УДК 519.852.33

В.А. ВАСЯНИН, Л.П. УШАКОВА

БАЛАНСИРОВКА МАТРИЦЫ КОНТЕЙНЕРНЫХ ПОТОКОВ В ЗАДАЧЕ ПЕРЕВОЗКИ МЕЛКОПАРТИОННЫХ ГРУЗОВ

Аннотация. *Рассматриваются два способа балансировки матрицы контейнерных потоков при решении задачи перевозки мелкопартионных грузов в контейнерах. Необходимость балансировки возникает из-за неравенства суммы исходящих и входящих потоков контейнеров в узлах транспортной сети. Предложена математическая модель и алгоритм решения задачи развозки порожних контейнеров, которые могут быть использованы для балансировки матрицы контейнерных потоков и последующего решения задачи распределения и маршрутизации потоков груженых и порожних контейнеров. Проведен обзор и анализ современных методов и алгоритмов решения транспортной задачи. Экспериментально показано, что оптимальная балансировка по сравнению с симметричной балансировкой позволяет значительно сократить суммарные затраты на транспортировку и обработку порожних контейнеров (на сетях от 100 до 4000 узлов в 17 и 174 раза соответственно).*

Ключевые слова: *перевозка мелкопартионных грузов в контейнерах, транспортная сеть, модели и алгоритмы решения транспортной задачи.*

Введение

При перевозке мелкопартионных грузов в жесткой таре (контейнерах) в многопродуктовой транспортной сети возникает задача балансировки матрицы потоков. Под многопродуктовой сетью понимается сеть, в которой каждый узел обменивается не взаимозаменяемыми грузами со всеми остальными узлами. Поскольку не для всех узлов сети выполняется условие баланса — равенства суммы исходящих и входящих контейнеров, актуальной становится задача развозки порожних контейнеров (РПК). Часто на практике балансировка матрицы потоков для каждой корреспондирующейся пары узлов осуществляется путем возврата порожних контейнеров в узел с их недостатком, т. е. имеет место симметричная балансировка, при которой учет рабочего парка контейнеров ведется одним из этих узлов. При наличии автоматизированной информационно-аналитической системы (АИАС) [1] управления процессами обработки и перевозки грузов выгоднее вести централизованный учет рабочего парка контейнеров и для балансировки потоков решать задачу оптимизации РПК. При этом появляется возможность значительного снижения приведенных затрат на обработку и перевозку потоков порожних контейнеров и оперативного управления их перераспределением при колебании нагрузок в сети.

Целью работы является построение математической модели, выбор методов и алгоритмов для решения задачи оптимизации РПК и экспериментальное сравнение методов симметричной и оптимальной балансировки матрицы потоков на численных примерах.

1. Математическая модель задачи развозки порожних контейнеров

Пусть $G(\mathrm{N},\mathrm{P})$ — неориентированная транспортная сеть с множеством узлов N, $n = |\mathrm{N}|$ и множеством дуг P, $p = |\mathrm{P}|$, $|\cdot|$ — знак мощности множества. Узлы сети соответствуют пунктам отправления, получения и перегрузки груженых и порожних контейнеров, а дуги — физическим отрезкам линий связи (например, по железным и автомобильным дорогам, морским путям и пр.), соединяющим два любых узла из множества N так, что между рассматриваемыми узлами на данном отрезке нет больше ни одного узла из N.

Потоки груженых контейнеров заданы матрицей $A = \| a_{ij} \|_{n\times n}$ среднесуточных значений, статистически рассчитанных для конкретных периодов планирования перевозок (например, в зависимости от сезонных колебаний величины потоков). Потоки a_{ij} из источников i в стоки j, $i, j = \overline{1,n}$ должны перевозиться в транспортных средствах с заданной периодичностью. Матрица A не сбалансирована, поскольку для большинства узлов не выполняется условие

$$\Delta_i = \sum_{j=1}^{n}(a_{ij} - a_{ji}) = 0,\ i = \overline{1,n}. \qquad (1)$$

Первый способ балансировки предусматривает построение симметричной матрицы $A^c = \| a_{ij}^c \|_{n\times n}$, каждый элемент которой определяется из выражения $a_{ij}^c = \max\{a_{ij}, a_{ji}\}$, $i, j = \overline{1,n}$. Такое построение приводит к дополнительным затратам на перевозку и погрузку/выгрузку порожних контейнеров, однако, упрощает организацию их обработки и отправки. Второй способ балансировки заключается в решении транспортной задачи для развозки порожних контейнеров.

Разделим все множество узлов, для которых не выполняется условие (1), на два подмножества A и B так, чтобы выполнялось $\Delta_i < 0$, $i \in \mathrm{A}$, $\Delta_i > 0$, $i \in \mathrm{B}$ и $\mathrm{A} \cup \mathrm{B} \subseteq \mathrm{N}$. Тогда в терминах транспортной задачи множество B является множеством потребителей порожних контейнеров, а A — множеством их поставщиков. Образуем вектор потребителей b_j, $j = \overline{1,k}$ с объемами потребления $b_j = \Delta_\xi$, для $\forall\ \xi \in \mathrm{B}$ и вектор поставщиков a_i, $i = \overline{1,l}$, с объемами предложения $a_i = |\Delta_\xi|$, для $\forall\ \xi \in \mathrm{A}$. Где k и l получены счетным перечислением множеств B и A соответственно. Векторам a_i и b_j поставим во взаимно однозначное соответствие векторы номеров узлов α_i и β_j из A и B. Сформулируем транспортную задачу. Требуется минимизировать

$$\sum_{i=1}^{l}\sum_{j=1}^{k} f_{ij}(x_{ij}, d_{ij}), \tag{2}$$

при ограничениях

$$\sum_{j=1}^{k} x_{ij} = a_i,\ i = \overline{1,l}, \tag{3}$$

$$\sum_{i=1}^{l} x_{ij} = b_j,\ j = \overline{1,k}, \tag{4}$$

$$\sum_{i=1}^{l} a_i = \sum_{j=1}^{k} b_j \quad , \tag{5}$$

где $x_{ij} \geq 0$ — число порожних контейнеров, отправляемых i-м поставщиком j-му потребителю; d_{ij} — расстояние перевозки от i до j; $f_{ij}(\cdot)$ — кусочно-выпуклые нелинейные функции транспортных затрат, зависящие в общем случае от перевозимого объема, расстояния, типа транспортного средства, выделенного для перевозки, маршрута перевозки, скорости движения и других факторов. Учет всех перечисленных факторов при решении задачи (2)–(5) не представляется возможным, поскольку перевозка порожних контейнеров должна выполняться вместе с гружеными. Маршруты перевозки определяются при решении отдельной задачи распределения и маршрутизации потоков [2]. Поэтому предположим, что на этапе балансировки матрицы контейнерных потоков транспортные затраты на перевозку порожних контейнеров пропорциональны объему и расстоянию. Такое допущение позволяет не учитывать нелинейность функций $f_{ij}(\cdot)$ и интерпретировать (2)–(5) как задачу минимизации «пробега» порожних контейнеров, подставив вместо $f_{ij}(\cdot)$ длины кратчайших путей $\widetilde{D}^* = \left\| \widetilde{d}_{ij}^* \right\|_{l\times k}$ из i в j. Предполагается, что предварительно на сети G рассчитаны кратчайшие пути $D^* = \left\| d_{ij}^* \right\|_{n\times n}$, а $\widetilde{d}_{ij}^* = d_{\alpha_i\beta_j}^*$, $i = \overline{1,l}$, $j = \overline{1,k}$. Отметим, что в (2) расстояния d_{ij} могут отличаться от расстояний по кратчайшим путям d_{ij}^*.

Задача преобразуется к линейному виду: минимизировать

$$\sum_{i=1}^{l}\sum_{j=1}^{k} \widetilde{d}_{ij}^* x_{ij} \tag{6}$$

при ограничениях (3)–(5). После решения задачи, балансировка осуществляется следующим образом: $a_{ij}^o = a_{ij}$, $i, j = \overline{1,n}$; $a_{\alpha_i\beta_j}^o = a_{\alpha_i\beta_j} + x_{ij}$, $i = \overline{1,l}$, $j = \overline{1,k}$, где a_{ij}^o — элементы сбалансированной матрицы $A^o = \left\| a_{ij}^o \right\|_{n\times n}$.

Пусть имеются операторы $\varphi^o : X = \| x_{ij} \|_{l \times k} \Rightarrow X^o = \| x_{ij}^o \|_{n \times n}$, где $x_{ij}^o = x_{\alpha_i \beta_j}$, $i, j = \overline{1, n}$; $\varphi^c : A = \| a_{ij} \|_{n \times n} \Rightarrow X^c = \| x_{ij}^c \|_{n \times n}$, где x_{ij}^c определяются из условия: если $a_{ij} \ge a_{ji}$, то $x_{ji}^c = a_{ij} - a_{ji}$, иначе $x_{ij}^c = a_{ji} - a_{ij}$, $i = \overline{1, n-1}$, $j = \overline{i+1, n}$.

Сравним способы балансировки с точки зрения затрат на перевозку и обработку порожних контейнеров. Поскольку на этапе решения задачи РПК, в силу сделанных допущений, говорить о реальных затратах не приходится, будем сравнивать два способа балансировки, используя следующие выражения для определения транспортных и погрузочно-разгрузочных затрат:

$$Z_{тр}^c = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij}^* (x_{ij}^c, d_{ij}^*),\ Z_{погр}^c = f_{конт}^* \left(2 \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}^c \right), \tag{7}$$

$$Z_{тр}^o = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij}^* (x_{ij}^o, d_{ij}^*),\ Z_{погр}^o = f_{конт}^* \left(2 \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}^o \right), \tag{8}$$

где $Z_{тр}^c$, $Z_{погр}^c$, $Z_{тр}^o$, $Z_{погр}^o$ — соответственно транспортные и погрузочно-разгрузочные затраты при симметричной балансировке и балансировке, основанной на оптимизации развозки порожних контейнеров; $f_{ij}^*(\cdot)$ — некоторые функции транспортных затрат, зависящие от объема и расстояния перевозки; $f_{конт}^*(\cdot)$ — функция затрат от объема погружаемых и выгружаемых контейнеров. В качестве $f_{ij}^*(\cdot)$ и $f_{конт}^*(\cdot)$ могут выступать функции среднегодовых приведенных затрат, учитывающие динамику перевозок и зависящие от рабочего парка транспортных средств и контейнеров, необходимых для нормального функционирования сети при колебаниях нагрузок в сети (см. разд. 3).

Величина снижения затрат

$$\Delta Z = \Delta Z_{тр} + \Delta Z_{погр} = (Z_{тр}^c - Z_{тр}^o) + (Z_{погр}^c - Z_{погр}^o) \tag{9}$$

может служить оценкой оптимального плана развозки порожних контейнеров.

2. Алгоритм решения задачи развозки порожних контейнеров. Обзор и выбор методов и алгоритмов решения транспортной задачи

При реализации алгоритма решения задачи нет необходимости создавать и хранить матрицы X^c и X^o, поэтому формулы (7) и (8) можно записать в виде

$$Z_{тр}^{c} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f_{ij}^{*}(abs(a_{ij} - a_{ji}), d_{ij}^{*}), \; Z_{погр}^{c} = f_{конт}^{*}\left(\sum_{i=1}^{n} \sum_{j=1}^{n} abs(a_{ij} - a_{ji}) \right), \quad (10)$$

$$Z_{тр}^{o} = \sum_{i=1}^{l} \sum_{j=1}^{k} f_{ij}^{*}(x_{ij}, \tilde{d}_{ij}^{*}), \qquad Z_{погр}^{o} = f_{конт}^{*}\left(2\sum_{i=1}^{l} \sum_{j=1}^{k} x_{ij} \right), \quad (11)$$

где abs означает абсолютную величину.

Приведем укрупненный алгоритм решения задачи РПК.

1. На сети G найти матрицу кратчайших путей $D^{*} = \| d_{ij}^{*} \|_{n \times n}$.

2. $k \leftarrow 0$; $l \leftarrow 0$.

3. Для $\{ i \mid i = \overline{1, n} \}$ выполнить пп. 4–11.

4. $SO \leftarrow 0$; $SI \leftarrow 0$.

5. Для $\{ j \mid j = \overline{1, n}, j \neq i \}$ выполнить пп. 6–7.

6. $SO \leftarrow SO + a_{ij}$; $SI \leftarrow SI + a_{ji}$.

7. Перейти к п. 5. *** Конец цикла по j

8. Если $SO > SI$, то перейти к п. 9, иначе перейти к п. 10.

9. $k \leftarrow k + 1$; $b_{k} \leftarrow (SO - SI)$; $\beta_{k} \leftarrow i$; перейти к п. 12.

10. Если $SO < SI$, то перейти к п. 11, иначе перейти к п. 12.

11. $l \leftarrow l + 1$; $a_{l} \leftarrow (SI - SO)$; $\alpha_{l} \leftarrow i$.

12. Перейти к п. 3. *** Конец цикла по i

13. По формулам (10) определить $Z_{тр}^{c}$ и $Z_{погр}^{c}$.

14. Для $\{ i, j \mid i = \overline{1, l}, \; j = \overline{1, k} \}$ выполнить пп. 15–16.

15. $\tilde{d}_{ij}^{*} \leftarrow d_{\alpha_i \beta_j}^{*}$.

16. Перейти к п. 14. *** Конец циклов по j и i

17. Решить транспортную задачу (6), (3)–(5).

18. По формулам (11) определить $Z_{тр}^{o}$ и $Z_{погр}^{o}$.

19. Для $\{ i, j \mid i = \overline{1, l}, \; j = \overline{1, k}, x_{ij} \neq 0 \}$ выполнить пп. 20–21.

20. $a_{\alpha_i \beta_j} \leftarrow a_{\alpha_i \beta_j} + x_{ij}$.

21. Перейти к п. 19. *** Конец циклов по j и i

22. По формуле (9) получить оценку ΔZ оптимального плана распределения порожних контейнеров.

23. Конец алгоритма.

Оценивая временную сложность приведенного алгоритма, отметим, что наиболее трудоемкими действиями в нем являются нахождение кратчайших путей и решение транспортной задачи (ТЗ). Для нахождения всех кратчайших путей в сети G например с помощью алгоритма Флойда (Floyd R.W., 1962 г.)

потребуется $O(n^3)$ времени. Временная сложность решения транспортной задачи зависит от применяемых методов и алгоритмов.

Впервые транспортная задача в строгой математической постановке была сформулирована Хичкоком [3] и исследована Купмансом [4]. Первым точным методом решения классической транспортной задачи был метод потенциалов, предложенный Л.В. Канторовичем и М.К. Гавуриным [5]. С тех пор появилось множество работ, связанных с постановкой и решением транспортных задач. Среди них можно выделить [6, 7], посвященные классификации и детальному рассмотрению существующих методов и алгоритмов решения транспортной задачи.

Большинство методов решения классической транспортной задачи основано на решении двойственной задачи — найти $\max\left(\sum_{i=1}^{l} a_i u_i + \sum_{j=1}^{k} b_j v_j\right)$ при $u_i + v_j \le c_{ij}$, $i = \overline{1,l}$, $j = \overline{1,k}$, где u_i, v_j не ограничены в знаке, а c_{ij}, — неотрицательные коэффициенты целевой функции прямой задачи: найти $\min \sum_{i=1}^{l} \sum_{j=1}^{k} c_{ij} x_{ij}$ при ограничениях (3)–(5).

В методе потенциалов для решения двойственной задачи по существу используется прямой симплекс метод (Dantzig G.B., 1947 г.) с учетом специфики транспортной задачи, а двойственные переменные (симплекс-множители) u_i, v_j являются потенциалами поставщиков и потребителей. Начальное допустимое базисное решение определяется методами «северо-западного угла», «минимального элемента», методом Фогеля и т.п. Для этого решения составляется система из $l + k - 1$ уравнений $u_i + v_j = c_{ij}$ и определяются начальные значения u_i, v_j и оценки для небазисных переменных $\tilde{c}_{ij} = u_i + v_j - c_{ij}$. Далее решение итеративно улучшается обычным путем: введением в базис новой переменной из числа небазисных, имеющих наибольшую положительную оценку $\tilde{c}_{ij}$ (при решении прямой задачи на минимум); выведением из базиса существующей переменной по условиям допустимости в симплекс-методе. Для этого в транспортной таблице для вводимой переменной строится замкнутый цикл из числа допустимых для вывода базисных переменных и выводится одна, имеющая наименьшее значение, которое и присваивается переменной, вводимой в базис. При этом пересчитываются значения всех базисных переменных так, чтобы не нарушались ограничения по a_i и b_j. В условиях допустимости в симплекс-методе знаменатель всегда равен единице, т. к. все коэффициенты в ограничениях на ресурсы для транспортной задачи равны нулю или единице. Итеративный процесс заканчивается, когда для всех небазисных переменных выполняются условия оптимальности симплекс-метода (для задачи на минимум, — все значения $\tilde{c}_{ij}$ должны быть неположительными, если оценки рассчитываются как $\tilde{c}_{ij} = u_i + v_j - c_{ij}$).

Временная сложность метода потенциалов зависит от эффективности реализации алгоритмов (матричных или сетевых) выполнения его основных шагов: нахождения начального допустимого базисного решения (начального опорного плана); определения переменной вводимой в базис и проверки на оптимальность текущего решения; определения переменной выводимой из базиса и пересчета новых значений базисных переменных. В алгоритмах могут использоваться различные способы представления данных — абстрактные типы данных (АТД): матричные, списочные, очереди с приоритетами, древовидные и др. Так, например, в сетевой реализации алгоритмов текущее базисное решение представляется остовным деревом и используются эффективные структуры данных и процедуры для представления и работы с графами. Наиболее трудоемкой операцией в методе потенциалов является определение оценок $\tilde{c}_{ij}$, т.к. сначала надо выполнить $l+k-1$ операций для вычисления потенциалов базисных переменных, а затем рассчитать потенциалы и оценки $\tilde{c}_{ij}$ для $l \times k-(l+k-1)$ небазисных переменных. При самом простом подходе для этого понадобится просмотр всех элементов транспортной таблицы (одновременно можно найти максимальный $\tilde{c}_{ij}$ для переменной, вводимой в базис, или определить оптимальность текущего решения). Трудоемкость шага выведения переменной из базиса зависит от способа представления текущего базисного решения (остовного дерева) в оперативной памяти компьютера и реализации процедур просмотра и пересчета элементов базиса. Эти процедуры могут быть очень эффективно реализованы при использовании современных АТД, поэтому временная сложность выполнения шага для $l+k$ циклически связанных значений невелика и нуль сравнима по сравнению с операциями расчета $\tilde{c}_{ij}$. Число итераций алгоритма для нахождения оптимального решения трудно выразить только через l и k, поскольку оно зависит от величины значений a_i и b_j, вырожденности базисных решений, применяемых структур данных задачи и процедур работы с ними. Однако ясно, что увеличить скорость работы прямого алгоритма симплекс метода можно за счет сокращения операций расчета $\tilde{c}_{ij}$ и проверки условия оптимальности текущего решения.

Широкое применение при решении транспортных задач получил венгерский метод [8]. В его основе лежит построение максимальных потоков через транспортную сеть с частично разрешёнными коммуникациями и последующее сокращение невязок. Венгерский метод решения использует схему прямо-двойственного алгоритма, в которой допустимое двойственное решение итеративно улучшается за счет решения задачи, двойственной к ограниченной прямой задаче.

Всегда можно получить допустимое решение двойственной задачи: $u_i = 0$, $v_j = \min c_{ij}$, $i=\overline{1,l}$, $j=\overline{1,k}$. Пусть пара индексов $ij \in \Omega$, если для них выполняется условие $u_i + v_j = c_{ij}$. Допустимые пары индексов $ij \in \{\Omega\}$ соответствуют допустимым столбцам в транспортной таблице для матричной реализации венгерского метода.

Запишем ограниченную прямую задачу, введя искусственные переменные $x_i \geq 0$, $i = \overline{1, l+k}$: найти

$\min \sum_{i=1}^{l+k} x_i$ при $\sum_j x_{ij} + x_i = a_i$, $i = \overline{1, l}$, $\sum_i x_{ij} + x_{l+j} = b_j$, $j = \overline{1, k}$, причем $x_{ij} \geq 0$, если $ij \in \{\Omega\}$ и $x_{ij} = 0$, если $ij \notin \{\Omega\}$. Эту задачу без переменных x_i можно записать как

$\max \sum_{ij \in \Omega} x_{ij}$ при $\sum_j x_{ij} \leq a_i$, $i = \overline{1, l}$, $\sum_i x_{ij} \leq b_j$, $j = \overline{1, k}$ при тех же ограничениях на x_{ij}.

Полученную задачу можно интерпретировать как задачу нахождения максимального потока в сети по допустимым парам $ij \in \{\Omega\}$ при оценке минимальной стоимости в ограниченной прямой задаче, равной

$$\Delta = \sum_{i=1}^{l} a_i + \sum_{j=1}^{k} b_j - 2\sum_{ij \in \Omega} x_{ij},$$

и использовать для ее решения эффективные сетевые алгоритмы.

Сетевую постановку транспортной задачи представляют в виде графа $\mathrm{G}(s, t, \mathrm{L}, \mathrm{K}, \mathrm{P})$ с обобщенным узлом отправления s, обобщенным узлом потребления t, множеством L, $|\mathrm{L}| = l$ узлов отправления, множеством K, $|\mathrm{K}| = k$ узлов потребления и множеством ориентированных дуг P. Узел s связан с узлами отправления исходящими дугами p_{si} с пропускными способностями $w_{si} = a_i$, $i = \overline{1, l}$. Узлы потребления связаны с узлом t входящими дугами p_{jt} с пропускными способностями $w_{jt} = b_j$, $j = \overline{1, k}$. Узлы отправления и потребления соединены направленными дугами с неограниченными пропускными способностями $w_{ij} = \infty$ только для пар индексов узлов $ij \in \{\Omega\}$.

Очевидно, что сложность решения транспортной задачи прямо-двойственным алгоритмом зависит от числа итераций улучшения значений двойственных переменных при решении ограниченной прямой задачи, которую нужно решать до тех пор, пока величина невязки Δ не станет равной нулю и через сеть будет пропущен максимальный поток величины $\sum_{i=1}^{l} a_i = \sum_{j=1}^{k} b_j$. На каждой итерации необходимо находить максимальный поток, а число итераций опять-таки трудно представить только через l и k.

Как известно, транспортная задача является частным случаем задачи о потоке минимальной стоимости, в которой все дуги допустимы и имеют стоимости и пропускные способности. Для любой конкретной задачи о потоке минимальной стоимости можно построить соответствующую ей транспортную задачу, решения которых будут совпадать. С этой точки зрения сложность решения транспортной задачи можно сопоставить со сложностью решения задачи о потоке минимальной стоимости.

Учитывая практическую важность транспортной задачи, методы и алгоритмы ее решения активно развивались в прошлом столетии и продолжают совершенствоваться в настоящее время. В работах [9–12] были предложены эффективные методы и алгоритмы для решения классической транспортной задачи. В частности, в [9] рассматривается эффективный алгоритм, основанный на поразрядном сокращении невязок, с общей временной сложностью $O(kl^2m)$, где $m=\lceil \log_2 \min(a_{\max}, b_{\max}) \rceil$, $a_{\max}=\max a_i$, $i=\overline{1,l}$, $b_{\max}=\max b_j$, $j=\overline{1,k}$. Работы [10, 11] посвящены эффективным реализациям метода потенциалов для транспортной задачи в сетевой постановке, в которых экономно организованы процедуры вычисления потенциалов и пересчета базиса. Рекурсивный вариант венгерского метода, использующий технику цепных списков, предложен в работе [12]. Обзор с 1966 по 1978 гг., экспериментальное сравнение и практические советы по применению методов и алгоритмов решения транспортных задач можно найти в зарубежных работах [13–17].

В СССР большинство из указанных алгоритмов прошло экспериментальную апробацию на конкурсе «Транспорт-81», проведенном по инициативе и под руководством Л.В. Канторовича [18]. Как показали экспериментальные исследования, наилучшей оказалась программа, реализующая вариант метода потенциалов, предложенного в [10].

Несмотря на то, что транспортная задача отнесена к классу полиномиально разрешимых, при сравнении различных алгоритмов ее решения в практических случаях приходится полагаться на экспериментальные результаты. Известны патологические случаи, когда для решения транспортной задачи и задачи о потоке минимальной стоимости требуется экспоненциальное число итераций, казалось бы, эффективных алгоритмов [19, 20].

Дальнейшее развитие методов и алгоритмов решения транспортной задачи нашло отражение в работах [21–25]. В [21] на примере транспортной задачи сравнивается быстродействие теоретически полиномиального двойственного сетевого симплекс-алгоритма с прямым сетевым алгоритмом. Приводятся результаты экспериментального сравнения двойственного алгоритма с известными реализациями прямого алгоритма в пакете программ Netflo. Показано, что на сетях до 300 узлов и 10000 дуг алгоритмы сравнимы по быстродействию. При превышении указанных границ прямой алгоритм быстрее двойственного на 30–60%. В работах [22, 23] приводятся эффективные реализации двойственного сетевого симплекс алгоритма и строго полиномиального алгоритма для решения транспортной задачи. Вопросы постоптимального анализа решения транспортной задачи для плавающих цен рассматриваются в работе [24]. Предлагается универсальный алгоритм «Push-and-Pull», который в отличие от прямого и двойственного симплекс метода, метода «падающих камней» (stepping-stone) не требует введения дополнительных и искусственных переменных для получения начального решения и других операций, связанных с циклическим обновлением базисных решений (не допускает вырожденных решений). Сравнивается вычислительная эффективность предложенного алгоритма с алгоритмами в пакете LINDO. Отмечается, что для проведения анализа на чувствительность и постоптимального анализа не могут быть в полной мере использованы существующие пакеты программ NETFLO, NETSOLVE, GENOS, CPLEX, QSB, LINDO.

В работе [25] рассмотрены способы реализации процедур ведущего преобразования в схеме прямого симплекс алгоритма для задач транспортного типа, позволяющие осуществлять перестройку базисного дерева за время линейное от числа вершин сети, существенно сократив при этом число проверок условия оптимальности. Основанием для проведения работ явилось то, что многие полиномиальные алгоритмы не могут составить конкуренцию прямому симплекс алгоритму при решении практических задач. Излагается способ учета дуг, заведомо удовлетворяющих условию оптимальности, основанный на упорядочении изучения базисного дерева. При использовании данного способа из поиска очередного кандидата для ввода в базис исключаются дуги, для которых установлено выполнение условия оптимальности. Предложенные в работе структуры данных и процедуры для них позволяют уменьшить вычислительную сложность всей итерации прямого симплекс-метода, а не только ведущего преобразования как в известных алгоритмах. Приводится техника программной реализации на языке C++ процедур для решения и постоптимизационного анализа транспортных задач соответственно в сетевой и матричной постановках. Отмечается, что предложенный способ дает возможность распараллеливания вычислений при решении многих задач об оптимальном потоке.

В [26] приводятся результаты экспериментальных исследований по решению транспортной задачи прямым симплекс алгоритмом при распараллеливании вычислений на 14 процессорах. Даны расчеты для сетей, содержащих до 3000 узлов поставщиков и потребителей (т. е. $l = k = 3000$). Экспериментально показано, что затраты времени для задач с одинаковым числом поставщиков и потребителей размерностью $n \times n$ растут пропорционально n^a, где $2 \le a \le 2{,}2$. Кроме того, в этой работе выполнен прекрасный исторический обзор статей, в которых предлагались новые АТД и процедуры для улучшения программной реализации прямого симплекс алгоритма для решения классической транспортной задачи.

Как уже указывалось, венгерский метод решения транспортной задачи связан с задачей нахождения максимального потока в сети, а сама транспортная задача является частным случаем задачи о потоке минимальной стоимости. Обширную библиографию по алгоритмам нахождения максимального потока и их временной сложности можно найти в [27, стр. 793–794]. Самая быстрая реализация алгоритма нахождения максимального потока разработана Гольдбергом и Рао [28], асимптотическая оценка их алгоритма составляет $O(\min(v^{2/3}, e^{1/2}) e \lg(v^2 / e + 2) \lg C)$, где v и e — число узлов и дуг сети, а C — максимальная пропускная способность дуги (для транспортной задачи $C = \max(a_i, b_j)$, $i = \overline{1,l}$, $j = \overline{1,k}$). Обзор развития методов и алгоритмов решения задачи о потоке минимальной стоимости и оценки их временной сложности приведены в работе [29, стр. 339–344, 395–397].

Кроме традиционных методов, для решения транспортной задачи успешно применяются и другие [30–32].

В работе [30] сделана попытка приспособления двойственного метода аффинного преобразования (метода внутренней точки Кармаркара) для решения потоковых задач. Сравнивается вычислительная эффективность сетевого симплексного алгоритма с двойственным методом аффинного преобразования на примерах решения задач о потоке минимальной стоимости, транспортной задачи и задачи о назначениях. Показано, что сетевой симплекс алгоритм на сетях большой размерности работает значительно быстрее. Для распараллеливания на кластере лучше второй алгоритм.

В [31] также реализован метод двойственного аффинного преобразования и проводится его сравнение с градиентным методом, сетевым симплекс методом (пакет NETFLO), методами декомпозиции (пакет RELAX). Для задачи о назначениях размерностью 500x500 (37501 дуг) при распараллеливании ее решения на 8 процессорах время работы алгоритма внутренней точки несколько больше, чем у градиентного алгоритма. По времени работы на сетях размерностью до 72000 узлов сетевой симплекс алгоритм и декомпозиционные алгоритмы оказались соизмеримыми, а алгоритм внутренней точки в большинстве случаев отставал от лучших результатов.

В недавно вышедшей книге [32], посвященной задачам дробного программирования и недифференцируемой оптимизации, также рассматриваются различные подходы к решению транспортной задачи, основанные на схемах декомпозиции по ограничениям и переменным с применением субградиентных методов. Отмечается, что их применение целесообразно, когда число потребителей намного больше числа поставщиков, а также, если в транспортную задачу включены дополнительные сложные ограничения.

В настоящее время для решения транспортных задач успешно применяются метаэвристические методы и алгоритмы, — генетические и эволюционные [33–36], имитации отжига [37], роя частиц [38, 39]. Так, например, в работе [35] приводится генетический алгоритм, позволяющий параллельно генерировать множество субоптимальных решений для линейной транспортной задачи. Дана реализация алгоритма на языке C++. Отмечается, что алгоритм может быть легко приспособлен для решения нелинейных задач. В [39] предложен многороевый алгоритм для приближенного решения транспортных задач с нелинейными функциями стоимости произвольного вида. Приведены результаты вычислительных экспериментов по исследованию эффективности решения нелинейных транспортных задач различной сложности в параллельной реализации разработанных алгоритмов.

В заключение обзора приведем несколько последних работ, представляющих интерес для выбора методов решения транспортной задачи [40–44]. В [40] предлагается метод решения задач транспортного типа, в котором последовательно решаются двумерные задачи с одной связывающей переменной. Для двумерных задач задаются коэффициенты целевой функции для связывающей переменной и формулируются одномерные задачи. Их решения могут дать исходный оптимум или определить систему ограничений на переменные. Допустимые решения построенной системы ограничений дают оптимальное решение исходной задачи. Если система ограничений не имеет допустимых решений, решается задача о максимальном потоке, находится множество взаимно удовлетворённых пар, формируется множество обобщённых производителей и потребителей и путём суммирования строится новая задача с меньшим числом ограничений. После этого процесс последовательного решения двумерных задач повторяется. Алгоритм строит последовательность псевдорешений с монотонным возрастанием функционала. Отмечается, что метод может быть распространен на широкий класс транспортных и распределительных задач.

В работах [41, 42] соответственно изучаются вопросы, связанные с вырождением базисных решений при построении начального опорного плана методом наименьшего элемента и влиянием выбора метода получения начального опорного плана на число итераций прямого симплекс алгоритма для транспортной задачи (его еще называют MODI или stepping stone методом [14]) для получения оптимального решения. Сравниваются методы северо-западного угла, выбора минимального элемента и метод Фогеля. Как и ожидалось, лучшими оказались метод наименьшего элемента и метод Фогеля. На больших размерностях метод северо-западного угла намного хуже (приблизительно в 3 раза).

В [43] рассматривается Shortlist Method, который является улучшением прямого симплекс метода и позволяет существенно сократить число просмотров небазисных переменных, включаемых в базис, и уменьшить объем памяти компьютера для хранения данных. Экспериментальные исследования показали, что предложенный алгоритм позволяет значительно уменьшить время решения транспортной задачи по сравнению с последней версией (state-of-the-art) симплекс алгоритма, реализованного в пакетах CRAN-Package emdist и LP_Solve. Алгоритм позволяет решать транспортные задачи большой размерности за разумное время (задача с 1000 поставщиками и потребителями решена за 1,2314 с на компьютере Intel Core i7 CPU, 3.20 GHz). Кроме того, в работе приводится новый метод построения начального допустимого решения и экспериментально показано, насколько сокращается время решения задачи при его использовании по сравнению с известными методами северо-западного угла, минимального элемента, Фогеля и др.

В работе [44] обсуждается транспортная задача с нечеткими ограничениями на ресурсы, когда значения a_i, $i=\overline{1,l}$ и b_j, $j=\overline{1,k}$ задаются наборами нечетких треугольных чисел. Возвращаясь к решению задачи развозки порожних контейнеров (6), (35), отметим, что в реальных сетях величина грузопотоков всегда изменяется во времени. Как правило, внешние входящие потоки мелкопартионных грузов не подчиняются какому-либо известному вероятностному закону распределения (биномиальному, пуассоновскому и пр.). Соответственно будут изменяться и грузопотоки в контейнерах, которые заданы матрицей A. Поэтому для определения, например, среднесуточных потоков между узлами для различных периодов на протяжении года и в зависимости от сезонных колебаний потоков, в АИАС [1] должны использоваться современные методы математического моделирования на основе временных рядов данных, которые позволяют получить высококачественные прогнозы для нелинейных и нестационарных процессов [45]. Нестационарность входящих потоков и других процессов, происходящих в сети, а также возможность с помощью АИАС оперативно управлять перераспределением потоков порожних контейнеров в реальном масштабе времени и послужили отправной точкой для выбора авторами детерминированной постановки задачи (6), (3–5). При наличии АИАС всегда известны реальные мелкопартионные потоки грузов и величина грузопотоков в контейнерах. Поэтому всегда можно оперативно решить задачу развозки порожних контейнеров точными методами. При этом небольшие колебания величины потоков будут незначительно отражаться на оптимальном распределении потоков порожних контейнеров и «гаситься» резервом грузоподъемности транспортных средств. Однако нужно оперативно следить, чтобы грузопоток груженых и порожних контейнеров не нарушал схемы распределения и маршрутизации потоков, полученной для периодов текущего планирования [2]. В то же время рассмотренная в [44] транспортная задача с нечеткими ограничениями на ресурсы может представлять интерес для случая, когда по каким-либо причинам АИАС временно неработоспособна.

3. Вычислительный эксперимент

Решение задачи развозки порожних контейнеров выполнялось на однородных транспортных сетях с числом узлов n = 100, 250, 500, 1000, 2000, 3000, 4000 и степенью узлов val = 5. Длины дуг сети и элементы матрицы потоков груженых контейнеров $A = \| a_{ij} \|_{n\times n}$ задавались датчиком равномерно распределенных псевдослучайных целых чисел в пределах от 80 до 300 км и от 1 до 20 контейнеров. Принимались следующие значения параметров: грузоподъемность транспортных средств W = 40 контейнеров; периодичность движения транспортных средств $T_{пер}$ = 24 ч; время стоянки транспортных средств в конечных пунктах следования $T_{ст}$ = 22 ч; средняя скорость движения транспортных средств $V_{ср}$ = 70 км/ч. Для расчета среднегодовых приведенных затрат на транспортировку и обработку порожних контейнеров использовались следующие тождественные (10) и (11) формулы в условных единицах стоимости:

$$Z^c_{тр} = \frac{1}{2}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{abs(a_{ij}-a_{ji})}{W}\left[\frac{13939{,}2(t_{ст}+2d^*_{ij}/V_{ср})}{t_{пер}}+7200+116{,}8d^*_{ij}\right],$$

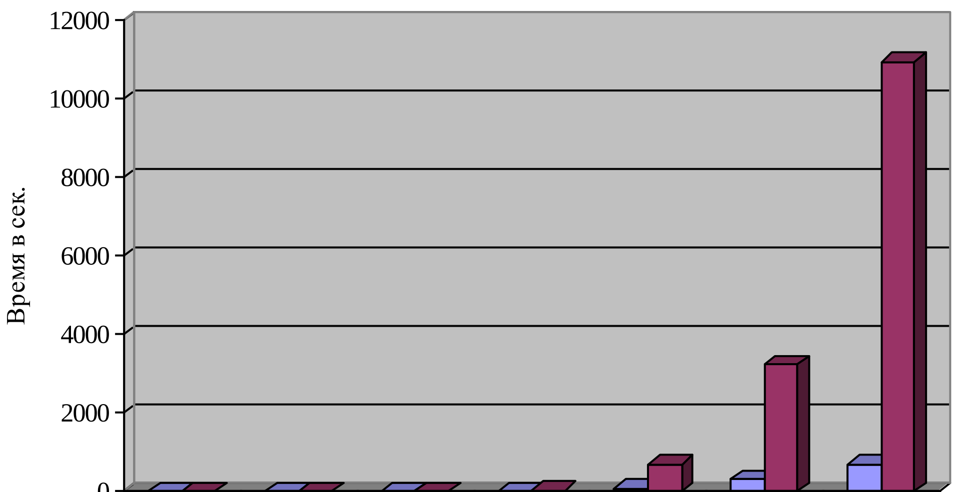
$$Z^c_{погр} = \sqrt{25595\left(\sum_{i=1}^{n}\sum_{j=1}^{n}abs(a_{ij}-a_{ji})\right)^2+2629\left(\sum_{i=1}^{n}\sum_{j=1}^{n}abs(a_{ij}-a_{ji})\right)}\ ,$$

$$Z^o_{тр} = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{k}\frac{x_{ij}}{W}\left[\frac{13939{,}2(t_{ст}+2\tilde{d}^*_{ij}/V_{ср})}{t_{пер}}+7200+116{,}8\tilde{d}^*_{ij}\right],$$

$$Z^o_{погр} = \sqrt{25595\sum_{i=1}^{l}\sum_{j=1}^{k}(2x_{ij})^2+2629\sum_{i=1}^{l}\sum_{j=1}^{k}2x_{ij}}\ .$$

Для нахождения кратчайших путей использовался алгоритм Флойда, а для решения транспортной задачи — венгерский алгоритм в матричной реализации. Из рис. 1 видно, что при $n \geq 1000$, $l \geq 503$, $k \geq 496$ начинает значительно возрастать время решения обеих задач, особенно у венгерского алгоритма. Сравнение методов балансировки удобнее проводить не по абсолютным значениям затрат $Z^c_{тр}$, $Z^o_{тр}$, $Z^c_{погр}$, $Z^o_{погр}$, а по снижению суммарных затрат $Z = (Z^c_{тр} + Z^c_{погр})/(Z^o_{тр} + Z^o_{погр})$ и снижению количества порожних контейнеров $U = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} abs(a_{ij} - a_{ji}) \right) / \left(2\sum_{i=1}^{l}\sum_{j=1}^{k} x_{ij} \right)$. На рис. 2 приведены соответствующие гистограммы.

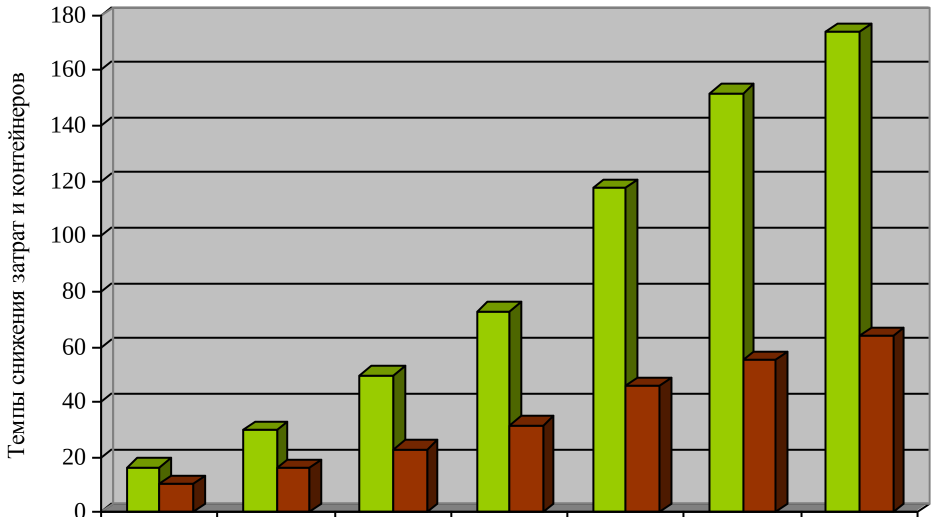
Программы составлены на языке Digital Visual Fortran и выполнялись под управлением операционной системы Windows Vista на ПЭВМ с процессором Intel Core 2 Duo с тактовой частотой 2,66 ГГц и оперативной памятью 2 Гб.



	100 (46,53)	250 (129,119)	500 (249,249)	1000 (503,496)	2000 (1015,983)	3000 (1520,147	4000 (1994,200
FLOYD	0,02	0,14	0,95	10,08	90,15	317,65	711,93
HM	0	0,22	3,2	51,07	706,4	3228,41	10940,01

Число узлов

Рис. 1 — Время построения кратчайших путей алгоритмом Флойда (FLOYD) и время решения транспортной задачи венгерским алгоритмом (HM) (в скобках указано соответственно число поставщиков и потребителей)



	100 (46,53)	250 (129,119)	500 (249,249)	1000 (503,496)	2000 (1015,983)	3000 (1520,1477)	4000 (1994,2005)
Z	16,63	30,04	49,75	72,96	117,31	151,92	173,76
U	10,14	16,24	23,11	31,63	45,7	55,49	63,72

Число узлов

Рис. 2 — Снижение суммарных затрат (Z) и количества порожних контейнеров (U) при применении оптимальной балансировки по сравнению с симметричной (в число раз)

Выводы

1. Рассмотрены два способа балансировки матрицы контейнерных потоков в транспортной сети — симметричный, часто применяемый на практике, и оптимальный, основанный на решении транспортной задачи. Предложена математическая модель и алгоритм решения задачи развозки порожних контейнеров, которые могут быть использованы для балансировки матрицы контейнерных потоков и последующего решения задачи распределения и маршрутизации потоков груженых и порожних контейнеров [2].

2. Для решения задачи нахождения кратчайших путей и транспортной задачи при числе узлов в сети, превышающем 1000, необходимо использовать более эффективные алгоритмы. В частности для решения транспортной задачи можно применять современные матричные и сетевые реализации прямого симплекс алгоритма [10, 25, 40, 43], технику распараллеливания вычислений на многоядерных процессорах и вычислительных кластерах [26], приближенные генетические, эволюционные и др. мета-алгоритмы [33–39].

3. Экспериментально показано, что оптимальная балансировка по сравнению с симметричной балансировкой позволяет значительно сократить суммарные затраты на транспортировку и обработку порожних контейнеров (на сетях от 100 до 4000 узлов в 17 и 174 раза соответственно).

СПИСОК ЛІТЕРАТУРИ

1. Васянин В.А. Автоматизация процессов принятия решений в многопродуктовых коммуникационных сетях с мелкопартионными дискретными потоками / В.А. Васянин, А.Н. Трофимчук // Екологічна безпека та природокористування: Зб. наук. праць. — Київ, 2010. — Вип. 5. — С. 172–213.
2. Васянин В.А. Задача распределения и маршрутизации транспортных блоков со смешанными вложениями и ее декомпозиция / В.А. Васянин // Проблемы управления и информатики. — 2015. — № 1. — С. 144–156.
3. Hitchcock F.L. The distribution of a product from several sources to numerous localities / F.L. Hitchcock // Journal Mathematics and Physics. — 1941. — Vol. 20. — N 2. — P. 224–230.
4. Koopmans T.C. Optimum Utilization of the transportation system / T.C. Koopmans // Proceeding of the International Statistical Conferences. — Washington, 1947. — N 5.
5. Канторович Л.В. Применение математических методов в вопросах анализа грузопотоков / Л.В. Канторович, М.К. Гавурин // Проблемы повышения эффективности работы транспорта. — М.: Изд-во АН СССР, 1949. — С. 110–138.
6. Гольдштейн Е.Г. Задачи линейного программирования транспортного типа / Е.Г. Гольдштейн, Д.Б. Юдин. — М.: Наука, 1969. — 374 с.
7. Габасов Р. Методы линейного программирования. Часть 2. Транспортные задачи / Р. Габасов, Р.М. Кириллова. — Минск: Изд. — Белорусск. ун-та, 1978. — 236 с.
8. Kuhn H.W. The Hungarian method for the assignment problem / H.W. Kuhn // Naval research logistics quaterly. — 1955. — Vol. 2. — N 1, 2. — P. 83–97.
9. Диниц Е.А. Алгоритм поразрядного сокращения невязок и транспортные задачи // Исследования по дискретной математике. — М.: Наука, 1973. — С. 46–57.
10. Ким И.В. Об эффективности алгоритмов решения двухкомпонентных задач линейного программирования // Экономика и матем. методы. — М., 1974. — Т. 10, вып. 3. — С. 621–631.
11. Прыгичев А. Н. Эффективная реализация метода потенциалов для транспортной задачи // Применение математики в экономике. — Л.: Изд-во ЛГУ, 1977. — Вып. 12. — С. 125–131.
12. Грибов А.В. Рекурсивное решение транспортных задач линейного программирования // Математика, Механика. Астрономия: Вестн. ЛГУ. — Л., 1978. — Вып. 4. — № 19. — С. 11–19.
13. Barr R.S. An improved version of the Out-of-Kilter method and a comparative study of computer codes / R.S. Barr, F. Glover, D. Klingman // Math. Prog. — 1974. — Vol. 7. — N 1. — P. 60–86.
14. Glover F. Implementation and computational comparisons of primal, dual, and primal-dual computer codes for minimum cost network flow problems / F. Glover, D. Karney, D. Klingman // Networks. — 1974. — Vol. 4. — N 3. — P. 191–212.
15. Charnes A. Past, present and future of large scale transshipment computer codes and applications / A. Charnes, D. Karney, D. Klingman, J. Stutz, F. Glover // Comput. & Ops. Res. — 1975. — Vol. 2. — P. 71–81.
16. Glover F. A practitioner's guide to the state of large scale network and network-related problems / F. Glover, D. Klingman. — National Computer Conference, 1976. — P. 945–950.
17. Mulvey J.V. Testing of a large-scale network optimization program / J.V. Mulvey // Math. Prog. — 1978. — Vol. 15. — N 3. — P. 291–314.
18. Бушуева С.Ф. Результаты конкурса «Транспорт-81» / С.Ф. Бушуева и др. // Экономика и матем. методы. — М., 1983. — Т. 19, вып. 1. — С. 176–181.
19. Zadeh N. More pathological examples for network flow problems / N. Zadeh // Math. Prog. — 1973. — Vol. 5. — N 2. — P. 217–224.

20. Zadeh N. A bad network problem for the simplex-method and other minimum cost flow algorithms / N. Zadeh // Math. Prog. — 1973. — Vol. 5. — N 3. — P. 255–266.
21. Ikura Y. Computational experience with a polynomial-time dual simplex algorithm for the transportation problem / Y. Ikura, G.L. Nemhauser // Discrete applied mathematics. — 1986. — Vol. 13. — P. 239–248.
22. Orlin J.B. Polynomial dual network simplex algorithms / J.B. Orlin, S.A. Plotkin, E. Tardos // Math. Program. — 1993. — Vol. 60A. — N 3. — P. 255–276.
23. Kleinschmidt P. A Strongly Polynomial Algorithm for the Transportation Problem / P. Kleinschmidt, H. Schannath // Mathematical Programming. — 1995. — Vol. 68. — N 1. — P. 1–13.
24. Adlakha V. Managing cost uncertainties in transportation and assignment problems / V. Adlakha, H. Arsham // Journal of applied mathematics & decision sciences. — 1998. — Vol. 2. — N 1. — P. 65–104.
25. Панюков А.В. Техника программной реализации потоковых алгоритмов / А.В. Панюков, В.А. Телегин // Вестник ЮУрГУ. — 2008. — № 27(127) . — С. 78–99.
26. Miller D.L. Solution of large dense transportation problems using a parallel primal algorithm / D.L. Miller, J.F. Pekny, G.L. Thompson. — Management Science Research Report No. 546, 1988. — 25 p.
27. Кормен Т. Алгоритмы: построение и анализ, 2-е издание / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн: Пер. с англ. — М.: Издательский дом «Вильямс», 2005. — 1296 с.
28. Goldberg A.V. Beyond the Flow Decomposition Barrier / A. V. Goldberg , S. Rao // Journal of the ACM. — 1998. — N 45. — P. 783–797.
29. Ahuja R.K. Network flows: theory, algorithms, and applications / R.K. Ahuja, T.L. Magnanti, J.B. Orlin. — Prentice-Hall, Inc. Upper Saddle River, New Jersey, 1993. — 846 p.
30. Armacost A. A computational comparison of the network simplex method with the dual affine scaling method / A. Armacost, S. Mehrotra // Opsearh. — 1991. — Vol. 28. — N 1. — P. 18–35.
31. Resende M. An implementation of the dual affine scaling algorithm for minimum cost flow on bipartite incapacitated networks / M. Resende, G. Veiga // SIAM Journal on Optimization. — 1992. — P. 1–22.
32. Соломон Д.И. Дробное программирование и недифференцируемая оптимизация. — Кишинев: Эврика, 2010. — 556 с.
33. Michalewicz Z. Genetic algorithms + data structures = evolution programs. — Berlin, Heidelberg: Springer, 1996. — 387 p.
34. Sheng S. Genetic algorithm for the transportation problem with discontinuous piecewise linear cost function / S. Sheng, Z. Dechen, X. Xiaofei // International Journal of Computer Science and Network Security. — 2006. — Vol. 6. — N 7A. — P. 182–190.
35. Емельянова Т.С. Об одном генетическом алгоритме решения транспортной задачи / Т.С. Емельянова. — Известия ТРТУ. Тематический выпуск «Интеллектуальные САПР». — Таганрог: Изд-во ТРТУ, 2007. — № 1 (73) . — С. 65–70.
36. Michalewicz Z. Evolutionary computation techniques for nonlinear programming problems / Z. Michalewicz // International Transactions in Operational Research. — 1994. — Vol. 1. — N 2. — P. 223–240.
37. Altiparmak F. An adaptive tabu-simulated annealing for concave cost transportation problems / F. Altiparmak, I. Karaoglan // Journal of the Operational Research Society. — 2008. — Vol. 59. — N 3. — P. 331–341.
38. Huang H. Particle swarm optimization algorithm for transportation problems / H. Huang, Z Hao // Particle swarm optimization, ed. Aleksandar Lazinica, InTech. — 2009. — P. 275–290.

39. Мальковский С.И. Решение нелинейных транспортных задач методом роя частиц / С.И. Мальковский, В.В. Пересветов // Моделирование систем. — 2012. — № 2, (32) С. 54–64.
40. Кузовлёв Д.И. Итеративный алгоритм для класса оптимизационных задач транспортного типа / Д.И. Кузовлев. — Автореферат диссертации на соискание ученой степени кандидата физико-математических наук, Москва, 2013. — 21 с.
41. Loch G.V. Computational Study of Degeneracy in Initial Basic Feasible Solution for the Transportation Problem / G.V. Loch, A.C. Lindbeck da Silva, T.C. Lindbeck da Silva // International Refereed Journal of Engineering and Science (IRJES). — 2014. — Vol. 3. — N 7. — P. 14–19.
42. Loch G.V. A Computational Study on the Number of Iterations to Solve the Transportation Problem / G.V. Loch, A.C. Lindbeck da Silva // Applied Mathematical Sciences. — 2014. — Vol. 8. — N 92. — P. 4579–4583.
43. Gottschlich C. The Shortlist Method for Fast Computation of the Earth Mover’s Distance and Finding Optimal Solutions to Transportation Problems / C. Gottschlich, D. Schuhmacher. — PLoS ONE 9(10): e110214. doi:10.1371/journal.pone.0110214. — 2014. — Vol. 9. — N 10. — P. 1–10, www.plosone.org.
44. Ивохин Е.В. О подходах к решению транспортной задачи с нечеткими ресурсами / Е.В. Ивохин, А.Б.С. Камл // Проблемы управления и информатики. — 2014. — № 5. — С. 122–133.
45. Довгий С.О., Бідюк П.І., Трофимчук О.М., Савенков О.І. Методи прогнозування в системах підтримки прийняття рішень. — К.: Азимут-Україна, 2011. — 608 с.

Стаття надійшла до редакції 03.12.2014