УДК 519.854.2: 004.023

А.Н. ТРОФИМЧУК, В.А. ВАСЯНИН, В.Н. КУЗЬМЕНКО

АЛГОРИТМЫ ОПТИМИЗАЦИИ УПАКОВОК МЕЛКОПАРТИОННЫХ КОРРЕСПОНДЕНЦИЙ В КОММУНИКАЦИОННЫХ СЕТЯХ

Аннотация. Рассмотрены алгоритмы решения задачи оптимизации упаковок, возникающей при сортировке и упаковке мелкопартионных грузов в контейнеры в магистральных транспортных сетях или объединении сообщений в виртуальные контейнеры в опорных сетях передачи данных. Предложена постановка задачи и обсуждаются ее особенности и подходы к решению. Проводится анализ сходимости и временной сложности ряда эвристических алгоритмов и с помощью вычислительных экспериментов исследуется их сравнительная эффективность. Экспериментально показано, что результаты решения задачи, полученные различными стратегиями оптимизации на сетях, содержащих до 500 узлов, отличаются не более чем на 2.65%.

Ключевые слова: магистральные коммуникационные сети, мелкопартионные потоки корреспонденций, оптимизация упаковок, эвристические алгоритмы, сходимость и временная сложность алгоритмов.

ВВЕДЕНИЕ

В магистральных транспортных сетях и опорных сетях передачи данных с мелкопартионными потоками корреспонденций от отправителей к получателям возникает задача упаковок, сущность которой заключается в концентрации потоков грузов и информации между узлами сети. Исходящие из узлов сети мелкопартионные корреспонденции с разными адресами назначения могут объединяться одна с другой и упаковываться в общие транспортные блоки (контейнеры или виртуальные контейнеры). В результате объединения потоков уменьшается количество транспортных блоков для транспортировки корреспонденций, сокращается число направлений сортировки корреспонденций в узлах сети в другие узлы, увеличивается коэффициент загрузки транспортных блоков и транспортных средств, более продуктивно используется высокая пропускная способность магистральных каналов связи. В то же время в отдельных узлах сети возникают транзитные потоки мелкопартионных корреспонденций, что приводит к увеличению общей стоимости обработки потоков в сети и увеличению времени доставки корреспонденций получателю.

В настоящей статье предлагаются алгоритмы для решения практических задач упаковок с функциями затрат на обработку и транспортировку мелкопартионных корреспонденций, которые основаны на дискретном аналоге метода локального спуска, когда окрестности метрического пространства допустимых решений выбираются из эвристических (интуитивных) соображений с учетом специфики структур данных и особенностей решаемой задачи. Обсуждаются вопросы сходимости и временной сложности алгоритмов, а также на численных примерах исследуется их сравнительная вычислительная эффективность.

ПОСТАНОВКА ЗАДАЧИ

Пусть $G(N,E)$ — неориентированная магистральная сеть с множеством узлов N, $n=|N|$, и множеством дуг E, $e=|E|$, $|\cdot|$ — знак мощности множества. Задано множество корреспонденций, каждая из которых характеризуется узлом отправления i, узлом назначения j и объемом корреспонденции $a_{ij} \in Z^{+}$, $i, j = \overline{1, n}$. Все корреспонденции должны транспортироваться по сети в некоторых транспортных блоках фиксированной емкости $\omega \in Z^{+}$. Величины a_{ii}, $i=\overline{1, n}$, определяют внутриузловые объемы на обработку корреспонденций, которые в транспортные блоки не упаковываются и по сети не транспортируются. Все корреспонденции однородны (одного типа), при транспортировке могут объединяться в различных узлах и упаковываться в транспортные блоки только целиком, т.е. запрещается их разветвление (расщепление, дробление на части) и транспортировка по нескольким путям. Пусть объемы корреспонденций заданы исходной матрицей требований $A=\| a_{ij} \|_{n \times n}$, $a_{ij} \in \{1, \ldots, \omega, \ldots, a_{\max}\}$, $i, j=\overline{1, n}$, в которой строки соответствуют узлам отправления, а столбцы — узлам назначения, и имеется некоторая преобразованная матрица $X=\| x_{ij} \|_{n \times n}$, элементы которой являются целочисленными искомыми переменными задачи упаковок.

Требуется найти минимум функции

$$\sum_{ij \in S} C_{\mathrm{tr}}^{ij}(u_{ij}, d_{ij})+\sum_{i=1}^{n} C_{\mathrm{sort}}^{i}(x_{i}, q_{i})+\sum_{i=1}^{n} C_{\mathrm{load}}^{i}(u_{i}) \tag{1}$$

при ограничениях

$$\sum_{j=1}^{n} x_{ij}-\sum_{j=1,\, j \neq i}^{n} a_{ij}=\sum_{j=1}^{n} x_{ji}-\sum_{j=1,\, j \neq i}^{n} a_{ji}, \; i=\overline{1, n}, \tag{2}$$

$$\sum_{j=1}^{n} x_{ij}-\sum_{j=1,\, j \neq i}^{n} a_{ij} \leq h_{i}, \; i=\overline{1, n}, \tag{3}$$

$$0 \leq x_{ij} \leq\left\lceil\frac{a_{ij}}{\omega}\right\rceil \omega \;\; \forall ij \in S, \tag{4}$$

где S — множество пар индексов ij корреспонденций, определенное на декартовом произведении $n \times n$; $x_{ij}=a_{ij}+\sum_{rs} a_{rs}^{*}$, если корреспонденция a_{ij} не объединялась ни с какой другой корреспонденцией ($\{a_{rs}^{*}\}$ — множество корреспонденций, объединенных с корреспонденцией a_{ij}), и $x_{ij}=0$, если корреспонденция a_{ij} объединялась с какой-либо другой корреспонденцией или $i=j$; $C_{\mathrm{tr}}^{ij}(u_{ij}, d_{ij})$ — нелинейная функция транспортных затрат, зависящая от количества транспортных блоков $u_{ij}=\left\lceil\frac{x_{ij}}{\omega}\right\rceil$ ($\lceil x\rceil$ — наименьшее целое, большее или равное x) и длины d_{ij} — пути их транспортировки между узлами i и j; $C_{\mathrm{sort}}^{i}(x_{i}, q_{i})$ — нелинейная функция затрат от суммарного объема $x_{i}=a_{ii}+\sum_{j=1,\, j \neq i}^{n}(a_{ij}+a_{ji})+\sum_{j=1}^{n} x_{ij}-\sum_{j=1,\, j \neq i}^{n} a_{ij}$ и количества направлений сортировки $q_{i}=q_{in}^{i}+\sum_{j=1}^{n} \delta_{ij}$ мелкопартионных корреспонденций, обрабатываемых в узле i ($\delta_{ij}=1$, если $x_{ij} \neq 0$, и $\delta_{ij}=0$, если $x_{ij}=0$, а q_{in}^{i} определяет заданное количество направлений сортировки для обработки корреспонденций a_{ii}, $i=\overline{1, n}$); $C_{\mathrm{load}}^{i}(u_{i})$ — нелинейная функция затрат от суммарного количества транспортных блоков $u_i = \sum_{j=1}^{n} (u_{ij} + u_{ji})$, обрабатываемых в узле i; h_i, $i = \overline{1, n}$, — максимальная пропускная способность i-го узла по обработке транзитных мелкопартионных корреспонденций.

При решении задачи также учитываются ограничения на время доставки $t_{ij} \le T_{ij}$ $\forall ij \in S$ и число транзитных объединений $\nu_{ij} \le \nu_{\max}$ $\forall ij \in S$ мелкопартионных корреспонденций при их транспортировке из узлов отправления в узлы назначения, где T_{ij} и $\nu_{\max}$ — соответственно заданное время доставки мелкопартионных корреспонденций получателю и максимально допустимое число транзитных объединений корреспонденций.

Первая составляющая функции (1) определяет транспортные затраты, вторая — затраты на сортировку, а третья — затраты на обработку транспортных блоков. Выражения (2), (3) и (4) представляют условия баланса, ограничения на пропускные способности узлов и значения переменных x_{ij}.

НЕКОТОРЫЕ ПРАКТИЧЕСКИЕ АСПЕКТЫ РЕШЕНИЯ ЗАДАЧ

В транспортных сетях мелкопартионные грузы, как правило, перевозятся в контейнерах, на поддонах, в пленочной или другой упаковке и т.п. Это связано с рядом факторов, наиболее важными из которых являются сохранность грузов, возможность их автоматизированной и механизированной обработки, хранение на открытых площадках, упрощение обработки сопроводительной документации и пр. Перед транспортировкой мелкопартионные грузы в узлах транспортной сети рассортировываются по адресам их доставки в сортировочных центрах, оснащенных высокопроизводительными автоматизированными сортировочными системами. Технологический процесс сортировки заключается в следующем. Грузы из общего накопителя или накопителей поступают на линии сортировки, в начале которых установлены считыватели штрих-кодов или другие более современные устройства распознавания адреса груза и сопроводительной информации о нем. В соответствии с расшифрованной информацией груз направляется в нужный адресный накопитель. Количество адресных накопителей соответствует количеству направлений сортировки грузов. В сетях передачи данных также выполняются подобные технологические операции сортировки информационных потоков. Современные опорные сети передачи данных с волоконно-оптическими линиями связи и технологиями спектрального уплотнения каналов WDM, DWDM, HDWDM (wavelength-division multiplexing, dense WDM, high dense WDM) позволяют мультиплексировать от 16 до 64 и более спектральных каналов в одной линии связи и передавать объемы информации до 27 Тбит/с (1Тбит = 10^{12} бит) и более. В таких сетях в роли «сортировочной машины» выступают мультиплексоры, которые объединяют отдельные информационные потоки в виртуальные контейнеры.

При практическом проектировании и анализе коммуникационных сетей должны использоваться реальные стоимостные показатели, например среднегодовые приведенные затраты на обработку и транспортировку потоков. На практике существует множество различных функций, используемых для расчета затрат на функционирование сети. По мере организационно-технического совершенствования сети эти функции постоянно подвергаются изменениям и уточнениям. В каждом случае определение адекватных функций затрат является сложной задачей, которая должна быть решена отдельно перед проведением численного моделирования. Очевидно, что при решении задач оптимизации в целевую функцию должны быть включены только наиболее важные составляющие затрат, зависящие от искомых переменных. В сформулированной задаче предполагается использование во всех составляющих целевой функции капитальных и эксплуатационных затрат, приведенных к сопоставимому виду.

Как правило, в математических моделях, описывающих процессы обработки и транспортировки потоков, затраты связывают с величиной потока по дугам сети или путям передачи потока. Для сетей передачи данных, где дуги ассоциируются с каналами связи, такие постановки оказываются достаточно приемлемыми. В случае транспортных сетей намного сложнее адекватно определить стоимостные функции, например $C_{\text{tr}}^{ij}(u_{ij}, d_{ij})$, а значит, и получить в результате решения задачи достоверный ответ. Представление потоковых задач в виде задач линейного программирования со стоимостными коэффициентами вообще может оказаться абсурдным с практической точки зрения. По-видимому, в реальных задачах нужно рассчитывать транспортные затраты по маршрутам транспортных средств, т.е. связывать объемы и пути распределения потоков с множеством искомых «оптимальных» маршрутов [1]. Для каждого определенного в результате решения маршрута, зная его характеристики (длину, грузоподъемность и тип транспортного средства и пр.), легко рассчитать среднегодовые приведенные затраты для каждого маршрута и получить более достоверную оценку транспортных затрат для всей сети перевозок. «Маршрутный» подход приемлем и для сетей передачи данных, так как каждый маршрут — это либо ориентированная дуга, либо последовательность таких дуг (скоммутированные каналы). В задаче (1)–(4) не вводятся маршрутные функции затрат, поскольку она является одной из первых подзадач обобщенной задачи обработки, распределения и маршрутизации мелкопартионных корреспонденций [2]. При ее решении рассчитывается только ориентировочная оценка транспортных затрат и нижняя граница затрат на обработку транспортных блоков. Поэтому $C_{\text{tr}}^{ij}(u_{ij}, d_{ij})$ может быть принята как функция удельной стоимости транспортировки потока величиной u_{ij} на расстояние d_{ij} от грузоподъемности транспортного средства или пропускной способности канала связи, задаваемой в виде параметра $w_\xi \in \{w_1, w_2, \ldots, w_\alpha\}$, $\xi = \overline{1, \alpha}$. Например, можно считать, что $C_{\text{tr}}^{ij}(u_{ij}, d_{ij}) = u_{ij}(k_1^\xi + k_2^\xi d_{ij}) / w_\xi \ \forall ij \in S$, где k_1^ξ, k_2^ξ — заданные коэффициенты. Учитывая, что функции C_{sort}^i и C_{load}^i должны отражать реальные затраты, при моделировании задачи необходимо так подобрать значения k_1^ξ, k_2^ξ, w_ξ, чтобы транспортные затраты были соизмеримы с затратами на сортировку мелкопартионных корреспонденций и обработку транспортных блоков в узлах сети, т.е. чтобы порядок величины транспортных затрат был близок к фактическим. Реальные транспортные затраты и затраты на обработку транспортных блоков рассчитываются при решении задачи распределения и маршрутизации потоков [1].

Еще одной особенностью рассматриваемой модели являются ограничения $t_{ij} \le T_{ij}$ на время доставки мелкопартионных корреспонденций конечным получателям и ограничения $\nu_{ij} \le \nu_{\max}$ на максимально допустимое число объединений корреспонденций с другими корреспонденциями. При решении задачи (1)–(4) рассчитываются только предварительные значения t_{ij}, так как до решения задачи распределения и маршрутизации потоков транспортных блоков неизвестны фактические пути следования мелкопартионных корреспонденций от отправителей к получателям.

АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧИ

В общем случае целевая функция (1) является невыпуклой и многоэкстремальной. Целочисленность переменных задачи и наличие ограничений

$t_{ij} \le T_{ij}$, $\nu_{ij} \le \nu_{\max}$ $\forall ij \in S$ приводит к необходимости поиска и разработки комбинаторных приближенных методов, использующих специфику структуры задачи, эвристические приемы и интерактивный режим поиска решения. Из постановки задачи (1)–(4) следует, что можно выписать начальное допустимое решение: $X = A$, где a_{ii}, $i = \overline{1, n}$, заменены нулями. Пусть в процессе решения задачи над элементами $x_{ik}, x_{kj}, x_{ij} \ne 0$, $k, i, j = \overline{1, n}$, $k \ne i \ne j$, матрицы X могут итеративно выполняться следующие фиксированные операции:

$$x_{ik} \leftarrow x_{ik} + x_{ij}\,,\; x_{kj} \leftarrow x_{kj} + x_{ij}\,,\; c_{ij} \leftarrow k,\, y_k \leftarrow y_k + x_{ij}, x_{ij} \leftarrow 0\,, \tag{5}$$

где символ $\leftarrow$ обозначает операцию присваивания; k — номер узла, в котором выполняется объединение корреспонденции x_{ij}; c_{ij} — элементы справочной матрицы объединения корреспонденций $C = \| c_{ij} \|_{n \times n}$; y_k — элементы вектора сумм транзитных корреспонденций $Y = \| y_k \|$, $k = \overline{1, n}$. Элементы матрицы C определяются так:

$$c_{ij} = \begin{cases} k, \text{ если корреспонденция } x_{ij} \text{ объединяется с корреспонденцией } x_{ik}, \\ i, \text{ если корреспонденция } x_{ij} \text{ не объединяется} \\ \qquad\qquad \text{ни с какой другой корреспонденцией,} \\ 0, \text{ если } i = j. \end{cases} \tag{6}$$

С помощью матрицы C и алгоритмов, приведенных в [3], можно определить последовательность $\Omega_{ij} = \{(i, k_1), (k_1, k_2), \ldots, (k_m, j)\}$ с промежуточными узлами $\{k_1, k_2, \ldots, k_m\}$ $(\nu_{ij} = | \{k_1, k_2, \ldots, k_m\} |)$, в которых выполнялось объединение каждой корреспонденции a_{ij}, а также найти множество других корреспонденций $\{a^*_{rs}\}$, которые уже были объединены с a_{ij} на предыдущих итерациях.

Предположим, что известна справочная матрица путей транспортировки потоков транспортных блоков $C^* = \| c^*_{ij} \|_{n \times n}$, полученная при построении кратчайших путей в сети G по ступенчатому критерию — минимум транзитных узлов на пути следования потока, минимум длины пути [4]. Элементы матрицы C^* определяются так:

$$c^*_{ij} = \begin{cases} 0, \text{ если } i = j, \\ k, \text{ если } u_{ij} \text{ следует в узел } j \text{ с перегрузкой в узле } k, \\ i, \text{ если } u_{ij} \text{ следует в узел } j \text{ без перегрузки.} \end{cases}$$

Отметим, что кратчайшие пути, построенные от узла i, представляют собой дерево с корнем в i, которое может быть однозначно описано строкой справочной матрицы C^*, а k является предпоследним узлом на кратчайшем пути от i до j. Поэтому с помощью этой матрицы легко определить число транзитных узлов на пути следования любого потока. Известна также матрица длин построенных путей $D = \| d_{ij} \|_{n \times n}$. Матрицы C^* и D используются при расчете значений функции $C^{ij}_{\text{tr}}(u_{ij}, d_{ij})$. Пусть t_a — заданное время на сортировку мелкопартионных корреспонденций в узле, t_b — среднее время на транзитную перегрузку или перекоммутацию транспортных блоков, V_{av} — средняя скорость движения транспортных средств (км/час) или скорость распространения сигнала по каналам связи (км/с). Требуется рассчитать значения t_{ij}. Для предварительной оценки времени доставки достаточно использовать справочную матрицу объединения корреспонденций $C = \| c_{ij} \|_{n \times n}$ и матрицы C^* и D. Приведем формулу для расчета t_{ij} $\forall ij \in S,\ i \neq j$:

$$t_{ij} = \begin{cases} 2t_a + d_{ij} / (V_{av}\theta) + \psi_{ij} t_b, \text{ если } c_{ij} = i, \\ t_a(\nu_{ij} + 2) + \sum\limits_{\xi\eta \in \Omega_{ij}} (d_{\xi\eta} / (V_{av}\theta) + \psi_{\xi\eta} t_b), \text{ если } c_{ij} \neq i, \end{cases} \qquad (7)$$

где $\psi_{\xi\eta}$ — число транзитных перегрузок u_{ij} на участке (ξ, η); θ — коэффициент нормирования времени (для транспортных сетей θ = 24 ч, для сетей передачи данных $\theta = 1$).

В [5] приведены результаты экспериментального исследования двух простых алгоритмов упаковок при изменении границ для начальных значений $\mathrm{MIN} \le (x_{ij} = a_{ij}) \le \mathrm{MAX}$ при фиксированном ω. В настоящей статье, как и в [5], рассматриваются три возможных случая при выполнении преобразования (5) для значения $\Delta^u_{ij,k} = u_{ik} + u_{kj} + u_{ij} - (u'_{ik} + u'_{kj})$ при $\Delta^u_{ij,k} < 0$, $\Delta^u_{ij,k} = 0$ и $\Delta^u_{ij,k} = 1$, где

$$u_{ik} = \left\lceil \frac{x_{ik}}{\omega} \right\rceil,\ u_{kj} = \left\lceil \frac{x_{kj}}{\omega} \right\rceil,\ u_{ij} = \left\lceil \frac{x_{ij}}{\omega} \right\rceil,\ u'_{ik} = \left\lceil \frac{x_{ik} + x_{ij}}{\omega} \right\rceil,\ u'_{kj} = \left\lceil \frac{x_{kj} + x_{ij}}{\omega} \right\rceil.$$

Показано, что для большинства интервалов изменения значений a_{ij} лучшие решения могут быть получены при строгом соблюдении ограничений (4) и их ослаблении, т.е. при $\Delta^u_{ij,k} > 0$ и $\Delta^u_{ij,k} \ge 0$.

Для описания алгоритма введем следующие обозначения. Пусть

$$\Delta F^-_{ij}(k) = \sum_{ij \in \{ik, kj, ij\}} C^{ij}_{\mathrm{tr}}(u_{ij}, d_{ij}) + \sum_{i \in \{i, k\}} C^i_{\mathrm{sort}}(x_i, q_i) + \sum_{i \in \{i, k, j\}} C^i_{\mathrm{load}}(u_i),$$

$$u_i = \sum_{l=1}^{n} (u_{il} + u_{lj});$$

$$\Delta F^+_{ij}(k) = \sum_{ij \in \{ik, kj\}} C^{ij}_{\mathrm{tr}}(u'_{ij}, d_{ij}) + C^i_{\mathrm{sort}}(x_i, q_i - 1) + C^k_{\mathrm{sort}}(x_k + x_{ij}, q_k) + \sum_{i \in \{i, k, j\}} C^i_{\mathrm{load}}(u'_i),$$

$u'_i = \sum_{l=1}^{n} (u'_{il} + u'_{li})$, $u'_{ij} = 0$; $k \in K$, K — множество узлов-претендентов, через которые может быть выполнено объединение корреспонденции x_{ij} без нарушения ограничений (3) и $t_{ij} \le T_{ij}$, $\nu_{ij} \le \nu_{\max}$. Для возможности «прорыва» из точек локального минимума в другие точки минимума во множество K можно включать узлы, для которых $\Delta^u_{ij,k} \ge 0$.

Приведем общую схему алгоритма. Знаки $\wedge$ и $\vee$ соответственно означают операции конъюнкции и дизъюнкции. Знаком *** отделяются комментарии.

Алгоритм optpac1

1. С помощью алгоритма [4] для заданных длин дуг сети G построить для всех корреспонденций кратчайшие пути по ступенчатому критерию — минимум транзитных узлов в пути, минимум длины пути и получить матрицы C^* и D.

2. Рассчитать полные затраты ZP (значение функции (1)) до оптимизации. $DELF \leftarrow ZP$.

3. Выбрать значение $\nu_{\max}$; $\nu \leftarrow 1$; $ZMIN \leftarrow \infty$.

4. Ввести исходную матрицу A; $X \leftarrow A$; установить $c_{ij} \leftarrow i\ \forall i \neq j$ и $x_{ii} \leftarrow 0$, $c_{ii} \leftarrow 0$, $i, j = \overline{1, n}$. *** c_{ij} — элементы справочной матрицы объединения корреспонденций (6).

5. Матрицы $\Lambda \leftarrow 0$, $B \leftarrow 1$; $b_{ii} \leftarrow 0$, $i = \overline{1, n}$; $C_1 \leftarrow 0$. *** $\Lambda = \| \Delta_{ij} \|_{n \times n}$, $B = \| b_{ij} \|_{n \times n}$ — рабочие матрицы, C_1 — счетчик входов в пп. 6–20.

6. Для $\{ i | i = \overline{1, n} \}$ выполнить пп. 7–11.

7. Для $\{ j | j = \overline{1, n} \}$ выполнить пп. 8–10.

8. Если $i \neq j \wedge b_{ij} = 1 \wedge x_{ij} \neq 0$, то $k \leftarrow i$, перейти к п. 9, иначе перейти к п. 10.

9. Найти множество K узлов-претендентов, с помощью которых может быть выполнено объединение корреспонденции x_{ij} без нарушения ограничений (3), $\Delta^u_{ij,k} \geq 0$ и ограничений $t_{ij} \leq T_{ij}$, $\nu_{ij} \leq \nu$, $t_{rs} \leq T_{rs}$, $\nu_{rs} \leq \nu$ для a_{ij} и множества корреспонденций $\{a^*_{rs}\}$, уже объединенных с a_{ij}; $\Delta_{ij} \leftarrow \max\limits_{k \in K} (\Delta F^-_{ij}(k) - \Delta F^+_{ij}(k))$, $c_{ij} \leftarrow \arg \max\limits_{k \in K} (\Delta F^-_{ij}(k) - \Delta F^+_{ij}(k))$.

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

11. Перейти к п. 6. *** Конец цикла по i.

12. Найти $\Delta^{\max}_{ij} = \max\limits_{ij \in S} \Delta_{ij}$. Если $\Delta^{\max}_{ij} > 0$, то $k \leftarrow c_{ij}$, перейти к п. 13, иначе перейти к п. 21.

13. Выполнить преобразование $x_{ik} \leftarrow x_{ik} + x_{ij}$, $x_{kj} \leftarrow x_{kj} + x_{ij}$, $y_k \leftarrow y_k + x_{ij}$, $x_{ij} \leftarrow 0$, $\Delta_{ij} \leftarrow 0$.

14. Матрица $B \leftarrow 0$. Для $\{ l | l = \overline{1, n} \}$ выполнить $b_{lk} \leftarrow 1$; $b_{kl} \leftarrow 1$; $b_{lj} \leftarrow 1$; $b_{jl} \leftarrow 1$; $b_{il} \leftarrow 1$; $b_{li} \leftarrow 1$.

15. Для $\{ l | l = \overline{1, n} \}$ выполнить пп. 16–19.

16. Для $\{ m | m = \overline{1, n} \}$ выполнить пп. 17–18.

17. Если $c_{lm} = i \vee c_{lm} = k \vee c_{lm} = j$, то $b_{lm} \leftarrow 1$, иначе перейти к п. 18.

18. Перейти к п. 16. *** Конец цикла по m.

19. Перейти к п. 15. *** Конец цикла по l.

20. $C_1 \leftarrow C_1 + 1$. Перейти к п. 6.

21. По формуле (7) рассчитать время доставки t_{ij} $\forall ij \in S$. *** Временная сложность $O(3n^2)$.

22. Рассчитать полные затраты ZP (значение функции (1)) после оптимизации; $DELF1 \leftarrow DELF$; $DELF \leftarrow DELF - ZP$. *** Временная сложность $O(n^2)$.

23. Если $DELF > 0$, то перейти к п. 24, иначе к п. 29.

24. Если $ZP < ZMIN$, то перейти к п. 25, иначе к п. 26.

25. $ZMIN \leftarrow ZP$; $ZSTOP \leftarrow ZP$; $\nu_{\text{opt}} \leftarrow \nu$; $\nu \leftarrow \nu + 1$. Если $\nu > \nu_{\max}$, то перейти к п. 30, иначе перейти к п. 4.

26. Найден локальный минимум $ZMIN$ при значении ν_{opt}. Если $\nu = \nu_{\text{opt}}$, то перейти к п. 30, иначе перейти к п. 27.

27. Если $ZSTOP \neq ZP$, то перейти к п. 28, иначе к п. 30.

28. $ZSTOP \leftarrow ZP$; $\nu \leftarrow \nu + 1$. Если $\nu > \nu_{\max}$, то перейти к п. 30, иначе перейти к п. 4.

29. Если $\nu = 1$, то найден локальный минимум $DELF1$ при значении $\nu_{\text{opt}} = \nu - 1$, иначе найден локальный минимум $ZMIN$ при значении ν_{opt}.

30. Конец алгоритма.

В алгоритме п. 5 $\Lambda = \| \Delta_{ij} \|_{n \times n}$ — рабочая матрица для значений $\Delta_{ij} = \Delta F^-_{ij}(k) - \Delta F^+_{ij}(k)$, а $B = \| b_{ij} \|_{n \times n}$ используется для ограничения пересчета оценок в Λ, так как после преобразования x_{ij} оценки в Λ должны быть пересчитаны только для множества корреспонденций $\{x_{rs}\}$, для которых полученные оценки были связаны с преобразованной корреспонденцией x_{ij}. В п. 9 $\forall x_{ij}, ij \in S$, выбирается наилучшая оценка Δ_{ij} на множестве K, а в п. 12 определяется корреспонденция x_{ij}, подлежащая преобразованию. Если $\Delta_{ij} = 0 \ \forall ij \in S$, то выполняется выход из внешнего цикла пп. 6–20 и оптимизация для текущего значения ν завершена. В противном случае выполняется преобразование x_{ij} при $\Delta_{ij}^{\max} = \max\limits_{ij \in S} \Delta_{ij}$ (п. 13), пересчитываются оценки в Λ (пп. 14–19), на единицу увеличивается счетчик C_1 и осуществляется переход на начало внешнего цикла пп. 6–20. В пп. 21–29 определяются результаты оптимизации для текущего значения ν и проверяются условия останова алгоритма. В случае нахождения локального минимума выводится его значение. Алгоритм заканчивает работу, если $\nu > \nu_{\max}$ или $ZSTOP = ZP$, т.е. когда значение целевой функции перестает изменяться. Если эти условия не выполняются, алгоритм продолжает работу, начиная с п. 4, для нового значения $\nu \leftarrow \nu + 1$.

Приведенный алгоритм аналогичен алгоритму метода вектора спада И.В. Сергиенко [6]. В данном случае радиус окрестности в алгоритме метода вектора спада — это нарастающие значения $\nu \leftarrow \nu + 1$, а действительные значения вектора спада — это значения Δ_{ij} матрицы Λ. Если $\Delta_{ij} = 0 \ \forall ij \in S$, то выполняется переход на увеличение $\nu \leftarrow \nu + 1$ до тех пор, пока $\nu \le \nu_{\max}$ или $ZSTOP \ne ZP$.

Пусть F_0 — значение функции (1) до оптимизации (т.е. значение (1) при $X = A$).

Теорема. Если множество M допустимых решений задачи (1)–(4), определяемых ограничениями (3), (4) и $t_{ij} \le T_{ij}$, $\nu_{ij} \le \nu_{\max}$, не пусто, то с помощью алгоритма optpac1 в худшем случае за асимптотическое время $O(Cn^5)$ будет получено некоторое текущее решение $F_\nu < F_0$ или найден локальный минимум $F_{\min}$ функции (1).

Доказательство. Функция (1) ограничена сверху значением F_0, и при $M = \varnothing$ это значение является решением задачи. Функция (1) ограничена также и снизу, поскольку при итеративном выполнении преобразований (5) при условии $\Delta_{ij,k}^u > 0$ или $\Delta_{ij,k}^u \ge 0$ и даже полном ослаблении ограничений (3) и $t_{ij} \le T_{ij}$, $\nu_{ij} \le \nu_{\max}$ (т.е. при $h_i = \infty$, $i = \overline{1, n}$; $T_{ij} = \infty$, $ij \in S$; $\nu_{\max} = \infty$) будет достигнуто то состояние, когда дальнейшее выполнение преобразования (5) для элементов матрицы X не может привести к уменьшению количества транспортных блоков. Во вложенных циклах по i и j (пп. 6–11) для каждой корреспонденции x_{ij} определяется множество K (п. 9) и по мощности этого множества выбирается максимальная оценка $\Delta_{ij} = \max\limits_{k \in K}(\Delta F_{ij}^-(k) - \Delta F_{ij}^+(k))$. Трудоемкость определения K и выбора Δ_{ij} составляет $O(C_2 n)$, где C_2 — некоторая константа, максимальное значение которой зависит от $|\{a_{rs}^*\}|$ и числа рекурсий алгоритмов [3] для нахождения $\{a_{rs}^*\}$ и проверки ограничений $t_{rs} \le T_{rs}$ и $\nu_{rs} \le \nu$. Поэтому выполнение действий в пп. 6–11 потребует порядка $O(C_2 n^3)$ времени. Операции в пп. 12, 14–19 имеют сложность порядка $O(2n^2 + n)$, поэтому сложность выполнения пп. 6–19 составит $O(C_2 n^3 + 2n^2 + n)$. На выполнение пп. 6–20 потребуется $O(C_1 C_2 n^3 + 2C_1 n^2 + C_1 n)$ времени для каждого значения $\nu = 1, \ldots, \nu_{\max}$. Внешний цикл в пп. 6–20 конечен, так как при заданных или незаданных значениях h_i и T_{ij} и текущем значении ν все элементы матрицы Λ в итоге получат нулевые значения, т.е. все x_{ij} будут преобразованы и определится текущее значение F_ν.

Внешний цикл по $\nu = 1, \ldots, \nu_{\max}$ (пп. 4–30) также конечен, что обеспечивается в пп. 23–28. Таким образом, в последовательности получаемых решений $F_1, F_2, \ldots, F_{\nu_{\max}}$ возможны следующие случаи:

1) $F_1 > F_2 > \ldots > F_{\nu_{\max}}$, т.е. значения функции цели монотонно уменьшаются и значение $F_{\nu_{\max}}$ является лучшим текущим решением;

2) $F_1 > F_2 > \ldots > F_{\xi-1} > F_{\xi} = F_{\xi+1}$, $\xi + 1 \le \nu_{\max}$, т.е. при ξ найден локальный минимум $F_{\xi} = F_{\min}$;

3) последовательность $F_1, F_2, \ldots, F_{\nu_{\max}}$ не монотонна; тогда алгоритм среди нескольких локальных минимумов (пп. 23–28) выберет наилучший при значении $\nu = \nu_{\text{opt}}$.

Во всех случаях останов алгоритма во внешнем цикле по ν гарантирован, поскольку в итоге будет выполнено одно из условий: $\nu > \nu_{\max}$ или $ZSTOP = ZP$.

Общее время работы алгоритма с учетом операций в пп. 21 и 22 в худшем случае составит $O(\nu_{\max}[C_1 C_2 n^3 + 2C_1 n^2 + C_1 n) + 4n^2])$. Величина C_1 существенно зависит от границ изменения значений a_{ij} по сравнению со значением ω, а также от области допустимых решений, определяемой ограничениями на h_i и T_{ij}. Значение C_1 может быть достаточно большим по сравнению со значением $\nu_{\max} C_2$ и достигать $(n^2 - n)$. Таким образом, в асимптотике время работы алгоритма не лучше, чем $O(Cn^5)$, где $C = \nu_{\max} C_2$.

Теорема доказана.

Учитывая большую сложность приведенного алгоритма для решения задачи (1)–(4), в работе [7] предложены ограниченный вариант алгоритма optpac1 — optpac2 и ряд эвристических алгоритмов, разработка которых обоснована тем, что для реальных коммуникационных сетей сложно определить функции C_{tr}^{ij}, C_{sort}^{i} и C_{load}^{i}, достаточно адекватно характеризующие затраты на процессы транспортировки и обработки потоков корреспонденций. Эвристические алгоритмы (optpac3, optpac4 и optpac5) ориентированы на максимальное сокращение транспортных блоков в сети и не используют в своей работе функций затрат. Однако полученное с помощью этих алгоритмов решение задачи по-прежнему оценивается. Для оценки выполняется расчет по заданным конкретным функциям годовых приведенных затрат.

Алгоритм optpac2 отличается от optpac1 ограниченным выбором узлов во множество K, в которое включаются только узлы, находящиеся на кратчайших путях, определяемых из справочной матрицы путей транспортировки потоков транспортных блоков C^*. Поэтому временная сложность действий в п. 9 может быть понижена до $O(C_2 n_{av})$, где n_{av} — среднее число транзитных узлов в пути от i до j в сети G и общая оценка будет не лучше $O(Cn^4 n_{av})$.

В алгоритмах optpac3 и optpac4 выбор x_{ij} из X происходит в двойном цикле по i и j из условия $ij \leftarrow \arg\min\limits_{i,\, j = \overline{1,\, n}} x_{ij}$ до тех пор, пока возможны преобразования (5), а в качестве k выбирается первый узел из полного или ограниченного множества K, если для k выполняются ограничения (3), $\Delta_{ij,\,k}^{u} \ge 0$ и ограничения $t_{ij} \le T_{ij}$, $\nu_{ij} \le \nu$, $t_{rs} \le T_{rs}$, $\nu_{rs} \le \nu$ для a_{ij} и множества корреспонденций $\{a_{rs}^*\}$, уже объединенных с a_{ij}. Временная сложность алгоритмов optpac3 и optpac4 соответственно составляет $O(\nu_{\max}[C_1(n^2 + C_2 n) + 4n^2])$ и $O(\nu_{\max}[C_1(n^2 + C_2 n_{av}) + 4n^2])$, а так как для этих алгоритмов $C_1 = n^2 - n$, окончательно получаем $O(\nu_{\max}[(n^4 + C_2 n^3 - n^3 - C_2 n^2) + 4n^2])$ и $O(\nu_{\max}[(n^4 + C_2 n^2 n_{av} - n^3 - C_2 n n_{av}) + 4n^2])$.

Алгоритм optpac5 отличается от алгоритмов optpac3 и optpac4 тем, что первоначально элементы матрицы X (когда X совпадает с A) сортируются по неубыванию их значений алгоритмом «карманной» (bucket sort) сортировки с построением вектора указателей на однонаправленные линейные списки элементов x_{ij} с одинаковым значением. Длина вектора указателей определяется максимальным значением $C_L = \lceil a_{\max} / \omega \rceil \omega$ в X. Индекс указателя в векторе указателей равен значению x_{ij}, а сам указатель содержит ссылку на первый элемент списка. Каждый элемент в списке содержит ссылку на следующий элемент и индексы i и j корреспонденции x_{ij}. Первоначально индексы корреспонденций в списках упорядочены в соответствии с просмотром элементов матрицы X слева направо и сверху вниз. В процессе работы алгоритма для изменяемых значений x_{ik} и x_{kj} эти списки динамически корректируются; этим поддерживается постоянное упорядочение. Выбор x_{ij} для выполнения преобразования (5) проводится в основном цикле по длине вектора указателей C_L в порядке возрастания значений x_{ij}. Узел k выбирается из ограниченного множества так же, как и в алгоритме optpac4. Поэтому временная сложность алгоритма optpac5 составляет $O(\nu_{\max}[C_1(n^2 n_L + C_L n_L C_2 n_{av}) + 4n^2]) = O(\nu_{\max}[n^2 n_L + C_L n_L C_2 n_{av} + 4n^2])$, так как C_1=1, где n_L определяется средним числом элементов в списке, имеющих одинаковое значение x_{ij}.

ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Во время проведения эксперимента сначала исследовалось поведение алгоритма optpac1 при изменении границ значений матрицы корреспонденций A по сравнению со значением ω, определяющим емкость транспортного блока, а затем сравнивались алгоритмы optpac1–optpac5 по быстродействию и полученным результатам. Исследования выполнялись на транспортной сети с длинами дуг r_{ij}, $ij \in E$, в пределах от 80 до 300 км. Значения матрицы корреспонденций и длины дуг генерировались датчиком псевдослучайных чисел. Использовались реальные функции годовых приведенных затрат C_{tr}^{ij}, C_{sort}^{i} и C_{load}^{i}, характерные для автотранспортных предприятий. В табл. 1 приведены результаты работы алгоритма optpac1 при заданных параметрах $n = 20$, $val = 5$ (степень узлов), $r_{ij} \in [80, 300]$, $ij \in E$, $\omega = 40$, $w_\xi = 40$, $t_a = 24$ ч, $t_b = 12$ ч, $V_{av} = 80$ км/ч при изменении интервалов значений a_{ij}, $i, j = \overline{1, n}$. Значения h_i, $i = \overline{1, n}$; T_{ij}, $ij \in S$; $\nu_{\max}$ не ограничены.

В табл. 2 приведены результаты сравнения алгоритмов optpac1 — optpac5 при увеличении n для $val = 5$, $r_{ij} \in [80, 300]$, $ij \in E$, $\omega = 40$, $w_\xi = 4000$, $t_a = 24$ ч, $t_b = 12$ ч, $V_{av} = 80$ км/ч, $a_{ij} \in [1, 120]$, $i, j = \overline{1, n}$; (O), (S) — соответственно означает, что алгоритм работает на полном или ограниченном множестве узлов претендентов для преобразования (5). Значения h_i, $i = \overline{1, n}$; T_{ij}, $ij \in S$; $\nu_{\max}$ при проведении эксперимента не ограничивались в целях максимально возможного расширения области допустимых решений задачи. В таблицах приняты обозначения: F_0, $F_{\min}$ — соответственно значения функции цели (1) (в условных единицах приведенных затрат) до и после оптимизации; $\Delta F_{\min}$ — отклонение целевой функции от лучшего значения (в процентах); $u_0 = \sum_{ij \in S} \left\lceil \frac{a_{ij}}{\omega} \right\rceil$, $u_{\min} = \sum_{ij \in S} \left\lceil \frac{x_{ij}}{\omega} \right\rceil$ — количество транспортных

Таблица 1. Результаты работы алгоритма optpac1

Интервалы значений a_{ij}, $i, j = \overline{1, n}$	F_0, у.е.	$F_0 - F_{\min}$, у.е. (экономия, %)	u_0, $u_{\min}$	ν_{opt}, ν	K_{av}	N_{av}	C_1	t_{av}, сут	t_{sol}, с
[1, 5]	518083	304316 (58.74)	380, 95	5, 7	0.522	4	285	3.67	43.52
[5, 10]	569149	242289 (42.57)	380, 144	4, 6	0.762	7	238	3.34	33.10
[5, 120]	1521120	70517.6 (4.64)	738, 674	1, 4	0.889	14	89	2.86	2.32
[1, 10]	548326	269046 (49.07)	380, 123	4, 7	0.715	6	260	3.45	42.15
[10, 20]	651824	151972 (23.31)	380, 230	1, 6	0.794	11	152	3.03	19.61
[10, 120]	1527020	61987.0 (4.065)	738, 681	2, 3	0.887	15	80	2.85	1.47
[1, 20]	607324	207486 (34.16)	380, 180	5, 6	0.787	8	204	3.20	30.65
[20, 40]	810251	0 (0.00)	380, 380	0, 1	0.690	19	0	2.57	0.00
[20, 120]	1548660	30307.9 (1.96)	738, 712	2, 3	0.884	15	73	2.78	0.86
[1, 40]	722062	115118 (15.94)	380, 270	5, 6	0.804	13	115	2.96	8.77
[40, 60]	1428870	40289.4 (2.82)	754, 754	1, 2	0.872	9	193	2.98	6.02
[40, 120]	1774980	18348.4 (1.03)	871, 871	1, 2	0.892	14	91	2.77	1.09
[1, 60]	954599	109222 (11.44)	495, 387	3, 5	0.836	13	118	2.99	8.27
[60, 80]	1639140	0 (0.00)	760, 760	0, 1	0.855	19	0	2.57	0.00
[60,120]	1950360	709.750 (0.04)	934, 934	1, 2	0.869	18	18	2.62	0.09
[1, 80]	1132660	86884.9 (7.67)	562, 479	5, 6	0.858	14	97	2.96	5.80
[80, 100]	2222300	0 (0.00)	1134, 1134	0, 1	0.744	19	0	2.57	0.02
[80, 120]	2283520	0 (0.00)	1135, 1135	0, 1	0.801	19	0	2.57	0.00
[1, 100]	1342400	83320.0 (6.21)	664, 586	3, 4	0.872	14	99	2.91	3.81
[100, 120]	2399780	0 (0.00)	1140, 1140	0, 1	0.907	19	0	2.57	0.00
[1, 120]	1519720	70138.6 (4.62)	738, 674	3, 4	0.887	14	90	2.90	2.31

блоков до и после оптимизации; ν_{opt}, ν — оптимальное и конечное число объединений корреспонденций (ν определяет число итераций алгоритмов для нахождения локального оптимума); $K_{av} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\delta_i}\sum_{j=1}^{n} x_{ij} \Big/ \left(\left\lceil \frac{x_{ij}}{\omega} \right\rceil \omega\right)$, $\delta_i = \sum_{j=1}^{n} \delta_{ij}$ — средний коэффициент загрузки транспортного блока; $N_{av} = \frac{1}{n}\sum_{i=1}^{n}\delta_i$ — среднее число направлений сортировки мелкопартионных корреспонденций в узле сети; C_1 — число вхождений в главный цикл оптимизации для выполнения преобразований (5); t_{av} — среднее время доставки мелкопартионных корреспонденций получателю (в сутках); t_{sol} — время решения задачи (в секундах).

Из табл. 1 видно, что на результаты решения задачи существенно влияют границы изменения значений a_{ij}, $i, j = \overline{1, n}$, по сравнению с величиной ω. При малой постоянной нижней границе и увеличении верхней границы значений a_{ij} ($a_{ij} \in [1, 5\text{–}120]$) экономия затрат составляет от 58.74 до 4.62%. При росте нижней и верхней границ и росте нижней и постоянной высокой верхней границе показатели экономии соответственно снижаются с 42.57% и 4.64% до нуля.

Для интервалов [20, 40], [60, 80], [80, 100], [80, 120], [100, 120] вообще невозможно уменьшить значение целевой функции, так как выполнение преобразований (5) может только увеличить количество транспортных блоков. В то же время для интервалов [40, 60], [40, 120], [60, 120], когда количество транспортных блоков до и после оптимизации не изменяется (при $\Delta_{ij,k}^{u} = 0$), имеем снижение затрат с 2,82 до 0,04%, что при большой размерности сети может составлять значительную сумму и представлять практический интерес. Несмотря на то, что значения h_i, $i = \overline{1, n}$; T_{ij}, $ij \in S$; $\nu_{\max}$ не ограничивались, среднее время доставки мелкопартионных корреспонденций получателю оставалось достаточно стабильным и не превышало 3.67 суток. Во всех случаях, когда оптимизация была возможна, достигался неплохой коэффициент загрузки транспортных блоков и уменьшалось количество направлений сортировки мелкопартионных корреспонденций в узлах сети.

Результаты работы алгоритмов optpac1–optpac5 при увеличении n от 25 до 500 (приведенные в табл. 2), показывают большую вычислительную трудоемкость алгоритмов optpac1 и optpac2. Алгоритм optpac1 дает лучшие результаты, но его «разумное» применение на обычных персональных компьютерах возможно на сетях, содержащих до 100 узлов. Алгоритмы optpac3–optpac5 работают значительно быстрее, но и отклонения от лучших результатов алгоритма optpac1 (на сетях до 100 узлов) составляют более 2%, что в абсолютном выражении по приведенным затратам достигает величины 462000 у.е. Начиная с размерности сети n = 75 алгоритм optpac3 показывает лучшие результаты, чем алгоритм optpac2, и при увеличении размерности сети остается лидером среди алгоритмов, не использующих в процессе оптимизации функции затрат. Для сетей большой размерности (от n = 200 и выше) даже 1.67–2.65% отклонения решений от лучшего локального минимума составляют в абсолютном выражении огромные суммы (от +1495500 до +14444000 у.е. для примера из табл. 2).

Особо отметим, что все алгоритмы допускают распараллеливание вычислительного процесса по значениям размера окрестности поиска локального минимума (параметра $\nu = 1, 2, \ldots, \nu_{\max}$). Поэтому решение задачи упаковок возможно на мощных вычислительных кластерах (High performance computing clusters, HPC) и многопроцессорных компьютерах, что существенно сократит время ре-

Таблица 2. Результаты работы алгоритмов optpac1–optpac5

Число узлов, n	Алгоритм	$F_{\min}$, у.е.	$\Delta F_{\min}$, %	ν_{opt}, ν	K_{av}	N_{av}	C_1	t_{av}, сут	t_{sol}, с
25	optpac1 (O)	0.150411E+07	0.00	2, 5	0.891	18	147	3.17	8.66
	optpac2 (S)	+16530	1.10	1, 2	0.805	20	90	2.79	0.25
	optpac3 (O)	+32730	2.18	1, 2	0.902	16	600	3.19	0.02
	optpac4 (S)	+31970	2.13	1, 2	0.832	19	600	2.82	0.05
	optpac5 (S)	+31970	2.13	1, 3	0.832	19	1	2.82	0.06
50	optpac1 (O)	0.588447E+07	0.00	4, 5	0.906	36	642	3.61	390.55
	optpac2 (S)	+86450	1.47	3, 4	0.838	38	513	3.15	9.61
	optpac3 (O)	+100510	1.71	3, 5	0.910	31	2450	3.68	0.36
	optpac4 (S)	+111360	1.89	2, 3	0.845	37	2450	3.16	0.16
	optpac5 (S)	+111160	1.89	2, 3	0.845	37	1	3.16	0.09
75	optpac1 (O)	0.129714E+08	0.00	2, 6	0.919	53	1540	4.00	4262.56
	optpac2 (S)	+234700	1.81	2, 3	0.848	57	1228	3.43	46.77
	optpac3 (O)	+167900	1.29	2, 4	0.917	46	5550	4.06	0.90
	optpac4 (S)	+261000	2.01	2, 3	0.848	55	5550	3.44	0.48
	optpac5 (S)	+261100	2.01	2, 3	0.848	55	1	3.44	0.12
100	optpac1 (O)	0.225900E+08	0.00	3, 7	0.921	71	2748	4.20	22998.70
	optpac2 (S)	+416600	1.84	3, 4	0.850	76	2220	3.59	224.25
	optpac3 (O)	+231300	1.02	4, 5	0.920	61	9900	4.25	3.10
	optpac4 (S)	+462000	2.05	2, 3	0.851	73	9900	3.61	1.40
	optpac5 (S)	+460700	2.04	2, 3	0.851	73	1	3.61	0.16
150	optpac2 (S)	+684400	1.35	3, 4	0.853	114	5173	3.88	1364.03
	optpac3 (O)	0.506015E+08	0.00	2, 4	0.924	91	22350	4.60	11.31
	optpac4 (S)	+745700	1.47	3, 4	0.853	110	22350	3.89	9.02
	optpac5 (S)	+749000	1.48	3, 4	0.853	110	1	3.89	0.44
200	optpac2 (S)	+1495500	1.67	2, 4	0,854	152	9240	4.07	4750.62
	optpac3 (O)	0.892978E+08	0.00	2, 5	0.926	121	39800	4.83	43.96
	optpac4 (S)	+1566600	1.75	3, 4	0.853	146	39800	4.09	28.97
	optpac5 (S)	+1567500	1.76	3, 4	0.853	146	1	4.10	0.84
250	optpac2 (S)	+2724000	1.97	3, 5	0,853	190	14523	4.24	16306.00
	optpac3 (O)	0.138052E+09	0.00	2, 4	0.928	150	62250	5.05	84.68
	optpac4 (S)	+2736000	1.98	3, 4	0.853	182	62250	4.26	72.34
	optpac5 (S)	+2735000	1.98	3, 4	0.853	182	1	4.26	1.65
300	optpac3 (O)	0.197984E+09	0.00	3, 4	0.929	179	89700	5.19	186.39
	optpac4 (S)	+4243000	2.14	3, 4	0.853	219	89700	4.37	163.72
	optpac5 (S)	+4258000	2.15	3, 4	0.853	219	1	4.37	3.12
350	optpac3 (O)	0.268714E+09	0.00	3, 4	0.930	209	122150	5.34	355.67
	optpac4 (S)	+6258000	2.33	3, 4	0.853	255	122150	4.51	317.09
	optpac5 (S)	+6257000	2.33	3, 4	0.853	255	1	4.51	5.99
400	optpac3 (O)	0.349885E+09	0.00	2, 4	0.931	238	159600	5.46	616.22
	optpac4 (S)	+8555000	2.45	3, 4	0.854	291	159600	4.61	551.29
	optpac5 (S)	+8554000	2.44	4, 5	0.854	291	1	4.60	15.01
450	optpac3 (O)	0.441988E+09	0.00	3, 4	0.932	267	202050	5.57	1009.65
	optpac4 (S)	+11290000	2.55	3, 4	0.853	327	202050	4.69	1031.15
	optpac5 (S)	+11292000	2.55	3, 4	0.853	327	1	4.69	25.02
500	optpac3 (O)	0.544746E+09	0.00	2, 4	0.932	297	249500	5.68	1831.16
	optpac4 (S)	+14418000	2.65	3, 4	0.853	363	249500	4.79	1793.43
	optpac5 (S)	+14444000	2.65	3, 4	0.853	364	1	4.79	47.85

шения и позволит использовать алгоритм optpac1 на сетях большой размерности. Проведенный эксперимент, безусловно, не претендует на полноту исследования описанных эвристических алгоритмов, поскольку такая работа связана со значительными затратами машинного времени и требует проверки стратегий оптимизации на представительном множестве исходных данных разных размерностей (генерируемых, например, различными датчиками случайных чисел). Несмотря на это, в процессе эксперимента подтвердились теоретические положения о сходимости и временной сложности алгоритмов, выявлена аномалия решения задачи для различных границ значений a_{ij} и ω.

Все расчеты получены на ПК с процессором Intel Core 2 Duo с тактовой частотой 2.66 ГГц и оперативной памятью 2 Гб.

ЗАКЛЮЧЕНИЕ

В работе предложена постановка задачи оптимизации упаковок, которая используется при сортировке и упаковке мелкопартионных грузов в контейнеры в транспортных сетях или объединении сообщений в виртуальные контейнеры в сетях передачи данных. Обсуждаются некоторые практические особенности ее решения для рассматриваемых сетей. Разработан ряд эвристических алгоритмов решения задачи и доказана их сходимость к локальному минимуму за полиномиальное время. Проведены экспериментальные исследования предложенных алгоритмов на работоспособность и выполнен сравнительный анализ их быстродействия на сетях, содержащих до 500 узлов. Алгоритмы решения задачи допускают распараллеливание процесса нахождения локальных оптимумов и для увеличения их быстродействия могут выполняться на вычислительных кластерах или многопроцессорных компьютерах. Результаты моделирования позволяют утверждать о практической значимости и актуальности решения задачи упаковок для коммуникационных сетей с мелкопартионными потоками корреспонденций.

СПИСОК ЛИТЕРАТУРЫ

1. Васянин В.А. Задача распределения и маршрутизации транспортных блоков со смешанными вложениями и ее декомпозиция // Проблемы управления и информатики. — 2015. — № 1. — С. 144–156.
2. Трофимчук А.Н., Васянин В.А. Моделирование упаковки, распределения и маршрутизации мелкопартионных потоков в многопродуктовой сети // Проблемы управления и информатики. — 2015. — № 4. — С. 132–146.
3. Васянин В.А. Справочная матрица слияния потоков в задачах оптимизации упаковок на многопродуктовых сетях // Системні дослідження та інформаційні технології. — 2014. — № 3. — С. 42–49.
4. Васянин В.А. Двухкритериальный лексикографический алгоритм построения всех кратчайших путей в сети // Кибернетика и системный анализ. — 2014. — № 5. — С. 122–131.
5. Трофимчук А.Н., Васянин В.А., Кузьменко В.Н. О сложности одной задачи оптимизации упаковок // Кибернетика и системный анализ. — 2016. — **52**, № 1. — С. 83–92.
6. Сергиенко И.В. Математические модели и методы решения задач дискретной оптимизации. — Киев: Наук. думка, 1988. — 472 с.
7. Васянин В.А. Сравнительная эффективность алгоритмов оптимизации упаковок в мультипотоковых сетях // Дискретные системы управления. — Киев: Ин-т кибернетики им. В.М. Глушкова АН УССР, 1988. — С. 36–45.

Поступила 23.08.2015