

УДК 519.852:656.2

ОПТИМИЗАЦИЯ РАЗВИТИЯ ТРАНСПОРТНОЙ СЕТИ СО СЛУЧАЙНЫМИ ПОТОКАМИ

П.Г. Мардашова, А.А. Стягов, Г.Л. Эпштейн

Московский государственный университет путей сообщения

Россия, 127994, г. Москва, ул. Образцова, д. 9, стр. 9

E-mail: polina.mardashova@gmail.com, styagov.al@gmail.com, egl413@gmail.com

Изложена постановка задачи оптимального добавления дуг в транспортной сети со случайным многопродуктовым потоком. Представлен алгоритм решения этой задачи в случае, когда каждый продукт имеет один источник и один сток. Алгоритм основан на применении симплекс-метода, метода ветвей и границ и статистического моделирования. Приведены результаты вычислительных экспериментов.

OPTIMIZATION OF ADDITION OF A TRANSPORT NETWORK WITH RANDOM FLOWS

/ P.G. Mardashova, A.A. Styagov, G.L. Epshteyn (Moscow state university of means of communication, st. Obraztsova 9/9, Moscow, 127994, Russia). It is set up the problem the optimum supplement of a transport network with a random multyproduct flow. When each product has one source and one sink the algorithm for decision of this problem is described. This algorithm uses the simplex-method, branches and borders method and the statistical simulation. The examples of computation is represented.

Введение

В [1] была описана постановка задачи развития транспортной сети с детерминированными потоками продуктов, а также приведен алгоритм решения этой задачи. Учитывая, что в транспортных сетях различной природы с многопродуктовыми потоками, например, железнодорожных, автодорожных, информационных существует неопределенность и нестационарность потоков продуктов, представляет интерес решение указанной выше задачи в условиях неопределенности продуктовых потоков. Одним из подходов к учету неопределенности является представление потоков в виде случайных величин с известными распределениями, например, равномерным или бета распределением.

Постановка задачи включает в себя минимизацию математического ожидания суммарной стоимости транспортировки и создания новых ребер графа.

$$M(CF) = M \left(\sum_{u_k \in U} c_{ko} \sum_{l=1}^L z_{ko}^l + \sum_{v_i \in V} p_i y_i + \sum_{u_k \in U_d} q_k x_k \right) \rightarrow \min, \quad (1)$$

где

c_{ko} – стоимость перемещения единицы любого продукта по дуге u_k ;

z_{ko}^l – количество продукта l -го типа, перевозимого по дуге u_k ;

p_i – транзитные платежи за перевозку единицы продукта через вершину v_i ;

y_i – общий объем транзитных потоков, через вершину v_i ;

q_k - стоимость создания дуги u_k ;

$x_k \in \{0,1\}$ - бинарная переменная, которая равна единице в случае, когда создается k -ая дуга.

Как и в детерминированной задаче для каждой реализации совокупности случайных потоков должны выполняться ограничения.

Условие баланса в каждой вершине i , записываемое для каждого продукта l :

$$\sum_{u_k \in U_i^+} z_{ko}^l - \sum_{u_k \in U_i^-} z_{ko}^l = r_{io}^l, \quad l = 1, \dots, L; \quad i = 1, \dots, n; \quad (2)$$

где

r_{io}^l - избыток продукта l -го типа в вершине v_i , положительный в источнике и отрицательный в стоке,

U_i^+ - множество исходящих из вершины дуг,

U_i^- - множество входящих в вершину дуг.

Выражения для транзитных потоков через вершины:

$$y_i = \sum_l \left(\sum_{u_k \in U_i^-} z_{ko}^l, r_{io}^l = 0 \right), \quad l = 1, \dots, L, \quad i = \overline{1, n}; \quad (3)$$

В (3) суммируются втекающие потоки продуктов, для которых данная вершина не является источником или стоком.

Ограничения транзитных потоков через вершины:

$$y_i \leq s_{v_i}, \quad i = \overline{1, n}; \quad (4)$$

s_{v_i} - транзитная пропускная способность вершины v_i .

Ограничения на пропускные способности существующих дуг:

$$\sum_l z_{ko}^l \leq s_k, \quad k = \overline{1, K - m}. \quad (5)$$

s_k - пропускная способность дуги u_k .

Условие существования дуги, получаемое суммированием по всем продуктам, перемещаемым по одной из m вновь создаваемых дуг:

$$\sum_l z_{ko}^l \leq s_k x_k, \quad k = \overline{K - m + 1, K}. \quad (6)$$

Ограничения на увеличение локальных степеней вершин.

$$\sum_{u_k \in U_{di}} x_k \leq t_i, \quad i = \overline{1, n}. \quad (7)$$

U_{di} - множество новых дуг, инцидентных вершине i ,

t_i - максимально допустимое количество таких дуг.

Заметим, что в силу случайности исходных потоков величины z_{ko}^l , y_i , r_{io}^l также являются случайными.

Приближенный алгоритм оптимизации развития транспортной сети со случайными потоками.

Для приближенного решения поставленной задачи предлагается алгоритм, основанный на замене минимизации математического ожидания целевой функции минимизацией целевой функции для каждой реализации случайных потоков.

Получаемое при этом значение минимума само по себе становится случайной величиной. Среднее минимальных значений функции CF не может превосходить среднего значения этой функции на всем множестве новых дуг.

$$M(\min CF) \leq \min M(CF) \quad (8)$$

В ходе минимизации для каждой реализации потоков накапливается статистика вхождения допустимых новых дуг в оптимальные решения. После окончания первого этапа решения задачи множество допустимых новых дуг упорядочивается по убыванию частоты вхождения дуг в оптимальные решения. Отметим, что на первом этапе определяется также оценка среднеквадратического отклонения случайной величины $\min CF$.

На втором этапе выполняется оценка критерия (10) для расширяемого набора дуг, начиная с наиболее часто применявшихся. В качестве приближенного решения принимается набор новых дуг, удовлетворяющий условию

$$M(CF, U_d^*) = \min \left(M(CF, U_d^k) - M(\min CF) \right), \quad k = \overline{K - m + 1, K}, \quad (9)$$

то есть дающий наименьшее превышение средних затрат над их нижней оценкой.

Заметим, что при выполнении первого этапа можно оценить, как вероятность отсутствия допустимых решений, так и вероятности нарушения ограничений.

Рассмотрим пример решения задачи развития сети со случайными потоками.

Пример 1.

Исходная транспортная сеть произвольного вида содержит 9 вершин и 14 ребер. На сети требуется передать 10 продуктовых потоков. Параметры сети и потоков приведены в [1].

В результате выполнения 2000 реализаций вероятностного моделирования были найдены частоты дополнительно создаваемых ребер, расположенные в таблице 1 в порядке убывания. В третьем столбце этой же таблицы записаны математические ожидания нормированной целевой функции (то есть приведенной к единице потока), также полученные с помощью 2000 реализаций. Каждое математическое ожидание соответствует добавлению данного ребра и всех расположенных выше. Зависимость математического ожидания от количества последовательно добавляемых ребер графически представлена на рис. 1. Нижняя горизонтальная линия на этом рисунке соответствует значению $M(\min CF)$, то есть оценке снизу минимума целевой функции (1). Верхняя горизонтальная прямая указывает значение суммы нижней оценки и ее среднеквадратического отклонения. Как видно, найденное приближенное значение мало отличается от нижней оценки точного минимума, то есть и от самого минимума, а возможно, и совпадает с точным минимумом.

Таблица 1. Частоты применения дополнительных ребер.

Новые ребра	Частота применения	Матем. ож. нормир. ЦФ
1--5	1941	40,728
1--9	1903	37,808
6--8	1783	36,618
2--7	1412	35,726
1--8	214	37,169
3--8	3	39,487

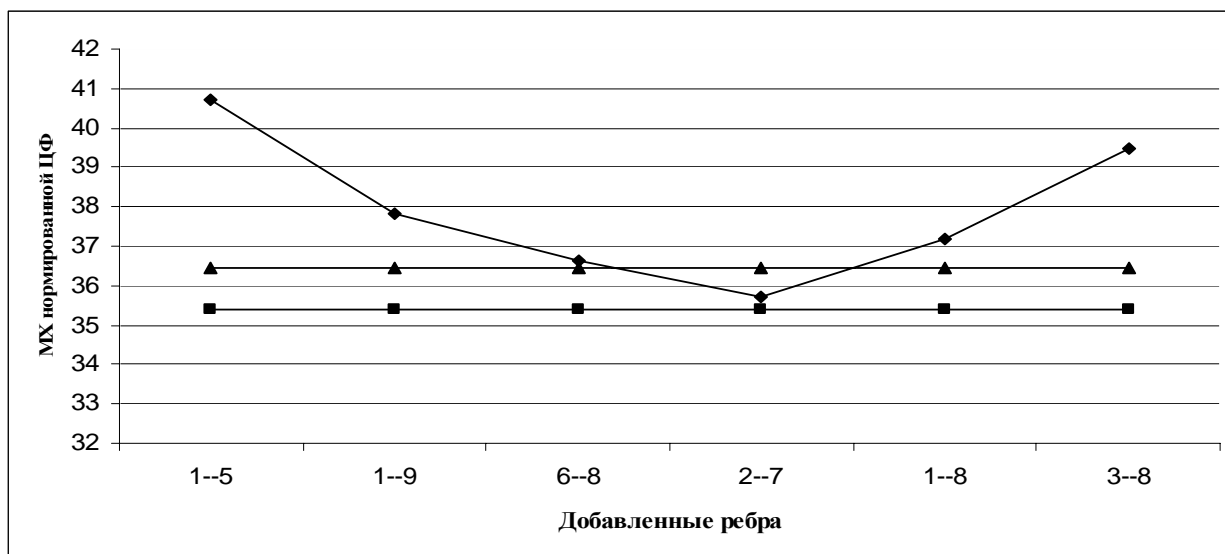


Рис. 1. Выбор оптимального решения.

Рассмотренный подход к решению задачи со случайными потоками предъявляет высокие требования к времени решения задачи минимизации при каждой реализации совокупности потоков. Как показали расчеты, значительное время занимает метод ветвей и границ, используемый для обеспечения бинарности переменных. Поэтому перед применением метода ветвей и границ был введен этап варьирования коэффициентов целевой функции. Этот прием основан на предположении, что в окрестности нецелочисленного решения имеются целочисленные крайние точки области допустимых решений. Если указанное предположение верно, то обнаружить такие точки можно путем случайного варьирования коэффициентов целевой функции.

Таким образом, после получения симплекс-методом нецелочисленного решения выполняется оценка размера дерева задач метода ветвей и границ (МВГ). Если размер дерева меньше порогового значения (то есть количество нецелочисленных переменных мало), то сразу применяется МВГ. В противном случае производится некоторое количество запусков симплекс-метода со случайным варьированием в небольших пределах коэффициентов целевой функции («покачивание» ЦФ). Если при этом обнаруживается целочисленное решение, в котором значение ЦФ достаточно близко к нижней оценке, то оно и принимается в качестве решения задачи при данной реализации потоков. Заметим, что в любой практической задаче исходные параметры известны с конечной точностью. А это позволяет задать допустимый уровень превышения нижней оценки.

Если описанный выше прием после заданного числа повторений не дает целочисленного решения, то выполняется МВГ.

На рис. 2 показаны зависимости времени решения от количества потоков, то есть от размера задачи, с применением «покачивания» и без этого приема для примера 1. Как видно, время решения значительно уменьшается, причем эффективность «покачивания» растет с увеличением количества потоков.

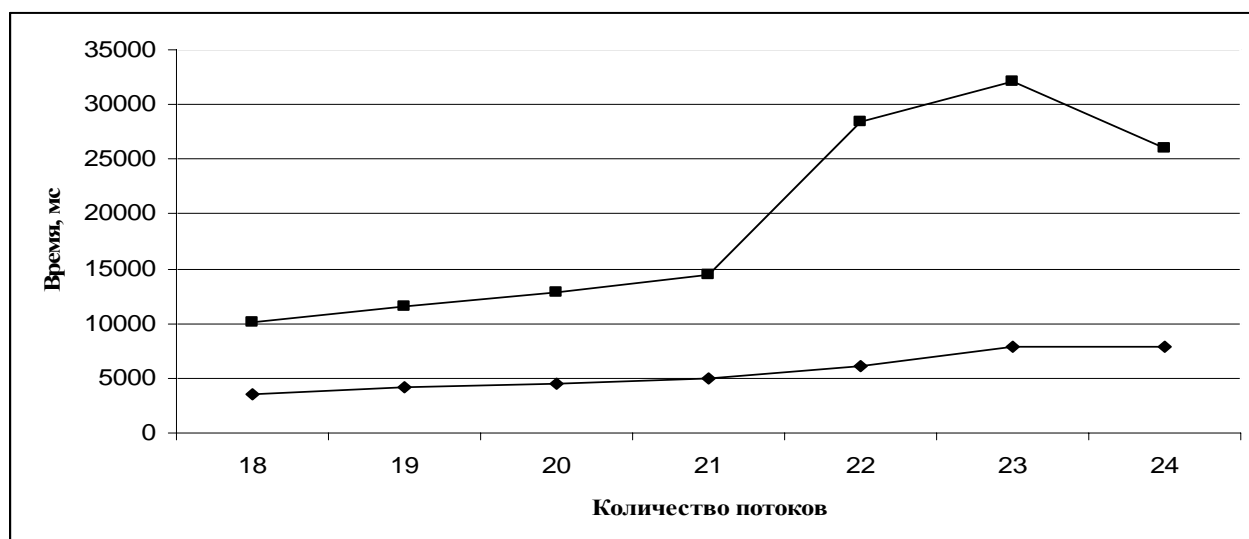


Рис. 2. Зависимость времени решения от количества потоков с применением «покачивания» ЦФ (нижняя кривая) и только с применением МВГ (верхняя кривая).

Дальнейшее сокращение времени получения результата связано с использованием параллельных вычислений на всех уровнях решения задачи со случайными потоками. Имеет место иерархия распараллеливания. На самом верхнем уровне – параллельный расчет реализаций. В ходе расчета каждой реализации – распараллеливание МВГ и «покачивания ЦФ». Наконец, применение параллельных процедур в модифицированном симплекс-методе.

Поясним подход к распараллеливанию симплекс-метода на примере решения одной реализации потоков в бесконтурной транспортной сети.

Пример 2.

Сеть содержит 12 вершин и распространяет 55 потоков продуктов между несмежными вершинами. Исходные данные представлены на рис. 3.

На главной диагонали матрицы приведены затраты на транзит единицы потока через соответствующую вершину, над главной диагональю расположены значения потоков продуктов, под ней, затраты на создания новых дуг (хорд).

Решение задачи графически представлено на рис. 4. Целочисленное решение получено только «покачиванием» ЦФ, причем потребовалось 24 испытания, значение ЦФ 16060 при нижней оценке 15840.

0	0	100	120	110	80	90	60	150	50	50	100
0	4	0	40	70	90	80	100	60	30	50	20
600	0	3	0	80	60	10	50	70	20	90	60
600	550	0	5	0	100	40	30	50	110	120	90
600	550	580	0	4	0	50	20	40	70	90	80
600	550	580	650	0	3	0	60	50	40	10	20
600	550	580	650	520	0	3	0	40	90	50	60
600	550	580	650	520	680	0	4	0	90	20	60
600	550	580	650	520	680	540	0	4	0	40	90
600	550	580	650	520	680	540	660	0	3	0	100
600	550	580	650	520	680	540	660	560	0	4	0
600	550	580	650	520	680	540	660	560	500	0	0

Рис. 3. Исходные данные к примеру 2.

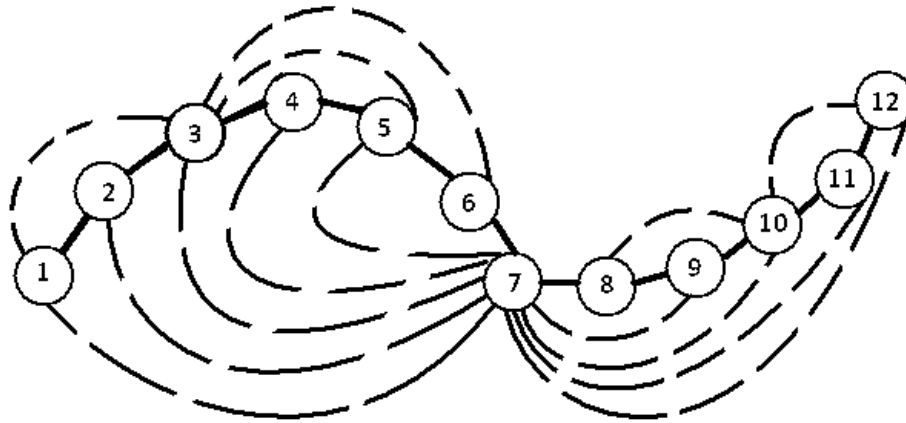


Рис. 4. Графическое представление решения примера 2. Добавленные дуги изображены прерывистыми линиями.

Пример 3.

В этом примере использованы те же данные, что и на рис. 3, но добавлено ограничение на увеличение локальной степени вершины 7, не более, чем на 2. результаты решения приведены на рис. 5. Введение нового ограничения привело к изменению целевой функции с 16060 до 16360. Так же, как и в примере 2, целочисленное решение получено только «покачиванием» ЦФ, причем потребовалось 24 испытания, при нижней оценке 15990.

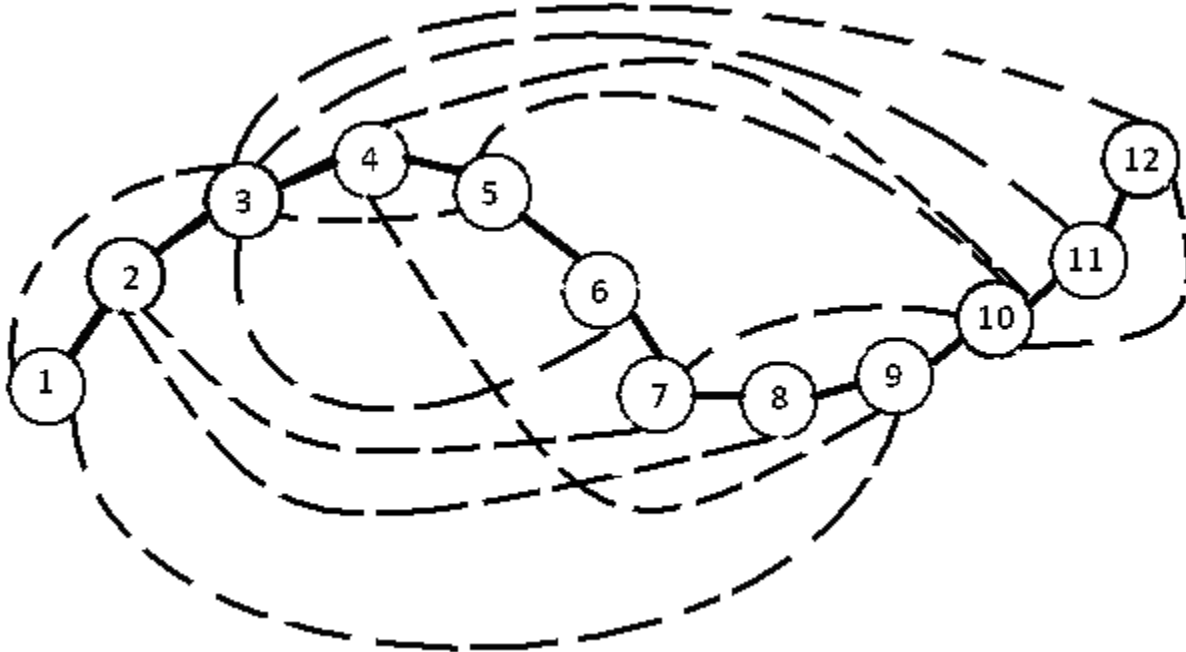


Рис.5. Графическое представление решения примера 3. Добавленные дуги изображены прерывистыми линиями.

Наибольший эффект от параллельных вычислений в симплекс-методе был получен в процедуре проверки критерия оптимальности и выбора переменной, вводимой в базис, то есть

путем распараллеливания пересчёта оценок небазисных столбцов. Эти оценки представляют собой выражение в скобках в (10)

$$P = c_B^T \beta - (\pi^T N - c_N^T) x_N \quad (10)$$

В (10) обозначены:

c_B^T - вектор-строка коэффициентов ЦФ при базисных переменных;

β - вектор свободных членов в текущем базисе;

$\pi^T = c_B^T B^{-1}$;

B^{-1} - текущая обратная матрица;

N - небазисная матрица;

c_N^T - вектор-строка коэффициентов ЦФ при небазисных переменных;

x_N - вектор небазисных переменных.

В начале работы симплекс-метода создаются дополнительные информационные потоки. Их количество определяется соображением, что всего потоков должно быть столько же, сколько ядер процессора ЭВМ. Количество ядер определяется свойством Environment.ProcessorCount. В каждом потоке запускается процедура CallFindForThread с параметром, указывающим на небазисный вектор, с которого нужно начинать считать оценки. В ней описан бесконечный цикл, содержащий 3 инструкции:

флаг ожидания сигнала из внешней среды на выполнение итерации

запуск процедуры FindDj, осуществляющей поиск оценок

флаг, передающий во внешнюю среду информацию о том, что FindDj закончила работу.

Процедура FindDj в качестве параметра получает номер небазисного вектора, с которого нужно начать расчёт оценок. Если вычисления осуществляются на ЭВМ с n -ядерным процессором, в указанной процедуре будет вычисляться оценка для каждого n -ого небазисного вектора, начиная с указанного в параметре. Например, на компьютере с четырёхядерным процессором, в FindDj, запущенной с параметром 3, будут вычисляться оценки для третьего, седьмого, десятого и т.п. векторов. Этот подход пояснен на рис. 5.

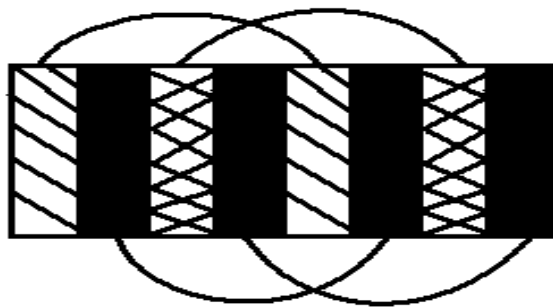


Рис. 6. Схема распараллеливания пересчета оценок небазисных столбцов.

Сопоставление результатов расчетов на 2-х ядерном и 4-х ядерном компьютерах приведено на рис. 7.

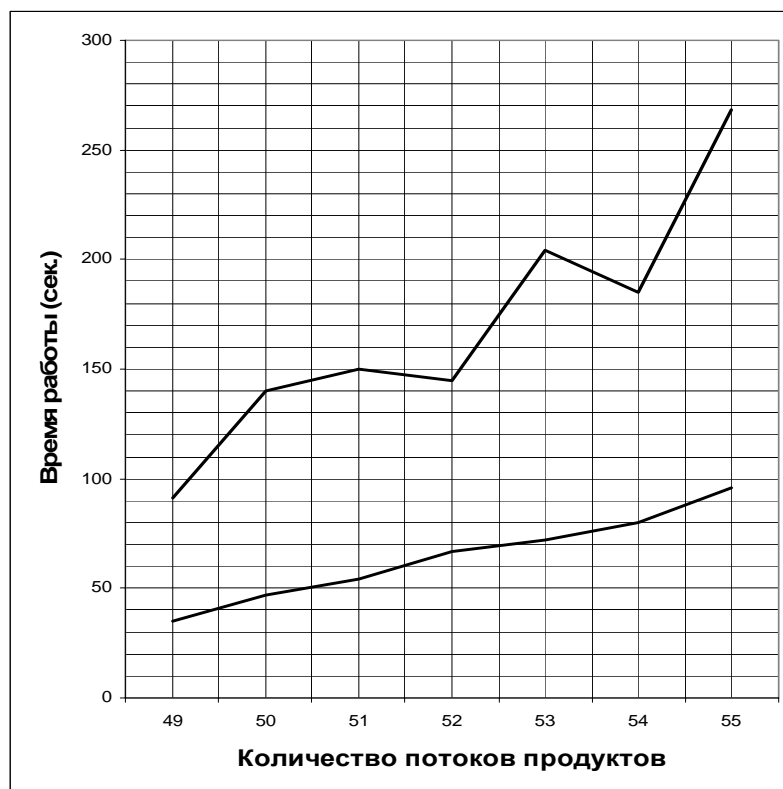


Рис. 7. Сопоставление времени решения на 2-х ядерном (верхняя кривая) и 4-х ядерном компьютерах.

Пример 4

Расчет плана формирования грузовых поездов на железнодорожном участке [2].

Рассмотрим участок с 10 станциями и 36 вагонопотоками между несмежными станциями в одном направлении. Исходные данные на рис. 8 содержат:

на главной диагонали затраты времени в часах на переработку одного вагона на сортировочной станции;

над главной диагональю вагонопотоки, то есть количество вагонов в течении периода планирования с грузами, адресованных между двумя несмежными станциями;

под главной диагональю часы вагононакопления, то есть затраты вагоночасов на формирование прямого маршрута без переработки поездов на промежуточных станциях.

$$\begin{pmatrix}
 0 & 0 & 100 & 120 & 110 & 80 & 90 & 60 & 150 & 50 \\
 0 & 4 & 0 & 50 & 100 & 40 & 70 & 90 & 80 & 100 \\
 600 & 0 & 3 & 0 & 60 & 30 & 50 & 20 & 80 & 60 \\
 600 & 550 & 0 & 5 & 0 & 10 & 50 & 70 & 20 & 90 \\
 600 & 550 & 580 & 0 & 4 & 0 & 60 & 100 & 40 & 30 \\
 600 & 550 & 580 & 650 & 0 & 3 & 0 & 70 & 50 & 100 \\
 600 & 550 & 580 & 650 & 520 & 0 & 3 & 0 & 20 & 40 \\
 600 & 550 & 580 & 650 & 520 & 630 & 0 & 4 & 0 & 60 \\
 600 & 550 & 580 & 650 & 520 & 630 & 570 & 0 & 4 & 0 \\
 600 & 550 & 580 & 650 & 520 & 630 & 570 & 610 & 0 & 0
 \end{pmatrix}$$

Рис. 8. Исходные данные к примеру 4.

Требуется составить план формирования прямых маршрутов, минимизирующий общие затраты вагоночасов.

Сформулированная задача является частным случаем задачи развития бесконтурной транспортной сети.

Результаты решения задачи показаны на рис. 9. Прерывистые линии соответствуют девяти прямым маршрутам, выбранным из 36 возможных.. Минимальное значение целевой функции 11150 вагоночасов. Оптимальный план обладает тем свойством, что любой из оставшихся 27 вагонопотоков имеет не более одного переформирования на промежуточных станциях.

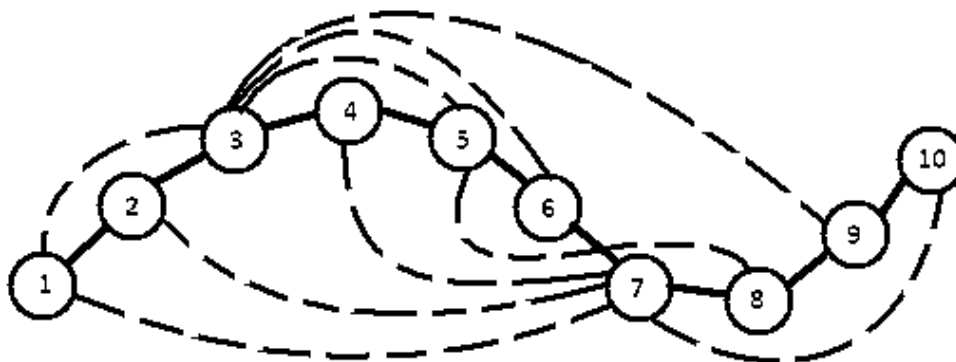


Рис. 9. Результаты решения задачи о плане формирования грузовых поездов.

Заключение

Вычислительные эксперименты показали, что предложенный в работе приближенный алгоритм оптимизации развития транспортной сети со случайными потоками продуктов дает результаты, близкие к нижней оценке целевой функции.

Выполнение значительного числа реализаций при статистическом моделировании потоков продуктов накладывает жесткие требования на время расчета одной реализации.

В силу специфики задачи одним из эффективных подходов к сокращению времени расчета явилось «покачивание целевой функции», то есть случайное варьирование ее коэффициентов, которое во многих случаях позволяет получать целочисленные оптимальные решения без применения МВГ.

В тех случаях, когда оптимальное целочисленное решение не попадает в одну из крайних точек линейной оболочки множества допустимых решений, результат достигается методом ветвей и границ.

В силу сказанного является актуальным иерархическое использование возможностей многоядерных вычислительных систем и сетей для набора необходимого количества реализаций, для применения метода ветвей и границ при каждой реализации и, наконец, при выполнении модифицированного симплекс-метода внутри. На данном этапе работы показано, что распараллеливание поиска разрешающего столбца в модифицированном симплекс-методе дает существенное убыстрение расчета.

Литература.

1. Мардашова П.Г., Эпштейн Г.Л. Задача развития транспортной сети с многопродуктовыми потоками. Труды Четвертой Международной конференции «Параллельные вычисления и задачи управления» PACO '2008. М.: 2008. стр. 1096 – 1102.
2. Гоманков Ф.С. Технология и организация перевозок на железнодорожном транспорте: Учебник для вузов. – М.: Транспорт, 1994. – 208 с.