ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ОПТИМИЗАЦИИ РАЗВИТИЯ ТРАНСПОРТНОЙ СЕТИ НА ОСНОВЕ АЛГОРИТМА ЭДМОНДСА-КАРПА

Н.Л. Григоренко, А.В. Жарков, Д.Г. Пивовартчук, Н.Н. Попова

1.1. Введение

Под транспортной сетью в данной работе понимается сеть поставок, объединяющая некоторое количество поставщиков, потребителей и дистрибьюторов. Поставщики генерируют некоторый поток, под которым, в каждом конкретном случае, можно понимать, например, энергию, энергоносители, товары и т.п. Потоки от поставщиков либо доходит непосредственно до потребителей, либо достигает распределителей, которые перераспределяют входящие в них потоки по нескольким направлениям, после чего потоки доходят до потребителей. Поток, дошедший до потребителя, поглощается им.

Целью рассматриваемой задачи является оптимизация управления развитием сети поставок от поставщика к потребителю. Сеть представляет собой набор заданных связей между поставщиками, дистрибьюторами и потребителями. Под оптимизацией развития сети поставок понимается увеличение пропускных способностей ее связей, с целью максимального удовлетворения спроса потребителей.

Задача рассматривается в предположении, что объемы поставок и спросы потребителей изменяются со временем. Развитие сети поставок должно учитывать этот факт и увеличивать пропускные способностей тех элементов сети, в направлении которых осуществляется и/или будут осуществляться основные поставки. При этом в рассматриваемой постановке задачи не предполагается, что задан единственный сценарий для изменение спроса и объема поставок в будущем. Предполагается, что для каждого потребителя/поставщика задан целый набор возможных сценариев (изменения спроса/предложения в будущем) и вероятность каждого возможного сценария. В условиях описанной неопределенности под решением задачи понимается адаптивная стратегия управления развитием сети поставок, то есть предполагается, что на каждом шаге рассматриваемого процесса наблюдаются текущие объемы поставок от всех поставщиков и текущие спросы всех потребителей и на основе этой информации принимается решение о направлениях дальнейшего развития сети.

На практике такая задача возникает в различных прикладных областях. Например, в процессе эксплуатации электрической сети имеется необходимость проводить различные мероприятия по поддержанию работоспособности системы, которые требуют отключения отдельных участков сети, ограничения их пропускной способности или ввод новых участков.

В данной работе предлагается алгоритм решения задачи оптимизации управления развитием инфраструктурой транспортной сети и математическое доказательство его корректности. Последовательный алгоритм решения данной задачи требует значительных вычислительных операций. Предлагается эффективный параллельный алгоритм.

1.2. Модель инфраструктуры ``поставщик--потребитель''.

Сеть поставок представляет собой направленный граф. Каждая вершина графа является одним из следующих объектов: *поставщик*, *потребитель* или *дистрибьютор*. Будем обозначать: поставщиков --- S_1, ..., S_{N_S}, потребителей --- D_1, ..., D_{N_D} и дистрибьюторов --- B_1, ..., B_{N_B}.

Каждая пара вершин может иметь направленную связь, которая характеризуется пропускной способностью (т.е. максимальным возможным потоком через эту связь в единицу времени). Если пропускная способность связи между двумя вершинами больше нуля, то между этими вершинами возможен поток, объем которого в единицу времени не превосходит пропускной способности связи.

Направленную связь, соединяющую вершины i, j, и направленную от вершины i к вершине j будем обозначать $C_{i,j}$. Соответствующую этой связи пропускную способность будем обозначать $c_{i,j}$, а поток через эту связь будем обозначать $f_{i,j}$. Например, связь, соединяющую поставщика S_i и потребителя D_j обозначим C_{s_i, d_j}, ее пропускную способность c_{s_i, d_j}, а поток f_{s_i, d_j}.

Будем считать, что выполняются следующие естественные предположения:

a) Если вершина соответствует поставщику, то эта вершина может иметь только исходящие связи с потребителями или дистрибьюторами;

b) Если вершина соответствует потребителю, то эта вершина может иметь только входящие связи от поставщиков или дистрибьюторов;

c) Если вершина соответствует дистрибьютору, то эта вершина имеет по крайней мере одну входящую и одну исходящую связь. дистрибьютор может иметь входящие связи от поставщиков или других дистрибьюторов и исходящие связи с потребителями или другими дистрибьюторами.

d) Каждый поставщик S_i, $i = 1,\ldots,N_S$, генерирует исходящий поток и характеризуется числом s_i, которое определяет объем генерируемого им потока в единицу времени.

e) Каждый потребитель D_i, $i = 1,\ldots,N_D$, принимает входящий поток и характеризуется числом d_i, которое определяет объем потока, который этот потребитель может принять в единицу времени.

f) Каждый дистрибьютор B_i, $i = 1,\ldots,N_B$, принимаемый входящий поток и превращает его в исходящий поток. дистрибьютор характеризуется числом b_i, которое определяет объем потока, который этот дистрибьютор может перераспределить.

g) Потоки, произведенные всеми поставщиками, распределяются по сети в соответствии с некоторым правилом, которое будем назвать *статическим правилом распределения потоков*.

h) Суммарный поток, ушедший от поставщиков, в полном объеме доходит до потребителей. Причем, не предполагается, что от потребителей уходит весь поток, который он может генерировать.

Таким образом, получаем следующую структуру функционирования инфраструктуры за единицу времени. Каждый поставщик производит некоторый объем исходящего потока. Потоки, произведенный всеми поставщиками, распределяются по сети в соответствии со статическим правилом распределения потоков (это правило будет определено далее). Потоки, дошедшие до потребителей, принимаются ими в соответствие с их возможностями.

Обозначим множество вершин, имеющих входящие связи в вершину k, через

$$C_{*,k} = \{C_{i,j} \mid j = k\},$$

а множество вершин, имеющих исходящие связи из вершины k, через

$$C_{k,*} = \{C_{i,j} \mid i = k\}.$$

Используя введенные обозначения, запишем ограничения, которым должны удовлетворять потоки $f_{i,j}$ по связями:

R1) Поток вдоль связи не превосходит пропускную способность связи

$$0 \le f_{i,j} \le c_{i,j}.$$

R2) Ограничение на поток производимый поставщиком S_i, $i = 1,\ldots,N_S$,

$$\sum_{j \in C_{s_i,*}} f_{s_i,j} \le s_i.$$

R3) Ограничение на поток принимаемый потребителем D_i, $i = 1,\ldots,N_D$,

$$\sum_{j \in C_{*,d_i}} f_{j,d_i} \le d_i.$$

R4) Ограничение на поток принимаемый и перераспределяемый дистрибьютором B_i, $i = 1,\ldots,N_B$,

$$\sum_{j \in C_{*,b_i}} f_{j,b_i} \le b_i.$$

R5) Сохранение суммарного объема потока при его распределении дистрибьютором B_i, $i = 1,\ldots,N_B$,

$$\sum_{j \in C_{*,b_i}} f_{j,b_i} = \sum_{j \in C_{b_i,*}} f_{b_i,j}.$$

Статическое правило распределения потоков задается задачей минимизации

$$\sum_{i=1}^{N_D} \left[d_i - \sum_{j \in C_{*,d_i}} f_{j,d_i} \right] \to \min_{\{f_{i,j}\}},$$

при ограничениях R1--R5. Задача $()$ означает стремление максимально удовлетворить спрос потребителей. Отметим, что минимум в последней задачи достигается, т.к. ограничения R1--R5 задают ограниченное

замкнутое множество, а функционал является непрерывной функцией своих аргументов.

1.3. Динамическая модель.

Рассмотрим заданный период времени функционирования инфраструктуры и предположим, что этот период разбит на N непересекающихся интервалов. Предполагаем, что потоки, а также состояния поставщиков, потребителей и дистрибьюторов постоянны на одном интервале времени. Для обозначения времени будем использовать переменную $t = 0,1,\ldots,N$. Считаем, что каждая переменная, зависящая от времени, постоянная на интервале $[t, t+1)$, $t = 0,1,\ldots,N-1$.

Начальные пропускные способности всех связей заданы

$$c_{i,j}(0) = c_{i,j}^0.$$

Предполагаем, что мы имеем возможность управлять пропускными способностями связей. Точнее, на каждом интервале времени мы имеем возможность увеличить пропускные способности связей, причем предполагаем, что связи, чьи пропускные способности могут быть увеличены разбиты на группы. К одной группе относятся связи, чьи пропускные способности должны быть увеличены за один и тот же интервал времени. К одной группе могут относится произвольные связи, например, одна связь или последовательность связей представляющая собой путь от некоторого поставщика к некоторому потребителю или набор несвязанных между собой связей. Предполагаем, что для каждой связи каждой группы задано то значение, на которое предполагается увеличивать пропускную способность этой связи. Таким образом, одна группа представляет собой некоторое одно мероприятие по увеличению пропускных способностей связей за один период времени.

Заданные группы связей будем обозначать $G^1,\ldots,G^M$, где каждая группа характеризуется набором

$$G^q = \{\bar{u}_{i,j}^q\}, \quad q = 1,\ldots,M,$$

где $\bar{u}_{i,j}^q$ -- заданное значение, на которое будет увеличена пропускная способность связи $C_{i,j}$ из группы G^q. Предполагаем, что $M \geq N$, т.е. количество возможных мероприятий не меньше, чем количество интервалов времени.

Обозначим через $G(t)$ множество индексов тех групп, которые еще не были задействованы к моменту времени t, и обозначим через $\bar{G}(t)$ индексы тех групп, которые уже были задействованы к моменту времени t.

Рассмотрим управляющий параметр $q(t)$, который определяет индекс группы связей, чьи пропускные способности будет увеличены на интервале времени $[t, t+1)$. Тогда можно записать следующие соотношения

$$\begin{cases} \bar{G}(t+1) = \bar{G}(t) \cup q(t), \\ \bar{G}(0) = \emptyset, \end{cases}$$

где управляющий параметр $q(t)$ может принимать значения только из множества $G(t)$.

Множество индексов $\bar{G}(t)$ для момента времени t однозначно определяет пропускные способности всех связей в момент t следующим образом

$$c_{i,j}(\bar{G}(t)) = c_{i,j}^0 + \sum_{q \in \bar{G}(t)} \bar{u}_{i,j}^q.$$

Отметим, что $\bar{G}(t)$ принимает значения из множества $\bar{\mathsf{G}}$, где через $\bar{\mathsf{G}}_k$ мы обозначили множество всех возможных выборок k чисел из множества $\{1,\ldots,M\}$ с точностью до их перестановок и $\bar{\mathsf{G}}_0 = \emptyset$.

Управление развитием инфраструктуры заключается в последовательном выборе на каждом интервале времени группы связей, чьи пропускные способности должны быть увеличены.

Будем предполагать, что задан набор *сценариев* (функций от времени), определяющий объемы потоков, которые будут производить каждый из поставщиков на каждом интервале времени, $s_i(t)$, объемы потоков, которые смогут принять каждый из потребителей на каждом интервале времени, $d_i(t)$, объемы потоков, которые сможет перераспределить каждый из дистрибьюторов на каждом интервале времени, $b_i(t)$. Таким образом, один сценарий представляет собой набор функций

$$(s_1(t),\ldots,s_{N_S}(t), d_1(t),\ldots,d_{N_D}(t), b_1(t),\ldots,b_{N_B}(t)).$$

Для того, чтобы описать целый набор сценариев и каждому из сценариев поставить в соответствие вероятность, воспользуемся понятием марковского процесса. Рассмотрим набор значений (s,d,b) -- те значения, которые характеризуют объемы, которые генерируются поставщиками, $s \in \mathsf{R}^{N_S}$, могут быть приняты потребителями, $d \in \mathsf{R}^{N_D}$, и могут быть распределены дистрибьюторами, $b \in \mathsf{R}^{N_B}$. Зафиксируем множество возможных наборов значений

$$\mathsf{H} = \left\{ H^e = (s^e, d^e, b^e) \right\}_{e=1,\ldots,E}$$

т.е. состояний рассматриваемого марковского процесса и зададим переходные вероятности между состояниями

$$p_{i+1}(s_{i+1}, d_{i+1}, b_{i+1} \mid s_i, d_i, b_i),$$

для всех $(s_{i+1}, d_{i+1}, b_{i+1})$, $(s_i, d_i, b_i) \in \mathsf{H}$. Так определенный марковский процесс будет описывать множество сценариев и вероятность каждого сценария.

Пусть задан некоторый сценарий, т.е. набор функции описывающий как будут изменяться поставки, спрос и возможности по распределению потоков. Выбирая на каждом интервале времени группу связей, чьи пропускные способности будут увеличены, в соответствии с равенствами (), (), получаем изменяющиеся во времени пропускные способности связей. Все это приводит к изменению потоков от интервала к интервалу, при этом потоки должны удовлетворять ограничениям R1--R5. Получаем следующие ограничения, которые должны выполняться для каждого $t = 0,1,\ldots,N$

$$\begin{cases} 0 \le f_{i,j}(t) & \le & c_{i,j}(\bar{G}(t)), \\ \sum\limits_{j \in C_{s_i,*}} f_{s_i,j}(t) & \le & s_i(t), \\ \sum\limits_{j \in C_{*,d_i}} f_{j,d_i}(t) & \le & d_i(t), \\ \sum\limits_{j \in C_{*,b_i}} f_{j,b_i}(t) & \le & b_i(t), \\ \sum\limits_{j \in C_{*,b_i}} f_{j,b_i}(t) & = & \sum\limits_{j \in C_{b_i,*}} f_{b_i,j}(t). \end{cases}$$

Обозначим через $F(\bar{G}(t),(s(t),d(t),b(t)))$ множество значений $\{f_{i,j}(t)\}$, удовлетворяющих системе (). Отметим, что это множество замкнутое и ограниченное.

1.4. Критерий качества и задача оптимизации.

Рассмотрим следующую критерий качества

$$I(s_0,d_0,b_0) = E\left[\sum_{t=0}^{N} \beta_t \sum_{i=1}^{N_D} \left(d_i(t) - \sum_{j \in C_{*,d_i}} f_{j,d_i}(t) \right) \right] \to \min_{\{f_{i,j}(t|\cdot)\}_{t=0,\ldots,N},\, \{q(t|\cdot)\}_{t=0,\ldots,N-1}},$$

где $\beta_k > 0$, $k = 0,\ldots,N$, --- весовые коэффициенты упорядоченные следующим образом

$$\beta_0 < \beta_1 < \ldots < \beta_N.$$

Эти коэффициенты отражают тот факт, что конечное состояние развиваемой инфраструктуры важнее чем промежуточные состояния. Математическое ожидание берется по все возможным сценариям, порожденным марковским процессом (), () с заданным начальным состоянием процесса (s_0,d_0,b_0). При этом, управляющий параметр $q(t)$ удовлетворяет ограничениям

$$\begin{cases} \bar{G}(t+1) & = & \bar{G}(t) \cup q(t), \quad q(t) \in G(t), \\ \bar{G}(0) & = & \emptyset, \end{cases}$$

а потоки удовлетворяют ограничениям

$$\begin{cases} 0 \le f_{i,j}(t) & \le & c_{i,j}(\bar{G}(t)), \\ \sum\limits_{j \in C_{s_i,*}} f_{s_i,j}(t) & \le & s_i(t), \\ \sum\limits_{j \in C_{*,d_i}} f_{j,d_i}(t) & \le & d_i(t), \\ \sum\limits_{j \in C_{*,b_i}} f_{j,b_i}(t) & \le & b_i(t), \\ \sum\limits_{j \in C_{*,b_i}} f_{j,b_i}(t) & = & \sum\limits_{j \in C_{b_i,*}} f_{b_i,j}(t). \end{cases}$$

Отметим, что минимизация осуществляется по множеству функций $\{f_{i,j}(t \mid \cdot)\}$, $q(t \mid \cdot)$, которые зависят от множества G незадействованных к моменту t групп и текущего состояния (s,d,b), т.е.

$$\{f_{i,j}(t \mid G,(s,d,b))\}, \quad q(t \mid G,(s,d,b)).$$

Таким образом, мы ищем оптимальное адаптивное управление развитием инфраструктуры, т.е. такое управление, которое по результатам наблюдений состояний поставщиков, потребителей и дистрибьюторов на каждом интервале времени определяет группу связей, чьи пропускные способности увеличить, из набора еще незадействованных групп. Отметим тот факт, что минимальное значение функционала достигается, т.к. аргументы функций $q(t \mid \cdot)$, $\{f_{i,j}(t \mid \cdot)\}$ принимают значения из конечных множеств, образ функций $q(t \mid \cdot)$ -- конечное множество, а образ функций $\{f_{i,j}(t \mid \cdot)\}$ -- замкнутое ограниченное множество $F(\bar{G}(t),(s(t),d(t),b(t)))$. Оптимальное значение функционала будем обозначать $I^*(s_0,d_0,b_0)$.

Рассмотрим вспомогательную задачу линейного программирования, зависящую от заданного множества индексов $\bar{G}$ и заданного набора значений (s,d,b):

$$\sum_{i=1}^{N_D} \left[d_i - \sum_{j \in C_{*,d_i}} f_{j,d_i} \right] \to \min_{\{f_{i,j}\}}$$

при ограничениях

$$\begin{cases} 0 \le f_{i,j} & \le & c_{i,j}(\bar{G}), \\ \sum\limits_{j \in C_{s_i,*}} f_{s_i,j} & \le & s_i, \\ \sum\limits_{j \in C_{*,d_i}} f_{j,d_i} & \le & d_i, \\ \sum\limits_{j \in C_{*,b_i}} f_{j,b_i} & \le & b_i, \\ \sum\limits_{j \in C_{*,b_i}} f_{j,b_i} & = & \sum\limits_{j \in C_{b_i,*}} f_{b_i,j}. \end{cases}$$

Оптимальное значение функционала в последней задаче обозначим $W(\bar{G},(s,d,b))$, а оптимальное решение $\{f^W_{i,j}(\bar{G},(s,d,b))\}$.

Решение исходной задачи ()--() основано на следующем утверждение:

Утверждение. Оптимальное значение функционала в задаче ()--() удовлетворяет равенству

$$I^*(s_0,d_0,b_0) = V_0(\emptyset,(s_0,d_0,b_0)),$$

где функция $V_0(\bar{G},(s,d,b))$ вычисляется из следующих рекуррентных соотношений:

1) Для $\tau = 0,\ldots,N-1$, всех $\bar{G}_\tau \in \bar{\mathsf{G}}_\tau$ и всех $(s_\tau,d_\tau,b_\tau) \in \mathsf{H}$

$$V_\tau(\bar{G}_\tau,(s_\tau,d_\tau,b_\tau)) = \beta_\tau W(\bar{G}_\tau,(s_\tau,d_\tau,b_\tau)) +$$

$$+ \min_{q_\tau \in G_\tau} \left\{ \sum_{(s_{\tau+1},d_{\tau+1},b_{\tau+1}) \in H} p_{\tau+1}(s_{\tau+1},d_{\tau+1},b_{\tau+1} \mid s_\tau,d_\tau,b_\tau) V_{\tau+1}(\bar{G}_\tau \cup q_\tau,(s_{\tau+1},d_{\tau+1},b_{\tau+1})) \right\}.$$

2) Для всех $\bar{G}_N \in \bar{G}_N$ и всех $(s_N,d_N,b_N) \in H$

$$V_N(\bar{G}_N,(s_N,d_N,b_N)) = \beta_N W(\bar{G}_N,(s_N,d_N,b_N)).$$

Оптимальное адаптивное управления в задаче ()--() имеет следующий вид: пусть в момент времени τ индексы уже включенных групп связей задаются множеством $G_\tau \in \bar{G}_\tau$ и наблюдается состояние $(s_\tau,d_\tau,b_\tau) \in H$, тогда

$$f^*_{i,j}(\tau \mid G_\tau,(s_\tau,d_\tau,b_\tau))) = f^W_{i,j}(\bar{G}_\tau,(s_\tau,d_\tau,b_\tau)),$$

$$q^*(\tau \mid G_\tau,(s_\tau,d_\tau,b_\tau))) =$$

$$= arg \min_{q_\tau \in G_\tau} \left\{ \sum_{(s_{\tau+1},d_{\tau+1},b_{\tau+1}) \in H} p_{\tau+1}(s_{\tau+1},d_{\tau+1},b_{\tau+1} \mid s_\tau,d_\tau,b_\tau) V_\tau(\bar{G}_\tau \cup q_\tau,(s_{\tau+1},d_{\tau+1},b_{\tau+1})) \right\}.$$

2. Параллельный алгоритм и его реализация.

Опишем схему нахождения оптимального решения задачи. Алгоритм включает в себя следующие шаги:

1) Решение экстремальных задач.

Для каждого $\tau = 0,1,\ldots,N$, множества $\bar{G} \in \bar{G}_\tau$ и набора $(s,d,b) \in H$ решаем вспомогательную задачу линейного программирования ()--() и находим значения $W(\bar{G},(s,d,b))$.

2) Решение уравнения Беллмана[2].

Для каждого $\tau = 0,1,\ldots,N$, множества $\bar{G} \in \bar{G}_\tau$ и набора $(s,d,b) \in H$ на основе рекуррентных соотношений ()--() вычисляем значения $V_\tau(\bar{G},(s,d,b))$.

3) Построение последовательности оптимального управления.

Пусть наблюдается последовательность состояний $\{(s_t,d_t,b_t)\}_{t=0,\ldots,N}$. Оптимальное управление строится следующим образом. Для начального момента времени задаем $\bar{G}(0) = \emptyset$ и $G(0) = \{1,\ldots,M\}$,

$$q_0^* = arg \min_{q_0 \in \{1,\ldots,M\}} \left\{ \sum_{(s_1,d_1,b_1) \in H} p_1(s_1,d_1,b_1) \mid s_0,d_0,b_0) V_0(q_0,(s_1,d_1,b_1)) \right\}.$$

Для момента времени $t = 1$, получаем $\bar{G}(1) = \{q_0^*\}$ и $G(1) = \{1,\ldots,M\} \setminus \{q_0^*\}$, значение q_1^* вычисляем по формуле (), и т.д. Для момент времени $t = k$, получаем $\bar{G}(k) = \{q_0^*,\ldots,q_{k-1}^*\}$ и $G(k) = \{1,\ldots,M\} \setminus \{q_0^*,\ldots,q_{k-1}^*\}$,

$$q_k^* = arg \min_{q_k \in G(k)} \left\{ \sum_{(s_{k+1},d_{k+1},b_{k+1}) \in H} p_{k+1}(s_{k+1},d_{k+1},b_{k+1}) \mid s_k,d_k,b_k) V_k(q_k,(s_{k+1},d_{k+1},b_{k+1})) \right\}.$$

Таким образом, получаем оптимальную последовательность $(q_0^*,\ldots,q_{N-1}^*)$ включения групп связей, соответствующую заданной наблюдаемой последовательности $\{(s_t,d_t,b_t)\}_{t=0,\ldots,N}$.

Обозначим:

N_L - число дуг графа транспортной сети

N_S - число поставщиков

N_D - число потребителей

N_B - число распределителей

E - число состояний марковского процесса

$Q = M$ - число групп, моментов времени

2.1 Решение экстремальных задач.

Для вычисления значений $W(\bar{G},(s,d,b))$ необходимо решить задачу линейного программирования[3] ()--(), которая представляет собой поиск максимального потока в графе транспортной сети при ограничениях (). Для решения этой задачи используется алгоритм Эдмондса-Карпа[1] на преобразованном графе:

- Добавляется вершина S, которая становится единственным источником потока;
- Добавляется дуга от S к S_i, $i = 1,\ldots,N_S$ с пропускной способностью $C_{S,S_i} = s_i$;
- Добавляется новая вершина D, которая становится единственным потребителей потока;
- Добавляется дуга от D к D_i, $i = 1,\ldots,N_D$ с пропускной способностью $C_{S,S_i} = d_i$;
- Каждая вершина B_i, $i = 1,\ldots,N_B$ заменяется двумя вершинами $B1_i$ и $B2_i$, соединенными дугой из $B1_i$ в $B2_i$ пропускной способности $C_{B1_i,B2_i} = b_i$;
- Ограничения пропускных способностей s_i, d_i, b_i с вершин снимаются;

Для преобразованного графа решается задача поиска максимального потока алгоритмом Эдмондса-Карпа и вычисляется значение $W(\bar{G},(s,d,b))$.

Решение экстремальных задач выполняется на P-1 slave-процессорах. Распределением задач управляет master-процесс методом динамической балансировки[5]. Этот метод позволяет добиться максимальной эффективности параллельных вычислений для данной задачи, т.к. априорной информации о количестве итераций в алгоритме Эдмондса-Карпа нет. Коммуникационные обмены реализованы через интерфейс MPI. Master-процесс рассылает номера состояний марковского процесса и индексы задействованных групп свободному slave-процессу, который выполняет поиск максимального потока и возвращает результат master-процессу. После решения всех экстремальных задач master-процесс производит решение уравнения Белламана.

2.2. Решение уравнения Беллмана и построение последовательности оптимального управления

Решение уравнения Беллмана осуществляется последовательно нахождением значений $V_\tau(\bar{G},(s,d,b))$ для $\tau = Q\ldots 0$ согласно формуле ().

2.3. Объемы использования оперативной памяти в параллельной реализации алгоритма

Для оценки параметров модели, для которой может быть решена задача оптимизации разработанной параллельной реализацией, необходимо оценить требуемые объемы оперативной памяти. Обозначим $N_V = 2 + N_S + N_D + 2N_B$ - число вершин в преобразованном графе. Для внутреннего представления графа транспортной сети требуется $K(7N_L + 13N_V)$ байт, информации о состояниях марковского процесса $KE(E + N_V)$, информации о группах связей $4KQ$. Коэффициент K зависит от используемого компилятора и для проведенного вычислительного эксперимента на суперкомпьютере Lomonosov $K = 8$. Суммарный объем требуемой оперативной памяти равен

$$K(7N_L + N_V(E + 13) + E^2 + 4Q)$$

байт.

2.4. Исследование эффективности и масштабируемости параллельного алгоритма

Для измерения эффективности и масштабируемости предложенного параллельного алгоритма были выполнены вычислительные эксперименты на тестовых входных данных. Эксперименты проводились на суперкомпьютере Lomonosov, установленном в Московском государственном университете им. М.В.Ломоносова.

Обозначим параметры, характеризующие размер задачи и влияющие на время ее решения: $N_1 = N_V N_L^2$ – оценка сложности поиска максимального потока алгоритмом Эдмондса — Карпа, $N_2 = E * 2^Q$ -- число экстремальных подзадач.

Составлены тестовые входных данные с различным числом моментов модельного времени $Q = 2..20$. Размер графа транспортной сети был подобран таким образом, чтобы время тестирования модели не превышало установленные ограничения. Связи между вершинами графа были выбраны следующим образом:

- $N_L / 2$ связей, соединяющих случайно выбранные поставщик и распределитель
- $N_L / 2$ связей, соединяющих случайно выбранные распределитель и сток

Группы дуг $G^1,\ldots,G^Q$: число связей в каждой группе N_L / Q, то есть все дуги графа транспортной

сети равномерно распределены по группам и каждая группа состоит из случайно выбранных дуг графа. Количество состояний системы $E = 64$. Матрица перехода из состояния в состояние P, такая что $P_{ij} = 1/E$. Подробный список значений параметров модели Q, N_L, N_S, N_D, N_B и соответствующих им значений вычислительной сложности задачи N_1, N_2 для всех тестовых моделей приведены в таблице

Таблица 1. Параметры тестовых моделей

№	N_1	N_2	N_L	N_S	N_D	N_B	Q
1	$63 \cdot 10^9$	2048	$11 \cdot 10^3$	130	130	130	6
2	$63 \cdot 10^9$	8192	$11 \cdot 10^3$	130	130	130	7
3	$63 \cdot 10^9$	32768	$11 \cdot 103$	130	130	130	9
4	$63 \cdot 10^9$	524288	$11 \cdot 10^3$	130	130	130	13
5	$3 \cdot 10^{21}$	128	10^7	10^7	10^7	10^7	1

В соответствии с формулой () и учитывая, что $N_L > N_V$, для 12 Гбайт ОЗУ узла суперкомпьютера Lomonosov возможно решение задач для транспортных сетей с числом дуг до 10^7. Для экспериментального подтверждения этой теоретической оценки сгенерирована модель 5 (см. таблицу 1) и для неё решена успешно задача оптимизации.

Ввиду стохастической генерации дуг в графе транспортной сети для каждого набора параметров сгенерировано 5 различных тестовых моделей. Итоговое время решения задачи оптимизации для каждого набора параметров рассчитано как среднее по времени решения задачи для каждой из 5 тестовой модели.

Эффективность параллельных вычислений для предложенного алгоритма определена как отношение ускорения к числу параллельных процессов[4].

Суперкомпьютер Lomonosov содержит 5130 вычислительных узлов, содержащих 2 четырехядерных процессора Intel Xeon X5570 Nehalem, 12 ГБайт оперативной памяти и интерфейсы ввода/вывода. Вычислительный эксперимент был проведен на тестовых задачах (см. табл. 1) с использованием процессоров от 2 до 2048. Узлы суперкомпьютера работали в режиме SMP с одной нитью.

На рисунках - показаны графики времени решения задачи, ускорения и эффективности параллельных вычислений.

Как видно из графика, на задачах с $N_2 > 32768$ эффективность параллельных вычислений близка к 1, что подтверждает хорошую масштабируемость предложенного алгоритма.

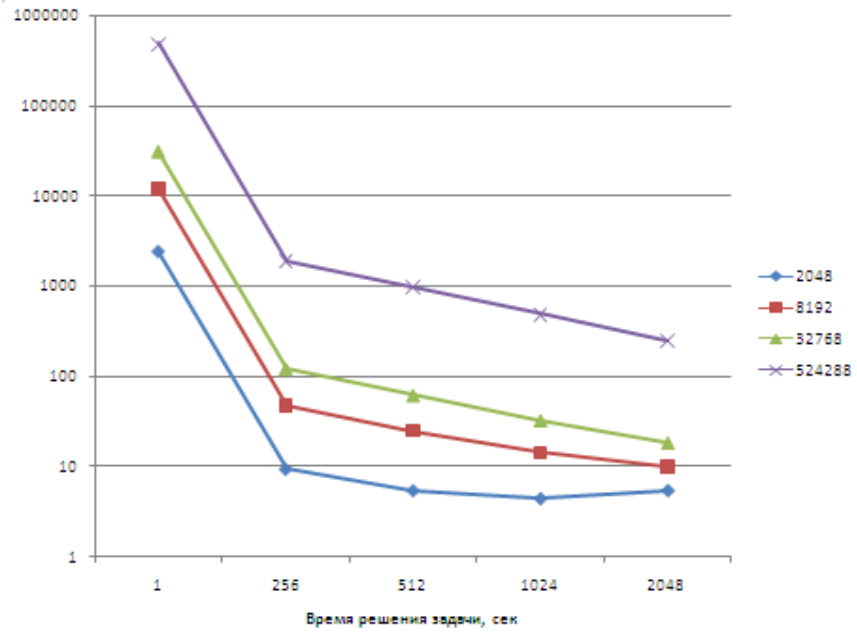

Рис. 1. Время решения задач различной вычислительной сложности

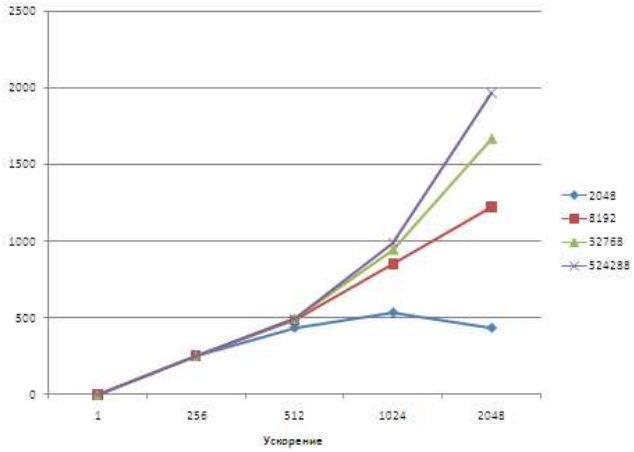

Рис. 2. Ускорение

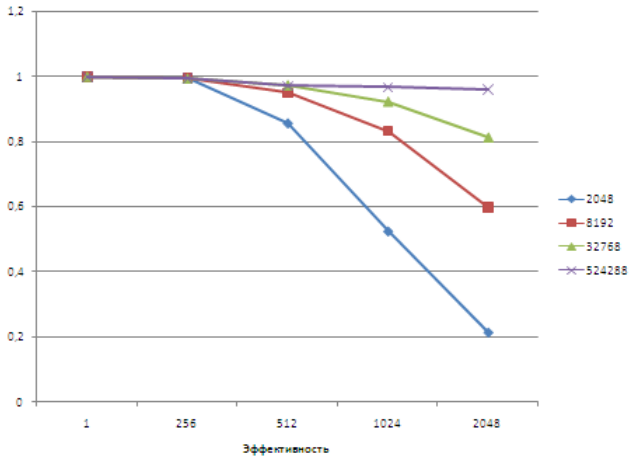

Рис. 3. Эффективность параллельных вычислений

3. Заключение

В работе предложен последовательный и параллельный алгоритмы оптимального управления динамической транспортной сетью. Предложен метод балансировки нагрузки, позволяющий получить максимальную эффективность решения данной задачи при применении параллельных вычислений. Проведено экспериментальное исследование эффективности алгоритма на суперкомпьютере Lomonosov. Определены параметры входных данных, влияющие на время ее решения: N_1 – оценка сложности поиска максимального потока алгоритмом Эдмондса-Карпа , N_2 – число экстремальных задач. Проведенный вычислительный эксперимент показал высокую (близкую к 1) эффективность алгоритма на любом количестве процессоров от 1 до 2048. Создание параллельного алгоритма решения задачи оптимизации управления транспортной сети позволяет повысить эффективность исследований в данной области и получить новые качественные результаты за счет увеличения детализованности графа и количества моментов модельного времени, а предложенная математическая модель динамической транспортной сети позволяет решать различные задачи моделирования развития, выполнения плановых мероприятий и определения ``узких'' мест сети.

ЛИТЕРТУРА:

1. Томас Кормен и др. Алгоритмы: построение и анализ = INTRODUCTION TO ALGORITHMS. — 2-е изд. — М.: «Вильямс», 2006.
2. Беллман Р. Динамическое программирование. М.: Издательство иностранной литературы, 1960.
3. Васильев Ф.П., Иваницкий А.Ю. Линейное программирование. М.: Факториал, 2003.
4. Крюков В.А. Разработка параллельных программ для вычислительных кластеров и сетей. Информационные технологии и вычислительные системы. 2003. 1-2. [PDF] (http://www.parallel.ru/tech/articles/krukov- cldvm2002f.pdf).
5. Gubenko G. Dynamic load Balancing for Distributed Memory Multiprocessors //Journal of parallel and distributed computing. – 1989. 7, pp. 279 -301.