© 2012 г. А.А. СТЯГОВ, аспирант
Г.Л. ЭПШТЕЙН, канд. техн. наук
(МИИТ, Москва)

О ВЫБОРЕ МЕТОДОВ ОПТИМИЗАЦИИ РАЗВИТИЯ ТРАНСПОРТНОЙ СЕТИ В МНОГОЯДЕРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

Описано применение многоядерных вычислительных систем для решения задачи развития транспортной сети. **Ключевые слова**: транспортная сеть, оптимизация, симплекс-метод, метод Ньютона, параллельные вычисления.

THE CHOICE OF METHODS OF OPTIMIZATION FOR DEVELOPMENT OF TRANSPORT NETWORK IN MULTI-CORE COMPUTING ENVIRONMENTS PACO'2012 / A.A. Styagov, G.L. Epshteyn (MIIT, Obrazcova str., 9, Moscow 127994, Russia. E-mail: Styagov.Al@gmail.com, egl413@gmail.com). Describes the use of multi-core computing systems to solve the problem of transportation network. K**ey words:** transport network, optimization, simplex-method, Newton's method, parallel computations.

1. Введение

Задача оптимизации развития транспортной сети путем добавления новых ребер включает в себя целевую функцию, учитывающую затраты на создание новых ребер и транспортировку продуктовых потоков, условия баланса в вершинах сети, ограничения потоков по ребрам и транзитных потоков через вершины, ограничения на увеличение локальных степеней вершин [1], [2]. В случае фиксированных затрат на создание нового ребра указанная проблема сводится к задаче линейного программирования с включением ряда бинарных переменных.

Основные трудности этой задачи связаны с быстрым ростом количества переменных и ограничений в зависимости от количества вершин и продуктовых потоков. Другим усложнением является наличие бинарных переменных и быстрый рост их количества с увеличением числа вершин сети. Так, для наиболее тяжелого случая сети в виде линейной цепочки рост числа ограничений и общего количества переменных представлен в таблице 1.

Таблица 1. Зависимость параметров задачи от числа вершин.

Количество вершин	Количество бинарных переменных	Ограничения	Переменные
5	6	31	52
8	21	203	350
10	36	486	852
12	55	990	1760
20	171	7296	13452

Поэтому для решения практических задач оптимизации развития сети необходимо выбрать метод, допускающий эффективное применение параллельных вычислений и обеспечивающий бинарность части переменных. Ниже приведены результаты сравнительного исследования двух методов получения нижних оценок оптимального значения целевой функции и способа достижения бинарности переменных с помощью случайных вариаций коэффициентов линейной целевой функции.

2. Общая схема решения задачи оптимизации развития сети

Блок-схема решения задачи добавления новых ребер транспортной сети представлена на рис. 1.

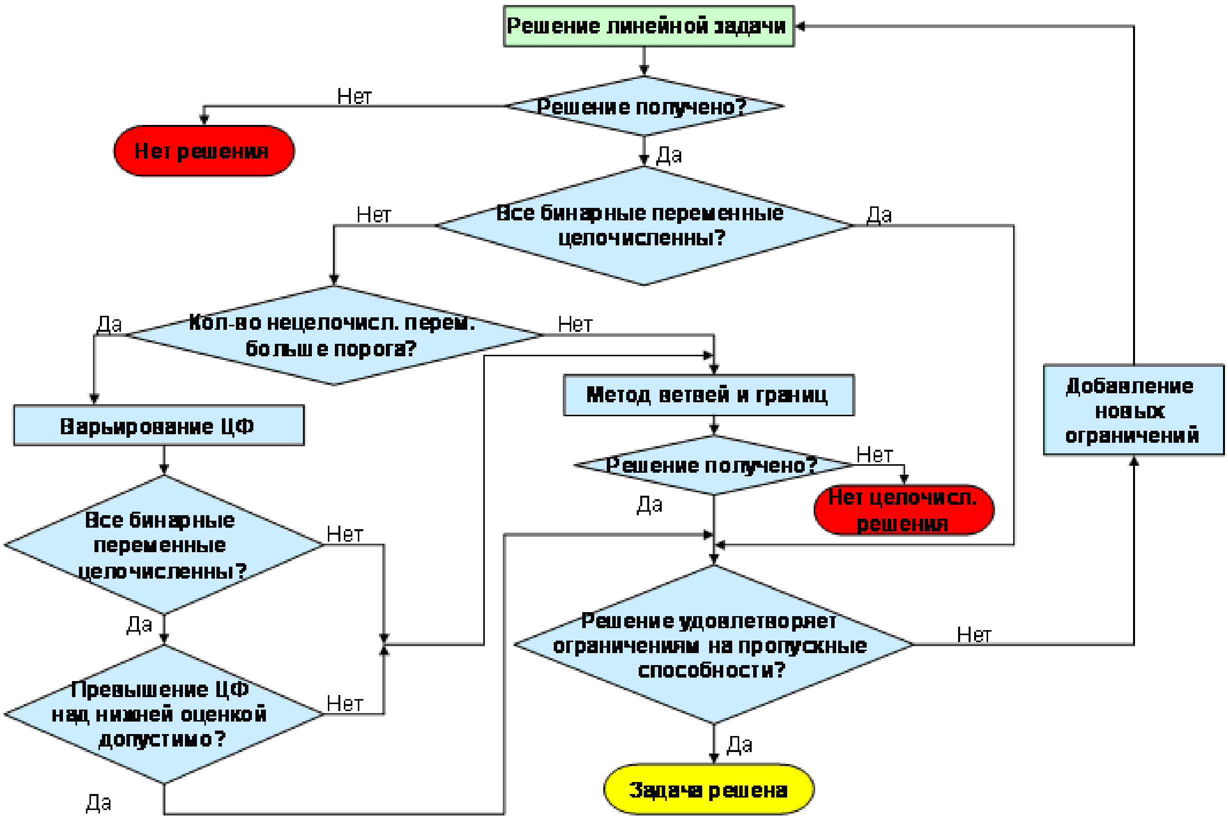


Рис.1. Блок-схема решения задачи развития транспортной сети.

Общий алгоритм решения задачи основан на последовательном учете требований целочисленности и дополнительных ограничений. Сначала решается непрерывная линейная задача. Оптимальное значение целевой функции этой задачи дает нижнюю оценку целевой функции исходной частично целочисленной задачи.

Если в оптимальном решении линейной задачи количество бинарных переменных не превышает заданного порога, то для получения целочисленности применяется метод ветвей и границ (МВГ). Если же количество нецелочисленных значений бинарных переменных превышает указанный порог, то предпринимается попытка получить целочисленные значения с помощью варьирования (покачивания) целевой функции.

При удачном исходе варьирования целевой функции проверяются дополнительные ограничения, например, на пропускные способности ребер и вершин или на локальные степени. Если решение не удовлетворяет дополнительным ограничениям, то повторно решается линейная задача.

В тех случаях, когда заданное число вариаций коэффициентов целевой функции не дает положительного результата, делается попытка применить МВГ при любом количестве бинарных переменных с нецелочисленными значениями.

Таким образом, либо будет найдено оптимальное решение, удовлетворяющее всем ограничениям, включая и условия бинарности части переменных, либо будет установлено отсутствие допустимых решений.

3. Нижняя оценка оптимального решения

3.1. Модифицированный симплекс-метод

Исследования, выполненные для поставленной задачи, показали, что наибольший эффект от использования многоядерных вычислительных систем для параллельных вычислений в симплекс-методе можно получить 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 (1)

$$P = c_B^T\beta - (\pi^T N - c_N^T)x_N \quad . \tag{1}$$

В (1) 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:

1) флаг ожидания сигнала из внешней среды на выполнение итерации
2) запуск процедуры FindDj, осуществляющей поиск оценок
3) 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.

При выполнении тестовых расчетов отмечен быстрый рост количества итераций при увеличении размера задачи (таблица 2).

Таким образом, при значительном росте числа итераций эффективность параллельных вычислений в симплекс-методе принципиально ограничена, а проблемы сохранения численной устойчивости и перепостроения обратной матрицы создают большие «накладные расходы» при увеличении размера задачи.

Таблица 2. Зависимость количества итераций от размера задачи.

Ограничения	Переменные	Заполненность %	Итерации поиска допустимого базиса	Итерации оптимизации
31	52	5.65	32	8
203	350	1.00	204	129
322	560	0.64	323	292
486	852	0.44	487	593
990	1760	0.20	991	2238
2366	4277	0.09	2367	12819
7296	13452	0.03	7297	115079

3.2. Обобщенный метод Ньютона

В работе [3] описан алгоритм обобщенного метода Ньютона для решения задачи

(2) $$c^T x \to \min, \ Ax = b, \ x > 0 .$$

Алгоритм основан на вычислении проекции произвольной точки области допустимых решений на множество оптимальных решений (2).

Этот метод более приспособлен для параллельных вычислений, чем симплекс-метод. Эффективность параллельных расчетов указанным методом повышается с увеличением размеров задачи, так как матрица Гессе имеет тем более доминирующую диагональ, чем больше количество переменных. А это приводит к уменьшению числа итераций предобусловленного метода сопряженных градиентов.

Основной недостаток обобщенного метода Ньютона состоит в высокой чувствительности количества внутренних итераций к начальному выбору множителей Лагранжа. Существенное влияние на количество итераций алгоритма оказывает значение множителя Лагранжа при целевой функции β .

Нижняя граница множителя Лагранжа β_* задается формулой

(3) $$\beta_* = \begin{cases} \max\limits_{i\in\sigma} \dfrac{\left(x_d + A_d^T \left(A_l A_l^T\right)^{-1} A_l \left(x_l^* - x_l\right)\right)_i}{\left(v_d^*\right)_i}, & \sigma \neq \varnothing, \\ \gamma > -\infty, & \sigma = \varnothing. \end{cases}$$

В (3) обозначены:

x_l^* - вектор ненулевых координат в проекции некоторой исходной точки на множество оптимальных решений,

x_l - вектор координат некоторой исходной точки, имеющих ненулевые координаты в проекции на множество оптимальных решений, l – количество ненулевых координат в проекции, $d = n - l$,

x_d - вектор координат некоторой исходной точки, имеющих нулевые координаты в проекции на множество оптимальных решений,

A_l, A_d - подматрицы матрицы ограничений прямой задачи линейного программирования (1), соответствующие переменным x_l и x_d,

v_d^* - вектор разностей между правыми и левыми частями ограничений двойственной задачи, соответствующим ненулевым координатам проекции x_l^*,

(4) $$v_d^* = c - A_d^T u^*,$$

σ - множество индексов i ненулевых координат вектора v_d^*,

u^* - оптимальное решение двойственной задачи,

γ - произвольное число.

Из (3) следует, что при малых значениях некоторых ненулевых координат v_d^* нижняя граница множителя Лагранжа β окажется очень большой, что затрудняет сходимость метода Ньютона.

Как показали расчеты тестовых примеров при одинаковых начальных значениях множителей Лагранжа, зависимость количества внутренних итераций от размера задачи имеет немонотонный характер (таблица 3). Это означает, что параметр β желательно выбирать индивидуально с учетом размера задачи и выбора начальной точки в пространстве переменных. Сравнивая таблицы 2 и 3, надо отметить, что обобщенный метод Ньютона требует значительно меньшего числа итераций, чем симплекс-метод.

Таблица 3. Зависимость количества внутренних итераций от размера задачи.

Ограничения	Переменные	Заполненность %	Количество внутренних итераций
31	52	5.65	16
203	350	1.00	355
322	560	0.64	343
486	852	0.44	207
990	1760	0.20	3852

Таким образом, практическое применение метода должно сопровождаться процедурой выбора начальных значений множителей Лагранжа.

4. Получение целочисленного решения

Оптимальное значение целевой функции, полученное с помощью симплекс-метода или обобщенного метода Ньютона, является нижней оценкой целевой функции задачи с бинарными переменными. По мере увеличения размера задачи растет и количество нецелочисленных значений бинарных переменных на этапе получения нижней оценки (таблица 4).

Таблица 4. Количество нецелочисленных значений бинарных переменных.

Ограничения	Переменные	Нецелочисленные бинарные переменные
486	852	20
990	1760	21
2366	4277	39
7296	13452	53

Наиболее надёжный подход к решению частично целочисленных задач заключается в применении метода ветвей и границ. Преимуществом этого метода является получение точного оптимального решения поставленной задачи, если таковое решение существует. Еще одним важным преимуществом этого метода является естественное распараллеливание расчета подзадач.

Существенный же минус – значительное время расчётов, ввиду больших размеров деревьев метода ветвей и границ по сравнению с возможностями многоядерной системы (например, 2170 вершин для задачи с 10 узлами и 36 потоками).

С целью сокращения времени решения задачи разветвление дерева подзадач выполнялось не введением дополнительных ограничений, а изменением целевой функции с учетом того, что в постановке задачи минимизации указана стоимость каждой бинарной переменной. Чтобы добиться исключения этой переменной из оптимального решения ее стоимость полагалась очень большой, а для гарантированного включения в оптимальное решение стоимость бинарной переменной берется нулевой. Разумеется, после получения решения выполняется правильный подсчет целевой функции. При указанном подходе для любой подзадачи текущее решение было допустимым и требовалось лишь очень малое число итераций для минимизации целевой функции.

Еще один способ достижения целочисленного решения оказался весьма эффективным, как с точки зрения параллельных вычислений, так и для уменьшения общего объема вычислительной работы. Этот способ опирается на предположение, что на границе области допустимых решений находятся целочисленные оптимальные решения, которые можно обнаружить «покачиванием» целевой функции, как показано на рисунках 2 и 3.

Вариация целевой функции выполняется путем случайных изменений ее коэффициентов. Максимальное количество реализаций случайных вариаций N_R и пределы изменения коэффициентов можно задать заранее. Тогда в зависимости от количества нецелочисленных значений k_{noint} и допустимого количества подзадач в методе ветвей и границ k_{MBB} можно оценить целесообразность применения вариации целевой функции вместо метода ветвей и границ с помощью неравенства

(5) $$k_{noint} > \log_2\left(k_{MBB}+2\right)-1.$$

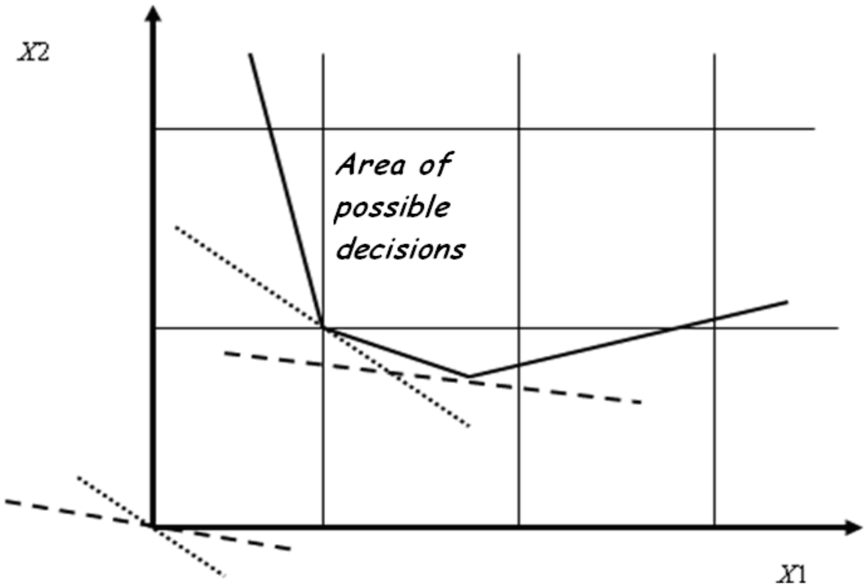


Рис.2. Вариация целевой функции для достижения целочисленного решения. Пунктирная прямая – исходное положение прямой уровня целевой функции. Точечная прямая – проварьированное положение прямой уровня..

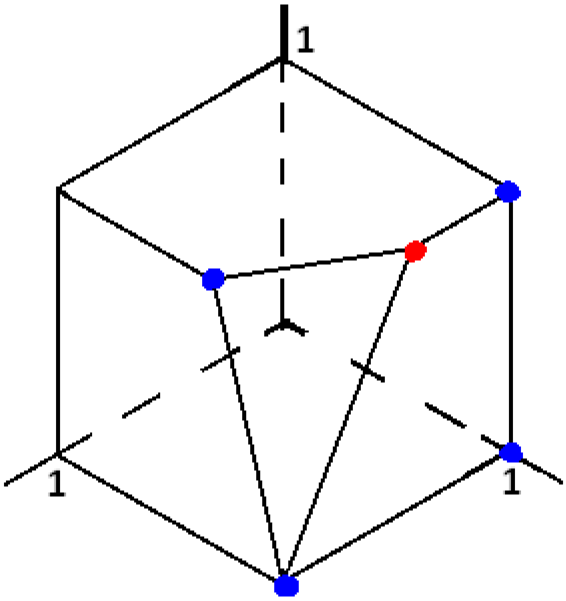
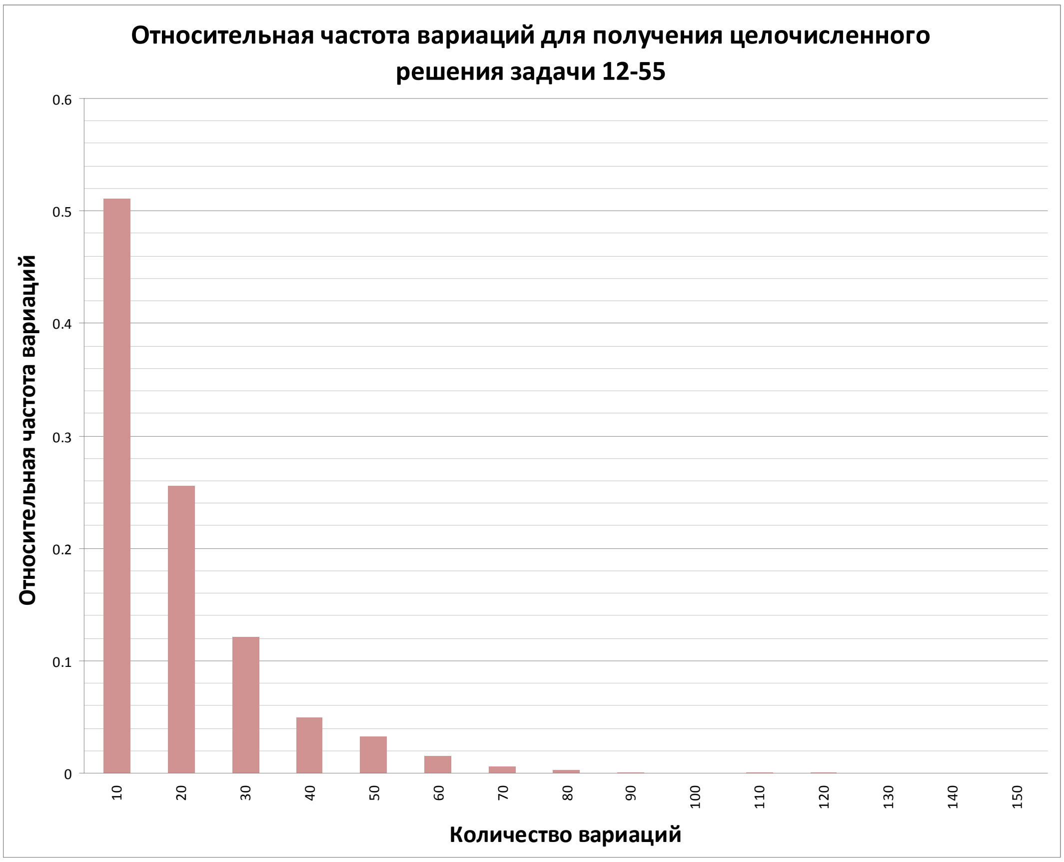


Рис.3. Пространственная иллюстрация варьирования целевой функции.
Красной точкой обозначено нецелочисленное оптимальное решение.

Вариация целевой функции может выполняться до первого достижения целочисленного решения со значением целевой функции, не превышающим нижнюю оценку более, чем на заданную константу. Константа определяется точностью исходных данных. Другая стратегия состоит в выполнении заданного числа вариаций с последующим отбором наилучшего целочисленного решения.

Заметим, что добавляя соответствующие ограничения в исходную задачу, можно добиться существования целочисленных точек на границе ОДР.

Вычислительный эксперимент с задачей 12 – 55 (12 вершин, 55 потоков, 990 ограничений, 1760 переменных) показал значительное сокращение длительности расчетов. Распределение количества испытаний до достижения целочисленного решения показано на рис. 4.



В таблице 5 приведено сравнение нижних оценок целевой функции и ее значений, полученных методом вариации. Все результаты оценивались по 1500 расчетам.

Таблица 5. Распределение относительной разности между ЦФ в найденном целочисленном решении и нижней оценкой ЦФ.

Разность менее	1%	2%	3%	4%	5%	6%
Количество	0	1167	173	128	30	2

.

СПИСОК ЛИТЕРАТУРЫ

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.: С. 1096 – 1102..
2. *Мардашова П.Г., Стягов А.А., Эпштейн Г.Л.* Оптимизация развития транспортной сети со случайными потоками. // Труды Пятой Международной конференции «Параллельные вычисления и задачи управления» PACO '2010. М.: 2010. С. 415 – 424.
3. *Гаранжа В.А., Голиков А.И., Евтушенко Ю.Г., Нгуен М.Х.* Параллельная реализация метода Ньютона для решения больших задач линейного программирования. Журнал вычислительной математики и математической физики, 2009, Т. 49, № 8, С. 1369 – 1384.