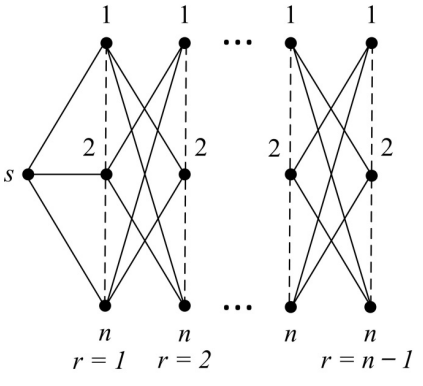

Рис. 2. Стянутое дерево всех путей D графа $G(V, E)$

4. 1. Метод решения задачи определения кратчайшего пути на основе рангового подхода

Как показано в работе [5], данная задача может интерпретироваться как задача линейного программирования на основе единичного потока минимальной стоимости. В терминах математического программирования она формулируется следующим образом: необходимо минимизировать

$$\sum_{i,j} l_{ij} X_{ij} \qquad (1)$$

при условии

$$\sum_i X_{ij} - \sum_i X_{ji} = \begin{cases} 1, & \text{если } i = s; \\ -1, & \text{если } i = t; \\ 0, & \text{если } i \neq s, t; \end{cases} \qquad (2)$$

$$X_{ij} \in \{0,1\}. \qquad (3)$$

Требуется определить поток $\{X_{ij}\}$, минимизирующий функционал (1). Если ребро (i, j) принадлежит кратчайшему пути, то X_{ij}=1, в противном случае – X_{ij}=0. Значения функционала определяет $P(s, t)$. Ограничения (2), (3) отображают условия непрерывности пути. Задачу определения кратчайшего пути предлагается решить с помощью рангового метода, использующего переход от графа G к стянутому дереву всех путей D.

Рассмотрим некоторый произвольный граф $G(V, E)$ с множеством вершин V и множеством ребер E, мощность которого не превышает значения $n(n-1)/2$. Если в графе выделить свободные вершины s и t, то множество всех путей $\{\mu_{st}\}$ между этой парой вершин можно представить в виде объединения подмножеств

$$M_{st} = M^{r-1}{}_{st} \cup M^{r-2}{}_{st} \ldots \cup M_{st}{}^{r-n-1}, \quad (4)$$

где $M^{r}{}_{st}$ – множества путей между вершинами s и t ранга r.

Пути ранга r=n–1 – пути с максимально возможным рангом – являются гамильтоновыми путями, проходящими через все вершины графа G. Таким образом, задача определения кратчайшего пути между заданной парой вершин s и t сформулирована как задача определения пути минимальной длины некоторого ранга r в множестве M_{st}:

$$\min M_{st} = \min \left\{ M^{r=1}{}_{st},\ M^{r=2}{}_{st}, \ldots,\ M^{r=n-1}{}_{st} \right\}. \quad (5)$$

Из (5) следует, что если в каждом из подмножеств M^{r}_{st}, $r = (1, n-1)$ определить кратчайший путь ранга r и обозначить полученное множество как $M^{*r}_{st}, r = (1, n-1)$, тогда задача определения кратчайшего пути сводится к задаче определения кратчайшего пути в множестве M^{*r}_{st}:

$$\min \left\{ M^{*r=1}{}_{st},\ M^{*r=2}{}_{st}, \ldots,\ M^{*r=n-1}{}_{st} \right\}. \quad (6)$$

Пусть некоторая процедура ***A*** позволяет строить кратчайшие пути между вершинами s и t произвольного ранга. Тогда в соответствии с (5), (6) требуется определить кратчайший путь между вершинами s и t. В случае определения кратчайших путей между вершиной s и всеми остальными вершинами графа G задача формулируется следующим образом: необходимо определить $\min\ \mu^{r}{}_{si} = \min \left\{ \mu^{r-1}{}_{si}, \mu^{r-2}{}_{si}, \ldots, \mu^{r-n-1}{}_{si} \right\}$, где i=(1,n), r=(1,n–1).

Переход от дерева всех путей графа к стянутому дереву всех путей D соответствует разбиению множества путей графа G на подмножества M^{*r}_{si}, при этом вершины графа D рассматриваются как множества путей M^{*r}_{si}.

4. 2. Алгоритмы определения кратчайших путей на основе рангового подхода

В последнее время для решения задачи нахождения кратчайшего пути в графе используются многопроцессорные вычислительные структуры [4]. Построение кратчайших путей основывается на следующей процедуре ***A***. Из вершины *s* строятся все возможные пути второго ранга ко всем вершинам графа *G*. Полученное множество путей M^2 разбивается на подмножества $M^2{}_i$ таким образом, что каждый из них заканчивался на i. В каждом подмножестве выделяется путь минимальной длины. В результате этого получается $M^2{}_{min}$ – множество путей второго ранга, в котором каждый путь является кратчайшим путем второго ранга от вершины s к соответствующей вершине t. Используя множество $M^2{}_{min}$, строятся все пути ранга 3 ко всем вершинам графа. Далее полученное множество M^3 разбивается на подмножества $M^3{}_i$, в каждом из которых определяется путь минимальной длины.

Выполнение процедуры ***A*** повторяется до получения путей максимально возможного ранга r=n–1. Очевидно, что после (n–1)-го шага повторения процедуры ***A*** будут получены все возможные кандидаты на кратчайшие пути между вершинами *s* и рангов r=1, 2, ..., n–1. Выбирая среди них для каждой вершины пути минимальной длины, определяем кратчайшие пути между вершиной *s* и остальными вершинами графа.

Таким образом, алгоритм определения кратчайших путей между заданной вершиной *s* и остальными вершинами графа реализуется следующим образом.

Алгоритм *A1*

Шаг 1. Процедура ***A*** повторяется (n–1) раз, в результате чего получается множество минимальных путей всех возможных рангов ко всем вершинам.

Шаг 2. В каждом из полученных на шаге 1 множеств путей выбираются минимальные пути к вершине i, которые и являются искомыми кратчайшими путями от вершины s к остальным вершинам графа G.

Утверждение 1. Алгоритм ***A1*** позволяет определить кратчайшие пути между заданной вершиной *s* и остальными вершинами графа G.

Доказательство. Пусть процедура построила все пути ранга 1 от вершины s к остальным вершинам графа и на их базе – все возможные пути ранга r=2, то есть сформированы все подмножества $M_i{}^{r=2}$. Выберем в каждом подмножестве кратчайший путь $\overset{*}{\mu}{}_i^{r=2}$. Ясно, что пути $\overset{*}{\mu}{}_i^{r=2}$ являются кратчайшими путями ранга r=2 к вершинам i, поскольку на этом шаге каждое из подмножеств содержит все возможные пути ранга r=2. Сформируем на базе путей $\{\overset{*}{\mu}{}_{si}^{r=2}\}$ в соответствии с алгоритмом ***A1*** все возможные пути ранга r=3, то есть подмножества $M_i{}^{r=3}$, и выделим в каждом пути кратчайший путь $\overset{*}{\mu}{}_{si}^{r=3}$. Докажем, что полученные таким образом пути $\overset{*}{\mu}{}_{si}^{r=3}$ являются кратчайшими путями ранга r=3 из вершины s в вершины $i = (\overline{1, n})$, i≠s. Для этого предположим, что к некоторой вершине i=j есть путь $\overset{**}{\mu}{}_{si}^{r=3}$, длина которого меньше, чем $\overset{*}{\mu}{}_{si}^{r=3}$. Поскольку длины путей ранга r=3 в соответствии с предложенной процедурой определяются как

$$d_{si}{}^{r=3} = \min \left\{ d_{si}{}^{r=2} + l_{ij} \right\}, i = (\overline{1,n}),\ j = (\overline{1,n}), i \neq j,$$

это предположение может быть выполнено, если в графе есть более короткий путь $\overset{**}{\mu}{}_{si}^{r=2}$ ранга r=2, чем пути $\overset{*}{\mu}{}_i^{r=2}$. Это невозможно, так как они в графе G являются кратчайшими путями ранга r=2.

Предположим, что алгоритм ***A1*** построил кратчайшие пути $\overset{*}{\mu}{}_{si}^{r=k}$ ранга r=k ко всем вершинам графа. Далее в соответствии с ***A1*** построим на их основе все возможные пути ранга r=k+1, то есть сформируем множества $M_i{}^{r=k+1}$ и в каждом множестве $M_i{}^{r=k+1}, i = (\overline{1,n})$ выделим кратчайшие пути $\overset{*}{\mu}{}_{si}^{r=k+1}$. Докажем, что они являются кратчайшими путями ранга r=k+1. Для этого повторим предыдущее рассуждение: предположим, что есть путь $\overset{**}{\mu}{}_{si}^{r=k+1}$ короче, чем путь $\overset{*}{\mu}{}_{si}^{r=k+1}$, что противоречит первоначальному предположению о том, что построенные алгоритмом ***A1*** пути $\overset{*}{\mu}{}_{si}^{r=k}$ являются кратчайшими путями ранга r=k и, следовательно, пути $\overset{*}{\mu}{}_{si}^{r=k+1}$ действительно являются кратчайшими путями ранга r=k+1.

Таким образом доказано, что алгоритм ***A1*** при r=3 определяет кратчайшие пути ранга r=3 и на их

основе алгоритм построил кратчайшие пути ранга r=k+1, и, следовательно, в соответствии с методом полной математической индукции алгоритм ***A1*** позволяет определять кратчайшие пути произвольного ранга.

После (n–1)-го шага во всех множествах M_i^r построены пути μ_{si}^r и тогда в соответствии с соотношением (6) определены кратчайшие пути между вершиной s и всеми остальными вершинами графа, что и требовалось доказать.

В процессе работы алгоритма ***A1*** для произвольных графов возможна ситуация, когда множества $\{M^r_i\}$ в процессе работы алгоритма могут оказаться пустыми. Это возможно в двух случаях: когда пути текущего ранга r к вершине i в графе G отсутствуют, и когда путь существует, но алгоритм ***A1*** не смог его построить – такую ситуацию назовем тупиковой. Покажем, что возникновение тупиковых ситуаций не нарушает оптимальности работы алгоритма ***A1***. Для этого рассмотрим механизм возникновения тупиков. Тупиковая ситуация может иметь место в случае, если $M_t^{r=q+1}=\varnothing$. Согласно алгоритму ***A1*** подмножество формируемых путей M^r_i может быть пустым, если вершина t вошла во все кратчайшие пути ранга r=q к вершинам $i=\overline{(1,n)}$. Сформулируем следующее утверждение.

Утверждение 2. Если в процессе работы алгоритма ***A1*** построены кратчайшие пути следующих рангов $\overset{*}{\mu}_{st}^{r=1}$, $\overset{*}{\mu}_{st}^{r=2}$, ..., $\overset{*}{\mu}_{st}^{r=q}$ и подмножество путей $M_{st}^{r=q+1}=\varnothing$, то кратчайшим путем между вершинами *s* и *t* является путь

$$\overset{*}{\mu}_{st}=\min\{\overset{*}{\mu}_{st}^{r=1}, \overset{*}{\mu}_{st}^{r=2}, \ldots, \overset{*}{\mu}_{st}^{r=q}\}.$$

Доказательство. Так как подмножество $M_{st}^{r=q+1}=\varnothing$, то вершина t вошла в пути $\overset{*}{\mu}_{si}^{r=q}$, представляющие собой кратчайшие пути ранга r=q из вершины s ко всем остальным вершинам графа $i=\overline{(1,n)}$. Предположим, что есть путь $\overset{**}{\mu}_{st}^{k=q+1}$ короче пути $\overset{*}{\mu}_{st}^{r=q+1}$. Это возможно, если существуют пути $\overset{**}{\mu}_{st}^{r=q}$ более короткие, чем пути $\overset{*}{\mu}_{st}^{r=q}$ ранга r=q, не проходящие через вершину t. Однако это противоречит предположению о том, что пути $\overset{*}{\mu}_{si}^{r=q}$, проходящие через вершину t, являются кратчайшими путями ранга r=q к вершинам i, и, следовательно, предположение о наличии более короткого пути $\overset{**}{\mu}_{st}^{k=q+1}$, чем путь $\overset{*}{\mu}_{st}$, неверно, что и требовалось доказать.

Справедливость утверждения 2 вытекает из свойства, присущего всем кратчайшим путям, – любой отрезок кратчайшего пути также есть кратчайший путь между соединенными им вершинами. Из утверждения 2 следует важный вывод: если веса в матрице длин $L=\|l_{ij}\|$ – положительные величины, то возникновение тупиковых ситуаций не влияет на оптимальность работы алгоритма ***A1***, поскольку сами кратчайшие пути являются источником возникновения тупиков. Значительно сложнее решить задачу, когда в матрице весов (длин) содержатся веса отрицательной величины и есть циклы отрицательной длины. В анализируемом графе в этом случае оптимальность работы алгоритма ***A1*** может нарушиться. В этом случае переходим к другому классу известных задач дискретной оптимизации – NP-полным задачам, возможность решения которых на графе G требует дополнительного исследования.

Фактически предложенный алгоритм ***A1*** отображает основное функциональное уравнение динамического программирования для данной задачи:

$$d^r_{min}[i]=\min_r\{d^{r-1}_{min}[i]+l_{ij}\}, \quad (7)$$

где $d^r_{min}[i]$ – минимальная длина (r–1)-го ранга от вершины s к вершине i.

Так как предложенный алгоритм разворачивает процесс динамического программирования от первого шагу к последнему, то нахождение оптимального решения возможно после нахождения всех условных локальных решений и соответствующих значений локальных минимумов. При этом величины кратчайших путей от корневой вершины s ко всем остальным вершинам i определяются следующим соотношением:

$$d_{min}[i]=\min_r\{d^{r-1}_{min}[i]\}. \quad (8)$$

Использование рассмотренного алгоритма в соответствии с принципом оптимальности Беллмана [7] позволяет определить кратчайшие пути из некоторой, произвольной вершины s графа G ко всем остальным вершинам этого графа.

Отличительной особенностью алгоритма ***A1*** является то, что на каждом шаге работы сравниваются величины путей одного и того же ранга. Это, в отличие от алгоритмов, основанных на методе расстановок меток, позволяет реализовать эффективное распараллеливание алгоритма определения кратчайших путей в графе.

4. 3. Алгоритм *A2* (параллельная реализация алгоритма *A1*)

Поскольку формирование множеств путей *M* на каждом ранге можно производить одновременно, в соответствии с соотношениями (7) и (8) для окончания работы алгоритма ***A1*** при его параллельной реализации необходимо (n–1) раз выполнить операции сложения и сравнения и одну операцию выбора минимального элемента в массиве, определяемую соотношением (7). После построения множества путей M^{r+1} в множествах предыдущего ранга M^r необходимо сохранять только экстремальные значения d^r_{min}, следовательно, объем памяти, необходимой для реализации алгоритма ***A2***, определяется множеством путей, формируемых на данном ранге. Количество таких путей равно n^2-3n+2, так как в каждом подмножестве M^r_i содержится (n–2) пути ранга r и число таких подмножеств равно (n–1). Временная сложность алгоритма ***A2*** не превышает O(n), требуемый объем памяти – $O(n^2)$. Временная сложность алгоритма ***A2*** в последовательном варианте его реализации не превысит $O(n^3)$. Следует отметить, что быстродействие алгоритма ***A2*** может быть повышено, если в процессе выполнения уравнения (7) в каждом из формирующих множеств M^r ввести дополнительную проверку:

$$d^r_{min}[i]\le d^{r+1}_{min}[i]. \quad (9)$$

Если на ранге *r* выполняются соотношения (9) для всех подмножеств, то дальше использовать его не следует, так как все полученные на следующих шагах величины путей более высокого ранга будут заведомо

длиннее, чем кратчайшие пути, полученные до ранга r+1. В худшем случае, когда кратчайшим путем есть гамильтонов путь, временная сложность равна $O(n)$. Последняя ситуация встречается довольно редко, и введение проверки неравенства (9) позволит значительно уменьшить количество шагов в процедуре ***A2***. Если определяется кратчайший путь в графе ранга r, то он будет найден на (r+1) шаге. Необходимая память для реализации предложенного алгоритма пропорциональна $O(n^2)$.

Практическая реализация ***A2*** осложняется тем, что для построения на их основе параллельных архитектур вычислительных систем, в процессорных элементах необходимо иметь (n–1) группу регистров для промежуточного хранения путей и при этом, если сеть задается неполным графом, большая часть регистров не используется, что существенным образом ограничивает размерность решаемых задач. Предположим, что известны самые короткие пути $\overset{*r}{\mu}_{sj}$ в каждом подмножестве m_{sj}^{r}. Тогда самый короткий путь от вершины s к вершине j равен

$$\overset{**}{\mu}_{sj} = \min_{r} \{\overset{*r=1}{\mu}_{sj}, \overset{*r=2}{\mu}_{sj}, ..., \overset{*r=n-1}{\mu}_{sj}\}.$$

Обозначим длину пути $\overset{*r}{\mu}_{sj}$ как $d(\overset{*r}{\mu}_{sj})$ и назовем ее локальным экстремумом ранга r к вершине j, $d(\overset{**r}{\mu}_{sj})$ – глобальным экстремумом ранга r к вершине j, и длину $d(\overset{*r}{\mu}_{sj}) = \min_{j} \{\overset{*r}{\mu}_{sj}\}$ – глобальным экстремумом на ярусе. Тогда задача определения КП может быть сформулирована как задача определения глобальных экстремумов в пространстве, определяемом стянутым деревом путей. Используя свойство, что любой отрезок КП является также КП между вершинами, определяющими данный отрезок, построим следующую процедуру определения КП между заданной вершиной s и всеми остальными вершинами произвольного графа G(V, E).

4. 4. Алгоритм *A3*

Шаг 1. Из вершины s строим пути $m_{sj}^{r=1}$, $j = \overline{(1, n-1)}$, выделяем в них локальные экстремумы и находим среди них глобальный экстремум $d(\overset{*r=1}{\mu}_{sj})$ на ярусе. Вершину j исключаем из дальнейшего анализа.

Шаг 2. На основе пути, соответствующего текущему глобальному экстремуму на ярусе, строим все возможные пути множества $m_{sj}^{r:=r+1}$ и подтягиваем в них пути, соответствующие локальным экстремумам предыдущего ранга. После этого определяем локальные экстремумы во множествах $m_{sj}^{r:=r+1}$ и среди них выделяем глобальный экстремум на ярусе (r+1), соответствующий некоторой вершине j. Вершину j исключаем из дальнейшего анализа.

Шаг 3. Проверяем, имеет ли место равенство r=n–1; если нет, то переходим к выполнению шага 2, иначе алгоритм заканчивает работу.

Поясним работу данной процедуры на примере. Пусть требуется определить КП в графе G, приведенном на рис. 3, от вершины 1 ко всем остальным вершинам графа. Процесс работы процедуры на всех ее шагах работы показан в табл. 1, в которой каждый квадрат соответствует вершине в стянутом дереве путей. В табл. 1 показаны пути, построенные процедурой на всех ярусах, причем символом * помечены пути, соответствующие локальным экстремумам, а символом ** – пути, соответствующие глобальным экстремумам на ярусах (в скобках возле каждого пути указана его длина).

Таблица 1

Процесс построения дерева кратчайших путей

12(1)** 2	---------- 2	---------- 2	---------- 2	12(1)
13(5)* 3	123(2)** 13(5) 3	---------- 3	---------- 3	123(2)
------- 4	124(4)* 4	1234(4) 124(4)** 4	---------- 4	124(4)
-------- 5	---------- 5	1235(6)* 5	1245(5)** 1235(6) 5	1245(5)
r=1	r=2	r=3	r=4	–

Следует отметить, что коэффициент ускорения для алгоритма ***A2*** составляет $n^2/\log_2 n$, а для алгоритма – ***A3*** $n/\log_2 n$, однако в алгоритме ***A2*** максимальное число путей, формируемых во множествах m_{sj}^{r} на произвольных рангах, не превышает (n–1), а для алгоритма ***A3*** – двух путей. Но при этом алгоритм ***A2*** всегда должен строить пути ранга r=n–1, а алгоритм ***A3*** – до некоторого значения r_k, определяемого неравенством (9). Результаты экспериментальных исследований средних значений рангов кратчайших путей в произвольных графах с весовыми характеристиками, изменяющимися по равномерному закону, для графов с числом вершин от 25 до 70 с плотностями ребер в диапазоне от 0,125 до 1 позволили получить следующие значения r_k (табл. 2).

Таблица 2

Значения среднего ранга кратчайших путей

n	25	30	35	40	45	50	60	70
r_k	7	8	8	8	9	9	9	10
max r	11	11	11	12	12	12	14	12

Таким образом, коэффициент выигрыша алгоритма ***A2*** по отношению к алгоритму ***A3*** определяется величиной n/r_k.

Для реализации первого этапа предлагаемого метода решения поставленной задачи целесообразно использовать предложенную параллельную реализацию алгоритма Форда (алгоритм ***A2***) и алгоритм Дейкстры (алгоритм ***A3***) на основе рангового подхода.

5. Разработка метода решения задачи нахождения кратчайшего гамильтонова цикла

Для реализации этапа 2 решения поставленной задачи рассматриваемого подхода необходимо разработать эффективный алгоритм определения кратчайшего гамильтонова цикла (КГЦ).

5. 1. Формализация задачи КГЦ

Рассмотрим произвольный полный неориентированный граф G(V, E) с n вершинами и m ребрами,

пронумерованными от 1 до m. Каждой вершине i графа инцидентно (n–1) ребро. Поставим в соответствие каждому r-му ребру переменную X_r, которая принимает значение, равное 1, если ребро входит в гамильтонов цикл, и равное 0 – в противном случае. Множество всех таких переменных обозначим W, мощность этого множества равна $\frac{n(n-1)}{2}$. Тогда сумма переменных $\sum_{r=1}^{n-1} X_r$, входящих в гамильтонов цикл для каждой вершины i, должна удовлетворять условию $\sum_{r=1}^{n-1} X_r=2$. Следовательно, система булевых уравнений

$$\sum_{r=1}^{n-1} X_r=2,\ (i=1),\ \sum_{r=1}^{n-1} X_r=2\ (i=2),\ ...,\ \sum_{r=1}^{n-1} X_r=2(i=n) \qquad (10)$$

описывает все возможные циклы в графе G(V, E), то есть для описания произвольного цикла в графе G(V, E) необходимо выделить такое подмножество переменных $\{X_r\}$, чтобы в каждом уравнении в (10) присутствовало ровно по две переменные. Если каждое ребро r и, соответственно, каждая переменная X_r имеет весовую характеристику P_r то, если в каждом i-м уравнении системы выделить подмножество таких переменных $\{X_r\}$, что в каждом уравнении (10) присутствует по две переменных и сумма весов переменных – минимальна, то данное подмножество образует цикл из ребер графа, и он будет иметь минимальную длину. Например, граф G, приведенный на рис. 3 с матрицей весов ***P*** и нумерацией ребер, приведенной на рис. 4, можно представить в виде следующей системы булевых уравнений:

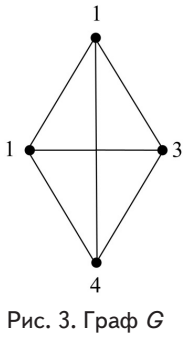

Рис. 3. Граф *G*

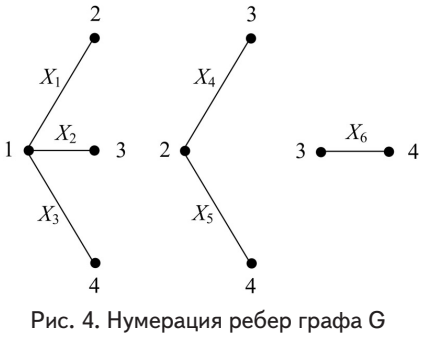

Рис. 4. Нумерация ребер графа G

$$\begin{aligned}
&X_1+X_2+X_3=2;\ \ i=1,\\
&X_1+X_4+X_5=2;\ \ i=2,\\
&X_2+X_4+X_6=2;\ \ i=3,\\
&X_3+X_5+X_6=2;\ \ i=4,
\end{aligned} \qquad (11)$$

и при этом каждое ребро в (11) характеризуется весовой характеристикой P_r (табл. 3).

Таблица 3

Описание графа на рис. 4

X_r	X_1	X_2	X_3	X_4	X_5	X_6
Концевые вершины ребра, (*i, j*)	(1,2)	(1,3)	(1,4)	(2,3)	(2,4)	(3,4)
Вес ребра, P_r	0	2	1	1	2	0

Для того чтобы произвольное уравнение в (10) выполнялось, в нем должны присутствовать только две переменные, равные 1, а остальные должны быть равны нулю, поскольку в гамильтонов цикл входит только по два ребра, инцидентных каждой вершине i. Обозначим множество всех возможных сочетаний пар переменных X_lX_k в каждом i-м уравнении через Ω_i. Тогда мощность этого множества будет равна

$$\alpha_i=\alpha=\frac{(n-1)(n-2)}{2},$$

так как для полного графа значения α_i во всех уравнениях будут одинаковы. Тогда уравнения (10) можно представить в следующем виде:

$$\sum_{x_lx_k\in\Omega_{i=1}} X_lX_k=1,\quad i=1;$$

$$\sum_{x_lx_k\in\Omega_{i=2}} X_lX_k=1,\quad i=2;\ ...,\ \sum_{x_lx_k\in\Omega_{i=n}} X_lX_k=1,\quad i=n. \qquad (12)$$

В соответствии с (12) уравнения (11) можно представить в виде:

$$\begin{aligned}
&X_1X_2+X_1X_3+X_2X_3=1;\ i=1,\\
&X_1X_4+X_1X_5+X_4X_5=1;\ i=2,\\
&X_2X_4+X_2X_6+X_4X_6=1;\ i=3,\\
&X_3X_5+X_3X_6+X_5X_6=1;\ i=4.
\end{aligned} \qquad (13)$$

Пары X_lX_k в выражениях (12) и (13) будем характеризовать суммарной весовой характеристикой P_{lk}, равной сумме весовых характеристик P_l и P_k переменных X_l и X_k соответственно. Особенностью решения системы (12) является то, что подмножества переменных $\{X_r\}$, удовлетворяющих системе квадратичных булевых уравнений (12), должны иметь в каждом из уравнений этой системы только по две переменных, равных 1, чтобы этот набор определял гамильтонов цикл в графе G(V, E). Таким образом, задачу определения кратчайшего гамильтонова цикла в графе G(V, E) можно рассматривать как задачу определения подмножества переменных $\{X_r\}$, удовлетворяющих системе квадратичных булевых уравнений (12) и имеющего минимальную суммарную весовую характеристику.

Каждое i-е уравнение в (12) будем характеризовать величиной $\Delta_i=P_i^{max}-P_i^{min}$, определяемой как разность между максимальной весовой характеристикой слагаемого и минимальной весовой характеристикой слагаемого в *i*-м уравнении и весовой характеристикой

$$\gamma_i=\sum_{i=1}^{\alpha_i} P_i-P_i^{min}.$$

Для уравнений (13) в соответствии с табл. 3 эти характеристики будут иметь вид

$$i=1)\ \overset{2}{X_1}X_2+\overset{1}{X_1}X_3+\overset{3}{X_2}X_3=1;\quad \Delta_1=2;\gamma_1=5,$$

$$i=2)\ \overset{1}{X_1}X_4+\overset{2}{X_1}X_5+\overset{3}{X_4}X_5=1;\quad \Delta_2=2;\gamma_2=5,$$

$$i=3)\ \overset{3}{X_2}X_4+\overset{2}{X_2}X_6+\overset{1}{X_4}X_6=1;\quad \Delta_3=2;\gamma_3=5,$$

$$i=4)\ \overset{3}{X_3}X_5+\overset{1}{X_3}X_6+\overset{2}{X_5}X_6=1;\quad \Delta_4=2;\gamma_4=5. \qquad (14)$$

В уравнениях (14) над каждой парой переменных X_lX_k указана сумма весовых характеристик P_l и P_k переменных X_l и X_k соответственно. Выделим в системе (14) базовое уравнение, определяемое на основе максимального значения параметра Δ_i. В случае, если максимальные значения окажутся равными, то в качестве базового уравнения выбирается то, которое имеет большее значение параметра γ_i; в случае, если Δ_i и γ_i окажутся равными, то в качестве базового уравнения выбирается любое из них. Так как в (14) в уравнениях параметры Δ_i и γ_i равны, то любое из них может быть выбрано в качестве базового уравнения. Поскольку каждая пара переменных соответствует объединению пары ребер $i-p-j$, то далее вместо уравнений (12) рассматриваются последовательности пар ребер, соответствующих парам переменных X_lX_k в (12). Таким образом, вместо уравнений (14) рассматриваются последовательности ребер следующего вида:

$$i=1)\ \overset{0}{2}-1-\overset{2}{3};\quad \overset{0}{2}-1-\overset{1}{4};\quad \overset{2}{3}-1-\overset{1}{4};$$

$$i=2)\ \overset{0}{1}-2-\overset{1}{3};\quad \overset{0}{1}-2-\overset{2}{4};\quad \overset{1}{3}-2-\overset{2}{4};$$

$$i=3)\ \overset{2}{1}-3-\overset{1}{2};\quad \overset{2}{1}-3-\overset{0}{4};\quad \overset{1}{2}-3-\overset{0}{4};$$

$$i=4)\ \overset{1}{1}-4-\overset{2}{2};\quad \overset{1}{1}-4-\overset{0}{3};\quad \overset{2}{2}-4-\overset{0}{3}. \qquad (15)$$

Над ребрами показаны их весовые характеристики, при этом каждая последовательность имеет те же характеристики Δ_i и γ_i, что и соответствующие уравнения в (14). Для решения системы уравнений (12) в последовательностях (15) необходимо указать в каждой последовательности одну пару ребер таким образом, чтобы они образовали цикл минимальной длины.

5. 2. Процедура формирования одного цикла *B*

Построение цикла начинается с последовательности, соответствующей базовому уравнению, выбирая в ней пару ребер с минимальным суммарным весом. Положим, что, например, пусть это будет пара ребер $i-s-j$ с суммарным весом $(P_{is}+P_{sj})$. Далее формирование цикла может осуществляться на основе объединения с другой парой ребер: либо по ребру $i-s$, либо по ребру $s-j$. Если осуществляется формирование цикла по ребру $s-j$, то осуществляется переход на последовательность $i=j$ и просматриваются в этой последовательности все пары ребер, содержащие ребро $s-j$, и среди них выбирается пара ребер с минимальным суммарным весом. Пусть это будет пара ребер $s-j-t$. Далее осуществляется объединение с другой парой по ребру $j-t$, для чего осуществляется переход на последовательность $i=t$ и в этой последовательности просматривается все пары ребер, содержащие ребро $j-t$, и среди них выбирается пара ребер с минимальным суммарным весом – пара $j-t-q$. На данный момент сформирована цепочка $i-s-j-t-q$, и далее аналогично, переходя на последовательность $i=q$, снова просматриваются в этой последовательности все пары ребер, содержащие ребро $t-q$ и среди них выбирается пара ребер с минимальным суммарным весом. Процедура продолжает работу до тех пор, пока не просмотрится $(n-1)$ последовательность, соответствующая системе уравнений (12). Последнюю последовательность можно не посещать, поскольку в ней гарантировано содержится ребро, замыкающее данную цепочку, так как в нем находятся все возможные сочетания ребер по два, инцидентных вершине, соответствующей номеру данной последовательности. При этом в процессе формирования цепочки на этапах выбора пары ребер, с помощью которой можно продлить цепочку, из анализа будут исключаться пары ребер вершины, которые уже присутствуют в цепочке или образуют ранний цикл. Таким образом, если сформирована цепочка $i-s-j-t-q$ и на основе ребра $t-q$ в q-й последовательности ищутся подмножества ребер, содержащих ребро $t-q$, то исключаются из анализа пары вида $t-q-s$, $t-q-j$, так как вершины s, j уже содержатся в сформированной цепочке, а также исключается пара вида $t-q-i$, если число ребер в цепочке меньше n, поскольку она образует ранний цикл.

На основе процедуры ***B*** предлагается следующая процедура ***A4*** определения кратчайшего гамильтонова цикла на основе решения системы уравнений (12), представленной последовательностями наборов ребер вида (14), соответствующих уравнениям системы (12).

5. 3. Процедура *A4*

Шаг. 1. Для последовательностей пар ребер, определяемых уравнениями (12), рассчитываются характеристики Δ_i и γ_i; выбирается последовательность, определяемая как базовое уравнение.

Шаг 2. В базовой последовательности выделяются пары ребер с минимальным весом (их, в общем случае, может быть λ), которые включаются в множество U_1. Таким образом, сформированы ребра первого яруса.

Шаг 3. Путем применения процедуры ***B*** выбирается пара ребер $i-s-j$ из U_1, $\lambda:=\lambda-1$, далее подсоединяется следующее ребро, или, если при выборе минимальной пары число минимальных одинаковых пар равно h, h ребер. Таким образом, сформированы ребра следующего яруса.

Шаг 4. На основе всех ребер, сформированных на предыдущем ярусе, используя процедуру ***B*** формируются ребра следующего яруса.

Шаг 5. Проверяется, равно ли число построенных ярусов $n-1$; если нет, то осуществляется переход к выполнению шага 4, иначе выполняется следую-щий шаг.

Шаг 6. Из всех построенных циклов выбирается цикл минимальной длины и заносится в множество U_2.

Шаг 7. Проверяется равенство $\lambda=0$, если оно не выполняется, то осуществляется переход к выполнению шага 3, иначе выполняется следующий шаг.

Шаг 8. Из множества U_2 выбирается самый короткий цикл.

На этом процедура заканчивает работу, так как кратчайший гамильтонов цикл найден.

Рассмотрим пример работы процедуры ***A4*** для 6-вершинного полного графа, заданного набором ребер $\{X_r\}$ с весовыми характеристиками $\{P_r\}$ (табл. 4).

Таблица 4

Описание 6-вершинного полного графа

X_r	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}
Концевые вершины ребер, (i,j)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(2,3)	(2,4)	(2,5)	(2,6)	(3,4)	(3,5)	(3,6)	(4,5)	(4,6)	(5,6)
Веса ребер, P_r	0	0	1	1	0	0	1	0,5	1	1	1	1	0	1	0

Для данного графа уравнения (12) и их характеристики Δ_i и γ_i будут иметь следующий вид:

$$1)\ X_1^0X_2+X_1^1X_3+X_1^1X_4+X_1^0X_5+X_2^1X_3+X_2^1X_4+$$
$$+X_2^0X_5+X_3^2X_4+X_3^1X_5+X_4^1X_5=1;\Delta_1=2;\gamma_1=8,$$

$$2)\ X_1^0X_6+X_1^1X_7+X_1^{0,5}X_8+X_1^1X_9+X_6^1X_7+X_6^{0,5}X_8+$$
$$+X_6^1X_9+X_7^{1,5}X_8+X_7^2X_9+X_8^{1,5}X_9=1;\Delta_2=2;\gamma_2=10,$$

$$3)\ X_2^0X_6+X_2^1X_{10}+X_2^1X_{11}+X_2^1X_{12}+X_6^1X_{10}+X_6^1X_{11}+$$
$$+X_6^1X_{12}+X_{10}^2X_{11}+X_{10}^2X_{12}+X_{11}^2X_{12}=1;\Delta_3=2;\gamma_3=12,$$

$$4)\ X_3^2X_7+X_3^2X_{10}+X_3^1X_{13}+X_3^2X_{14}+X_7^2X_{10}+X_7^1X_{13}+$$
$$+X_7^2X_{14}+X_{10}^1X_{13}+X_{10}^2X_{14}+X_{13}^1X_{14}=1;\Delta_4=1;\gamma_4=15,$$

$$5)\ X_4^{1,5}X_8+X_4^2X_{11}+X_4^1X_{13}+X_4^1X_{15}+X_8^{1,5}X_{11}+X_8^{0,5}X_{13}+$$
$$+X_8^{0,5}X_{15}+X_{11}^1X_{13}+X_{11}^1X_{15}+X_{13}^0X_{15}=1;\Delta_5=2;\gamma_5=12,$$

$$6)\ X_5^1X_9+X_5^1X_{12}+X_5^1X_{14}+X_5^0X_{15}+X_9^2X_{12}+X_9^2X_{14}+$$
$$+X_9^1X_{15}+X_{12}^2X_{14}+X_{12}^1X_{15}+X_{14}^1X_{15}=1;\Delta_2=2;\gamma_6=13^*. \quad (16)$$

Составляются последовательности ребер, соответствующие уравнениям (16) (в последовательностях над каждым ребром показана его весовая характеристика):

$$1)\ 2\overset{0}{-}1\overset{0}{-}3;\ \ 2\overset{0}{-}1\overset{1}{-}4;\ \ 2\overset{0}{-}1\overset{0}{-}6;\ \ 2\overset{0}{-}1\overset{0}{-}5;\ \ 3\overset{0}{-}1\overset{1}{-}4;$$
$$5\overset{1}{-}1\overset{0}{-}3;\ \ 6\overset{0}{-}1\overset{0}{-}3;\ \ 4\overset{1}{-}1\overset{1}{-}5;\ \ 6\overset{0}{-}1\overset{1}{-}4;\ \ 5\overset{1}{-}1\overset{0}{-}6;$$

$$2)\ 1\overset{0}{-}2\overset{0}{-}3;\ \ 1\overset{0}{-}2\overset{1}{-}4;\ \ 1\overset{0}{-}2\overset{0,5}{-}5;\ \ 1\overset{0}{-}2\overset{1}{-}6;\ \ 4\overset{1}{-}2\overset{0}{-}3;$$
$$5\overset{0,5}{-}2\overset{0}{-}3;\ \ 3\overset{0}{-}2\overset{1}{-}6;\ \ 4\overset{1}{-}2\overset{0,5}{-}5;\ \ 4\overset{1}{-}2\overset{1}{-}6;\ \ 6\overset{1}{-}2\overset{0,5}{-}5;$$

$$3)\ 1\overset{0}{-}3\overset{0}{-}2;\ \ 1\overset{0}{-}3\overset{1}{-}4;\ \ 1\overset{0}{-}3\overset{1}{-}5;\ \ 1\overset{0}{-}3\overset{1}{-}6;\ \ 2\overset{0}{-}3\overset{1}{-}4;$$
$$2\overset{0}{-}3\overset{1}{-}5;\ \ 2\overset{0}{-}3\overset{1}{-}6;\ \ 4\overset{1}{-}3\overset{1}{-}5;\ \ 4\overset{1}{-}3\overset{1}{-}6;\ \ 5\overset{1}{-}3\overset{1}{-}6;$$

$$4)\ 1\overset{1}{-}4\overset{1}{-}2;\ \ 1\overset{1}{-}4\overset{1}{-}3;\ \ 1\overset{1}{-}4\overset{0}{-}5;\ \ 1\overset{1}{-}4\overset{1}{-}6;\ \ 2\overset{1}{-}4\overset{1}{-}3;$$
$$2\overset{1}{-}4\overset{0}{-}5;\ \ 2\overset{1}{-}4\overset{1}{-}6;\ \ 3\overset{1}{-}4\overset{0}{-}5;\ \ 3\overset{1}{-}4\overset{1}{-}6;\ \ 5\overset{0}{-}4\overset{1}{-}6;$$

$$5)\ 1\overset{1}{-}5\overset{0,5}{-}2;\ \ 1\overset{1}{-}5\overset{1}{-}3;\ \ 1\overset{1}{-}5\overset{0}{-}4;\ \ 1\overset{1}{-}5\overset{0}{-}6;\ \ 2\overset{0,5}{-}5\overset{1}{-}3;$$
$$2\overset{0,5}{-}5\overset{0}{-}4;\ \ 2\overset{0,5}{-}5\overset{0}{-}6;\ \ 3\overset{1}{-}5\overset{0}{-}4;\ \ 3\overset{1}{-}5\overset{0}{-}6;\ \ 4\overset{0}{-}5\overset{0}{-}6;$$

$$6)\ 1\overset{0}{-}6\overset{1}{-}2;\ \ 1\overset{0}{-}6\overset{1}{-}3;\ \ 1\overset{0}{-}6\overset{1}{-}4;\ \ 1\overset{0}{-}6\overset{0}{-}5^*;\ \ 2\overset{1}{-}6\overset{1}{-}3;$$
$$2\overset{1}{-}6\overset{1}{-}4;\ \ 2\overset{1}{-}6\overset{0}{-}5;\ \ 3\overset{1}{-}6\overset{1}{-}4;\ \ 3\overset{1}{-}6\overset{0}{-}5;\ \ 4\overset{1}{-}6\overset{0}{-}5^*. \quad (17)$$

В последовательности, соответствующей базовому уравнению (она помечена звездочкой), выбирается пара ребер с минимальным суммарным весом – это ребра $1\overset{0}{-}6\overset{0}{-}5$ – с суммарной весовой характеристикой, равной 0. Далее на основе ребра 6–5 (это ребро первого яруса) формируются ребра следующего яруса. Для этого в последовательности 5) ищется пара ребер, содержащая ребро 6–5 с минимальным весом, которую можно подсоединить к предыдущей. Такой парой является последовательность ребер $4\overset{0}{-}5\overset{0}{-}6$, и таким образом, сформированы ребра второго яруса. Объединяя пары $1\overset{0}{-}6\overset{0}{-}5$ и $4\overset{0}{-}5\overset{0}{-}6$, получим цепочку $1\overset{0}{-}6\overset{0}{-}5\overset{0}{-}4$. Далее в последовательности 4) ищется пара (пары), содержащая ребро 5–4 с минимальным весом – такими парами являются ребра $1\overset{1}{-}4\overset{0}{-}5;\ \ 3\overset{1}{-}4\overset{0}{-}5;\ \ 5\overset{0}{-}4\overset{1}{-}6$. При этом подсоединение пары ребер $1\overset{1}{-}4\overset{0}{-}5$ приводит к образованию раннего цикла $1\overset{0}{-}6\overset{0}{-}5\overset{0}{-}4\overset{1}{-}1$, а подсоединение пары $5\overset{0}{-}4\overset{1}{-}6$ невозможно, поскольку вершина 6 уже присутствует в решении. Поэтому можно подсоединить к предыдущему решению либо пару $3\overset{1}{-}4\overset{0}{-}5$, либо $2\overset{1}{-}4\overset{0}{-}5$, имеющие одинаковый вес (сформированы ребра третьего яруса). Объединяя их с предыдущим решением, получим две ветки: $1\overset{0}{-}6\overset{0}{-}5\overset{0}{-}4\overset{1}{-}3$ и $1\overset{0}{-}6\overset{0}{-}5\overset{0}{-}4\overset{1}{-}2$. Сначала продолжается построение первой ветки: для этого основе ребра 4–3 формируются ребра четвертого яруса, для чего в 3) последовательности ищется пара, содержащая ребро 4–3 с минимальным весом. Такой парой являются ребра $2\overset{0}{-}3\overset{1}{-}4$ и $1\overset{0}{-}3\overset{1}{-}4$, но объединение с $1\overset{0}{-}3\overset{1}{-}4$ приводит к получению раннего цикла $1\overset{0}{-}6\overset{0}{-}5\overset{0}{-}4\overset{1}{-}3\overset{0}{-}1$, поэтому объединяется $2\overset{0}{-}3\overset{1}{-}4$ с предыдущим решением, что приводит к решению $1\overset{0}{-}6\overset{0}{-}5\overset{0}{-}4\overset{1}{-}3\overset{0}{-}2$. Далее на основе ребра 3–2 во 2) последовательности формируются ребра пятого яруса, ищется пара, содержащая ребро 3–2 с минимальным весом. Такой парой являются ребра $1\overset{0}{-}2\overset{0}{-}3$, объединение которых с предыдущим решени-

ем позволяет получить $1\overset{0}{-}6\overset{0}{-}5\overset{0}{-}4\overset{1}{-}3\overset{0}{-}2\overset{0}{-}1$. Получен цикл, поскольку номер яруса стал равным 6–1=5, и при этом во всех последовательностях присутствует пара ребер и в последовательности 1), которую не анализировалась, – это пара $2\overset{0}{-}1\overset{0}{-}6$ из данного цикла. Аналогичным способом строится вторая ветка, которая будет иметь вид $1\overset{0}{-}6\overset{0}{-}5\overset{0}{-}4\overset{1}{-}2\overset{0}{-}3\overset{0}{-}1$, являющаяся циклом с длиной, равной 1. Поскольку оба цикла имеют одинаковую длину, то они в исходном графе являются кратчайшими. Следует отметить, что наращивание цепочек ребер можно вести в любую сторону и при этом результат не изменится.

Рассмотрим еще один пример с полным 6-вершинным графом, в котором всем ребрам, образующим цикл 1–2–3–4–5–6–1, присвоен вес, равный нулю, а остальным ребрам – произвольным образом присвоены веса, равные 1 и 0. Особенностью данного графа является то, что длины всех кратчайших гамильтоновых циклов в нем равны 0. Граф задается следующим набором ребер $\{X_r\}$ с весовыми характеристиками $\{P_r\}$ в виде табл. 5.

Составим последовательности ребер, соответствующие уравнениям (12):

$$
\begin{aligned}
&1)\ 2\overset{0}{-}1\overset{0}{-}3;\ \ 2\overset{0}{-}1\overset{1}{-}4;\ \ 2\overset{0}{-}1\overset{0}{-}6;\ \ 2\overset{0}{-}1\overset{0}{-}5;\ \ 3\overset{0}{-}1\overset{1}{-}4;\\
&\quad 5\overset{1}{-}1\overset{0}{-}3;\ \ 6\overset{0}{-}1\overset{0}{-}3;\ \ 4\overset{1}{-}1\overset{1}{-}5;\ \ 6\overset{0}{-}1\overset{1}{-}4;\ \ 5\overset{1}{-}1\overset{0}{-}6;\\
&2)\ 1\overset{0}{-}2\overset{0}{-}3;\ \ 1\overset{0}{-}2\overset{1}{-}4;\ \ 1\overset{0}{-}2\overset{0}{-}5;\ \ 1\overset{0}{-}2\overset{0}{-}6;\ \ 4\overset{1}{-}2\overset{0}{-}3;\\
&\quad 5\overset{0}{-}2\overset{0}{-}3;\ \ 3\overset{0}{-}2\overset{0}{-}6;\ \ 4\overset{1}{-}2\overset{0}{-}5;\ \ 4\overset{1}{-}2\overset{0}{-}6;\ \ 6\overset{0}{-}2\overset{0}{-}5;\\
&3)\ 1\overset{0}{-}3\overset{0}{-}2;\ \ 1\overset{0}{-}3\overset{0}{-}4;\ \ 1\overset{0}{-}3\overset{0}{-}5;\ \ 1\overset{0}{-}3\overset{1}{-}6;\ \ 2\overset{0}{-}3\overset{0}{-}4;\\
&\quad 2\overset{0}{-}3\overset{0}{-}5;\ \ 2\overset{0}{-}3\overset{0}{-}6;\ \ 4\overset{0}{-}3\overset{0}{-}5;\ \ 4\overset{0}{-}3\overset{1}{-}6;\ \ 5\overset{0}{-}3\overset{0}{-}6;\\
&4)\ 1\overset{1}{-}4\overset{1}{-}2;\ \ 1\overset{1}{-}4\overset{0}{-}3;\ \ 1\overset{1}{-}4\overset{0}{-}5;\ \ 1\overset{1}{-}4\overset{0}{-}6;\ \ 2\overset{1}{-}4\overset{0}{-}3;\\
&\quad 2\overset{1}{-}4\overset{0}{-}5;\ \ 2\overset{1}{-}4\overset{0}{-}6;\ \ 3\overset{0}{-}4\overset{0}{-}5;\ \ 3\overset{0}{-}4\overset{0}{-}6;\ \ 5\overset{0}{-}4\overset{0}{-}6^{*};\\
&5)\ 1\overset{1}{-}5\overset{0}{-}2;\ \ 1\overset{1}{-}5\overset{0}{-}3;\ \ 1\overset{1}{-}5\overset{0}{-}4;\ \ 1\overset{1}{-}5\overset{0}{-}6;\ \ 2\overset{0}{-}5\overset{0}{-}3;\\
&\quad 2\overset{0}{-}5\overset{0}{-}4;\ \ 2\overset{0}{-}5\overset{0}{-}6;\ \ 3\overset{0}{-}5\overset{0}{-}4;\ \ 3\overset{0}{-}5\overset{0}{-}6;\ \ 4\overset{0}{-}5\overset{0}{-}6;\\
&6)\ 1\overset{0}{-}6\overset{1}{-}2;\ \ 1\overset{0}{-}6\overset{1}{-}3;\ \ 1\overset{0}{-}6\overset{1}{-}4;\ \ 1\overset{0}{-}6\overset{0}{-}5;\ \ 2\overset{1}{-}6\overset{1}{-}3;\\
&\quad 2\overset{1}{-}6\overset{1}{-}4;\ \ 2\overset{1}{-}6\overset{0}{-}5;\ \ 3\overset{1}{-}6\overset{1}{-}4;\ \ 3\overset{1}{-}6\overset{0}{-}5;\ \ 4\overset{1}{-}6\overset{0}{-}5. \qquad (18)
\end{aligned}
$$

Базовым уравнением является уравнение, составленное для 4-й вершины, с максимальными характеристиками $\Delta_{i=4}=2$ и $\gamma_{i=4}=8$, соответствующая ему последовательность ребер помечена звездочкой в (18). В последовательности 4) из (18) три пары ребер имеют минимальный вес, равный 0: 3–4–5; 3–4–6; 5–4–6. На основе этих пар процедура ***A4*** построит три дерева (рис. 5).

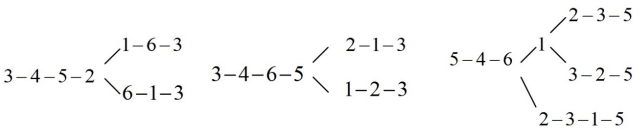

Рис. 5. Деревья, построенные процедурой *A4*

На рис. 5 все ребра деревьев имеют вес, равный 0, то есть процедура построила в исходном графе все кратчайшие циклы, которые можно построить на основе пар ребер 3–4–5; 3–4–6; 5–4–6. В общем случае, кратчайшие циклы, построенные от разных пар ребер, соответствующих базовому уравнению, могут иметь разную длину. Поэтому следует выбрать из них самый короткий , что реализуется процедурой ***A4*** на восьмом шаге ее работы.

Рассмотрим вопрос оптимальности работы процедуры ***A4***. Последовательности ребер, соответствующих уравнениям (12), можно рассматривать как множества $\{U_i\}$ объектов $\{Q_r\}$, имеющих определенную стоимость $\{C_r\}$. Требуется выбрать по одному объекту из каждого множества таким образом, чтобы суммарная стоимость выбранных объектов была минимальной при этом после выбора очередного объекта следующий можно выбрать только в случае, если он удовлетворяет некоторым правилам R. Совокупность объектов из каждого множества U_i, удовлетворяющих правилу R, будем называть полной. Роль правила R реализуется процедурой ***B***. Работа процедуры ***A4*** заключается в том, что в некотором множестве $U_{i=s}$ выбирается объект $Q_{r=d}(C^{min}_{r=d})$ минимальной стоимости, и при этом все остальные объекты, находящиеся в этом множестве, имеют большую стоимость. Далее среди всех объектов, удовлетворяющих правилу R в множестве U_i, находим следующий объект $Q_{r+1}(C^{min}_{r+1})$ с минимальной стоимостью, при этом стоимость $C^{min}_{r+1}<C_r\in U_i$, где C_r – стоимости объектов, принадлежащих U_i, но неудовлетворяющих правилам R. Процедура продолжает работать до тех пор, пока в соответствии с правилами R из всех множеств U_i не будет выбрано по одному объекту с суммарной стоимостью S_d. Правило R отождествляется с процедурой ***B,*** а это значит, что если процедура ***A4*** построила один цикл, то он является кратчайшим при условии, что каждый раз при выборе пар ребер в последовательностях нет ситуации, когда в последовательностях встречаются несколько пар ребер с минимальным весом. В дальнейшем о такой ситуации будем говорить как о некотором свойстве ν. При наличии свойства ν при решении уравнения (12) процедура ***A4*** строит λ деревьев всех цепочек из ребер, которые на последнем шаге замыкаются в циклы с μ ветками, где λ – количество пар ребер с минимальным весом в базовой последовательности с минимальным весом.

Таблица 5

Описание 6-вершинного полного графа с нулевым весом в цикле

X_r	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}
Концевые вершины ребер (i,j)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(2,3)	(2,4)	(2,5)	(2,6)	(3,4)	(3,5)	(3,6)	(4,5)	(4,6)	(5,6)
Веса ребер P_r	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0

Каждое дерево, построенное на основе одной из пар ребер X_lX_k с минимальным весом, принадлежащих базовой последовательности, является деревом всех кратчайших цепочек, которое можно построить на основе ребер соответствующих паре переменных X_lX_k. После того, как все деревья построены процедура ***A4*** из всех возможных кратчайших циклов, построенных с учетом возникновения свойства ν, выбирает самый короткий цикл, который и является решением рассматриваемой задачи. Следует отметить, что в принципе начало работы процедуры ***A4*** можно начинать с любой последовательности, но в процедуре ***A4*** выбирается базовая последовательность с наибольшим значением параметров Δ_i и γ_i, что позволяет существенно сократить число веток при построении деревьев кратчайших цепей. Это можно пояснить на примере для графа, заданного табл. 1. Если построение начинается не с базовой последовательности i=6 с $\Delta_{i=6}=2$ и $\gamma_{i=6}=13$, а с последовательности i=2 с $\Delta_{i=2}=2$ и $\gamma_{i=2}=10$, то в этом случае дерево цепей строится на основе пары ребер $1\overset{0}{-}2\overset{0}{-}3$ с минимальным суммарным весом, равным 0, и при этом в построенном дереве цепей будет не две ветки, а три (рис. 6).

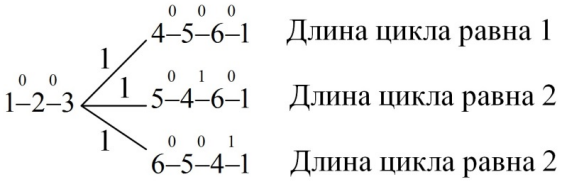

Рис. 6. Дерево, построенное процедурой *A4* на основе пары ребер 1–2–3

В худшем случае, общее число веток, построенных в деревьях кратчайших цепей процедурой ***A4***, равно $\phi=\lambda\cdot\mu$, и оно зависит от числа пар ребер минимального веса, используемых на каждом шаге работы процедуры ***A4***. Сколько таких пар, столько и ветвлений в дереве, построенных процедурой ***A4***. Оценим временную сложность работы процедуры ***A4***.

Число пар ребер во всех последовательностях, соответствующих уравнениям (12), равно

$$\alpha=\frac{(n-1)(n-2)}{2},$$

поэтому для анализа одной последовательности требуется не более $(n-1)(n-2)<n^2$ операций сравнения. Так как требуется проанализировать *n* таких последовательностей, то для построения одного цикла с помощью процедуры ***A4*** понадобится не более n^3 операций сравнения. Для расчета параметров Δ_i и γ_i понадобится не более n^2 и $2\log_2 n$ операций выбора минимального элемента из n^2 чисел. Поскольку общее число веток, построенных процедурой ***A4***, равно ϕ, то для определения длин всех веток, содержащих по *n* ребер, понадобится $\phi\cdot n$ операций сложения. Для построения всех веток для выбора ветки минимальной длины требуется еще $\log_2\phi$ операций (для выбора минимального элемента из массива из ϕ чисел). Таким образом, общая временная сложность работы процедуры не превысит:

$$O(n^3+n^2+2\log_2 n+n\phi)\phi+$$
$$+\log_2\phi\approx O(n^3(1+\frac{\phi}{n^2})\phi+\log_2\phi). \quad (20)$$

Ясно, что если построить деревья от всех пар ребер в произвольной последовательности без выбора минимальных элементов, то дерево будет содержать число веток, равное

$$\frac{(n-1)!}{2}.$$

Процедура ***A4*** строит не все деревья, а равное количеству пар ребер с минимальными весами, находящихся в базовой последовательности. Число веток в самом дереве зависит только от числа пар ребер с минимальными весами на каждом шаге процедуры ***A4***. Так, в примере анализа 6-вершинного графа (табл. 4), в базовом множестве оказались две пары ребер минимального веса, и процедура ***A4*** построила дерево, состоящее из одной ветки, то есть из множества 60 различных циклов, выполнив около $2n^3$ элементарных операций, процедура построила кратчайший цикл. В случае задания 6-вершинного графа, заданного табл. 5, 10 ребер имеют вес, равный 0, и пять ребер – равных 1. Таким образом, в анализируемых последовательностях почти 60 % ребер участвуют в образовании минимальных пар ребер и при этом процедура строит три дерева (рис. 7), содержащие всего 7 веток, и для определения кратчайшего гамильтонова цикла процедура ***A4*** выполнила не более $7n^3$ элементарных операций. Из приведенных примеров следует, что при увеличении диапазона изменения весов ребер можно ожидать, что среднее число веток, которые будет строить предложенная процедура ***A4***, будет соизмерима с размерностью задачи *n* из за того, что в процессе работы процедуры ***A4*** число пар ребер с минимальным весом на каждом шаге ее работы будет уменьшаться, и при этом алгоритм ее решения будет иметь в среднем полиномиальную сложность. Для проверки данной гипотезы проведен эксперимент. При равномерном законе распределения весов ребер определялось среднее значение числа веток, построенных процедурой ***A4*** при изменении весовых характеристик ребер в диапазонах от 0 до 5; от 0 до 30 и от 0 до 50 и при различных значениях n. Все результаты получены с доверительной вероятностью 0,95. Погрешность предложенной процедуры при равномерном законе распределения весовых характеристик ребер графа при изменении числа вершин в диапазоне от 10 до 100 не превысила в среднем 20 %. Преимуществом данной процедуры является то, что ее можно адаптировать для решения задачи определения гамильтоновости произвольных графов, и с увеличением диапазона изменения весовых коэффициентов ребер графа число выполняемых шагов процедурой ***A4*** уменьшается, как и погрешность ее работы.

6. Обсуждение результатов исследования предложенного метода мониторинга РВС на основе методов определения кратчайших путей и кратчайших гамильтоновых циклов в графе

Использование двухэтапного подхода для решения задачи мониторинга РВС на основе методов, использу-

ющих ранговый подход, позволяет реализовать эффективное распараллеливание их реализации на основе технологий CUDA. Для реализации этапа 1 при числе узлов топологии РВС, равного n, требуется время, не превышающее значение $O(n)$. Аналогично можно реализовать распараллеливание метода решения задачи на этапе 2. При этом, как показало экспериментальное исследование, с увеличением диапазона изменения временных задержек, возникающих при передаче информации между узлами РВС, что возможно при высокой интенсивности и объемах, передаваемых по коммуникационным каналам РВС данных, уменьшается время выполнения и повышается точность решения задач на этапе 2.

7. Выводы

1. Предложена модель мониторинга состояния объектов мониторинга РВС на основе минимизации суммарного времени запусков удаленных сервисов мониторинга объектов на ресурсах РВС.

2. Предложен двухэтапный метод решения задачи минимизации суммарной задержки запуска удаленных агентов, использующий на этапе 1 стянутое дерево всех путей и распараллеливание процесса решения на узлах РВС, что частично решает проблему необходимости разработки специальных архитектур для решения поставленных задач.

3. Для решения задачи этапа 2 предложен метод определения кратчайших гамильтоновых циклов с полиномиальной временной сложностью, который позволяет эффективно решить задачу определения последовательности запуска удаленных сервисов для мониторинга объектов на ресурсах РВС в реальном времени. Особенностью предложенного метода является сведение решения ЗК к задаче решения системы квадратичных булевых уравнений с задаваемыми характеристиками.

Предложенные для реализации обоих этапов методы и алгоритмы позволяют использовать технологии параллельного программирования их реализации с использованием CUDA-технологий, позволяющих значительно повысить качество работы системы мониторинга РВС.

Литература

1. Nagios – The Industry Standard in IT Infrastructure Monitoring [Electronic resource]. – Available at: http://www.nagios.org
2. Icinga. Open Source Monitoring [Electronic resource]. – Available at: https://www.icinga.org
3. Минухин, С. В. Информационные технологии реализации двухуровневой модели планирования пакетов заданий в распределенной вычислительной системе на основе решения задачи о наименьшем покрытии [Текст] / С. В. Минухин // Системи управління, навігації та зв'язку. – 2015. – Вип. 1 (33). – С. 111–115.
4. Пономаренко, В. С. Методы и модели планирования ресурсов в GRID-системах [Текст]: монография / В. С. Пономаренко, С. В. Листровой, С. В. Минухин и др. – Харьков: ИНЖЭК, 2008. – 408 с.
5. Кофман, А. Методы и модели исследования операций. Целочисленное программирование [Текст] / А. Кофман, А. Анри-Лабродер. – М.: Мир, 1977. – 236 с.
6. Вагнер, Г. Основы исследования операций [Текст] / Г. Вагнер. – М.: Мир. 1973. – Т. 2. – 489 с.
7. Bellman, R. Dynamic Programming Treatment of the Travelling Salesman Problem [Text] / R. Bellman // Journal of the ACM. – 1962. – Vol. 9, Issue 1. – P. 61–63. doi: 10.1145/321105.321111
8. Held, M. The Traveling-Salesman Problem and Minimum Spanning Trees [Text] / M. Held, R. M. Karp // Mathematical Programming. – 1971. – Vol. 1, Issue 1. – P. 6–25. doi: 10.1007/bf01584070
9. Gurevich, Y. Expected computation time for Hamiltonian path problem [Text] / Y. Gurevich, S. Shelah // SIAM Journal on Computing. – 1987. – Vol. 16, Issue 3. – P. 486–502. doi: 10.1137/0216034
10. Bjorklund, A. Exact Algorithms for Exact Satisfiability and Number of Perfect Matchings [Text] / A. Bjorklund, T. Husfeldt // Algorithmica. – 2008. – Vol. 52. – P. 226–249. doi: 10.1007/s00453-007-9149-8
11. Koivisto, M. A space-time tradeoff for permutation problems [Text] / M. Koivisto, P. Parviainen // ACM-SIAM Symposium on Discrete Algorithms: Proceedings of the Twenty-First Annual. – Austin, 2010. – P. 484–492. doi: 10.1137/1.9781611973075.41
12. Kohn, S. A generating function approach to the traveling salesman problem [Text] / S. Kohn, A. Gottlieb, M. Kohn // ACM '77 : Proceedings of the 1977 annual conference. – New York, USA, 1977. – P. 294–300. doi: 10.1145/800179.810218
13. Karp, R. M. Dynamic Programming Meets the Principle of Inclusion and Exclusion [Text] / R. M. Karp // Operations Research Letters. – 1982. – Vol. 1, Issue 2. – P. 49–51. doi: 10.1016/0167-6377(82)90044-x
14. Bax, E. A Finite-Difference Sieve to Count Paths and Cycles by Length [Text] / E. Bax, J. Franklin // Information Processing Letters. – 1996. – Vol. 60, Issue 4. – P. 171–176. doi: 10.1016/s0020-0190(96)00159-7
15. Bjorklund A. Determinant Sums for Undirected Hamiltonicity [Text] / A. Bj orklund // Foundations of Computer Science : Proceedings of the 51st Annual Symposium. – Washington, DC, 2010. – P. 173–182. doi: 10.1109/FOCS.2010.24
16. Woeginger, G. J. Open Problems Around Exact Algorithms [Text] / G. J. Woeginger // Discrete Applied Mathematics. – 2008. – Vol. 156, Issue 3. – P. 397–405. doi: 10.1016/j.dam.2007.03.023
17. Eppstein, D. The traveling salesman problem for cubic graphs [Text] / D. Eppstein // Algorithms and Data Structures. Lecture Notes in Computer Science. – 2003. – Vol. 2748. – P. 307–318. doi: 10.1007/978-3-540-45078-8_27
18. Iwama, K. An improved exact algorithm for cubic graph TSP [Text] / K. Iwama, T. Nakashima // Computing and Combinatorics. Lecture Notes in Computer Science. – 2007. – Vol. 4598. – P. 108–117. doi: 10.1007/978-3-540-73545-8_13
19. Gebauer, H. Finding and enumerating hamilton cycles in 4-regular graphs [Text] / H. Gebauer // Theoretical Computer Science. – 2011. – Vol. 412, Issue 35. – P. 4579–4591. doi: 10.1016/j.tcs.2011.04.038

20. Могила, І. А. Вплив параметрів мурашиного алгоритму на розв'язок задачі комівояжера [Текст] / І. А. Могила, І. І. Лобач, О. А. Якимець // Східно-Європейський журнал передових технологій. – 2014. – Т. 4, № 4 (70). – С. 18–23. doi: 10.15587/1729-4061.2014.26290
21. Боронихина, Е. А. Сравнение методов решения задачи коммивояжера. Ч. 3 [Текст]: материалы XIII Междунар. науч.-практ. конф. им. А. Ф. Терпугова/ Е. А. Боронихина, В. А. Сибирякова // Информационные технологии и математическое моделирование И74 (ИТММ–2014). – Томск: Изд-во Том. ун-та, 2014. – С. 18–21.
22. Курейчик, В. М. Об алгоритмах решения задачи коммивояжера с временными ограничениями [Текст] / В. М. Курейчик, А. В. Мартынов // Информатика, вычислительная техника и инженерное образование. – 2014. – № 1 (16). – С. 1–13.
23. Частикова, В. А. Разработка и сравнительный анализ эвристических алгоритмов для поиска наименьшего гамильтонова цикла в полном графе [Текст] / В. А. Частикова, К. А. Власов // Фундаментальные исследования. – 2013. – № 10. – С. 63–67.
24. Li, Y. A New Exact Algorithm for Traveling Salesman Problem with Time Complexity Interval (O(n4), O(n3*2n)) [Electronic resource] / Y. Li. – Available at: http://arxiv.org/abs/1412.2437
25. Серая, О. В. Анализ методов решения транспортных задач со случайными стоимостями перевозок [Текст] / О. В. Серая // Информационно-управляющие системы на железнодорожном транспорте. – 2013. – № 4. – С. 42–45.
26. Listrovoy, S. V. Method of Minimum Covering Problem Solution on the Basis of Rank Approach [Text] / S. V. Listrovoy, Yu. GUL // Engineering Simulation. – 1999. – Vol. 17. – P. 73–89.
27. Listrovoy, S. V. Solution method on the basis of rank approach for integer linear problems with boolean variables [Text] / S. V. Listrovoy, D. Yu. Golubnichiy, E. S. Listrovaya // Engineering Simulation. – 1999. – Vol. 16. – P. 707–725.

В роботі розглянуто проблему підвищення ефективності роботи магістральних водоводів в сучасних умовах при переході на трьохзонні тарифи з електроенергії. Запропоновано ефективний метод вирішення цієї проблеми, заснований на використанні специфічних особливостей магістральних водоводів як стохастичних об'єктів, що функціонують в стохастичному середовищі

Ключові слова: оптимальне стохастичне управління, ймовірнісні обмеження на фазові змінні, магістральний водовід

В работе рассматривается проблема повышения эффективности работы магистральных водоводов в современных условиях при переходе на трёхзонные тарифы по электроэнергии. Предложен эффективный метод решения этой проблемы, основанный на использовании специфических особенностей магистральных водоводов как стохастических объектов, функционирующих в стохастической среде

Ключевые слова: оптимальное стохастическое управление, вероятностные ограничения на фазовые переменные, магистральный водовод

УДК 628.12, 628.14
DOI: 10.15587/1729-4061.2015.55469

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ И МЕТОД ОПТИМАЛЬНОГО СТОХАСТИЧЕСКОГО УПРАВЛЕНИЯ РЕЖИМАМИ РАБОТЫ МАГИСТРАЛЬНОГО ВОДОВОДА

А. Д. Тевяшев
Доктор технических наук, профессор,
заведующий кафедрой*
E-mail: tad45@mail.ru
О. И. Матвиенко
Аспирант*
E-mail: olga_mat@ukr.net
*Кафедра прикладной математики
Харьковский национальный университет радиоэлектроники
пр. Ленина, 14, г. Харьков, Украина, 61166

1. Введение

Резкое возрастание тарифов на электроэнергию и введение трёхзонного тарифа создали необходимые условия для перехода к энергосберегающим технологиям управления магистральным водоводом (МВ).

МВ представляет собой сложную техническую систему, предназначенную для транспорта воды на большие расстояния. МВ состоит из последовательности многоцеховых насосных станций (НС) и многониточных магистральных трубопроводов [1]. На входе каждой НС имеется резервуары чистой воды (РЧВ). В РЧВ первой НС МВ поступает подготовленная вода с одного или нескольких подъёмов. На выходе МВ, как правило, имеются РЧВ значительной ёмкости, используемые в качестве источников водоснабжения для городов и населённых пунктов [2].