ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль–сентябрь 2022. Т. 29, № 3. С. 64–84

УДК 519.863 DOI 10.33048/daio.2022.29.736

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ КРУПНОМАСШТАБНЫХ ТРАНСПОРТНО-ЭКСПЕДИЦИОННЫХ СИСТЕМ

Е. А. Нурминский[1, a], *Н. Б. Шамрай*[2, b]

[1] Дальневосточный федеральный университет,
пос. Аякс, 10, 690922 Владивосток, Россия
[2] Институт автоматики и процессов управления ДВО РАН,
ул. Радио, 5, 690041 Владивосток, Россия
E-mail: [a] `nurminskiy.ea@dvfu.ru`, [b] `shamray@dvo.ru`

Аннотация. Моделирование масштабных экономических систем стало актуальной проблемой для крупного бизнеса. Такие реалии вынуждают плановые и аналитические отделы погружаться в сферу анализа и управления большими данными. Программные пакеты общего назначения не всегда подходят для решения возникающих проблем. Отсюда исходит запрос в академическое сообщество на разработку подходящего инструментария для работы с моделями на миллионы переменных и гигабайты данных. В статье описан опыт моделирования одной довольно распространённой транспортно-экспедиционной задачи, излагаются математические и вычислительные идеи, применённые для её исследования. Табл. 3, библиогр. 21.

Ключевые слова: математическое моделирование, линейная оптимизация, транспортно-экспедиционная система промышленного уровня, проекционный алгоритм.

Введение

С точки зрения теории исследования операций в работе рассматривается задача линейного программирования

$$\min_{x \in X} cx, \tag{1}$$

которая содержательно заключается в оптимизации управления большим парком железнодорожных вагонов различных типов в интересах

Исследование выполнено при финансовой поддержке Министерства науки и высшего образования России (проект № 075–02–2022–880).

крупного транспортного оператора, предоставляющего своим клиентам услуги по доставке разнородных грузов.

Опуская некоторые специфические детали, задачу (1) можно рассматривать как многоиндексную транспортную задачу линейной оптимизации. Основная сложность в работе с ней возникает из-за запутанных связей между индексами множеств, применяемых для описания различных технологических ограничений. В реальной жизни эти связи меняются от одной реализации проблемы к другой, часто зависят от специальных решений оператора, состояния транспортной инфраструктуры и т. д.

Бесспорно, чем выше степень детализации и реалистичность разрабатываемой модели, тем качественнее результаты её реализации. «Плата» за достижение высокого качества — это увеличение размерности задачи как с точки зрения ресурсозатрат на поддержку исходных данных, так и существенного роста числа переменных и ограничений. Даже если каждое из индексных множеств задачи (1) имеет довольно скромную мощность, их мультипликативный эффект легко увеличивает размерность итоговой задачи до сотен миллионов переменных, что представляет собой основную трудность для её моделирования и решения.

К счастью, класс транспортно-экспедиционных задач линейной оптимизации обладает рядом специфических особенностей, которые могут быть использованы для разработки новых подходов к их решению. Одной из таких особенностей является высокая разреженность матриц, формирующих задачу, и это, как правило, учитывается в современном ПО для оптимизации (см., например, [1–4]).

Другим направлением работы с оптимизационными задачами гигантской размерности является декомпозиция — разделение исходной проблемы на более мелкие подзадачи, которые решаются независимо, и последующая организация процесса координации решений для получения общего оптимума. Это весьма популярная область исследований со времён основополагающих работ Дж. Данцига и Ф. Вулфа [5], и просто невозможно в рамках статьи рассмотреть даже основные направления этих исследований. Однако, и здесь успех во многом зависит от структуры решаемой задачи, следовательно, декомпозиционный подход весьма специфичен.

В работе предлагается объединить оба упомянутых подхода для решения рассматриваемой задачи: декомпозировать исходную проблему на несколько основных блоков и далее полагаться на автоматическое обнаружение структуры внутри каждого из блоков, используя шаблоны разреженности.

Статья организована следующим образом. В разд. 1 приведена краткая постановка задачи и далее, в разд. 2 — соответствующая ей математическая формализация. В разд. 3 описан опыт программной реализации

и отмечены структурные особенности задачи. В разд. 4 обсуждаются теоретические аспекты решения линейных задач большой размерности и приводятся результаты структурного анализа некоторых практических транспортно-экспедиционных задач. В заключении резюмируются полученные результаты.

1. Краткое описание проблемы

Рассмотрим задачу управления парком грузовых железнодорожных вагонов, с которой сталкиваются транспортные операторы, выполняющие заказы на доставку грузов по железнодорожной сети. Транспортная сеть представлена набором железнодорожных маршрутов, соединяющих станции отправления и прибытия. Каждый маршрут рассматривается как отдельный самостоятельный объект со своими характеристиками (время в пути, стоимость и т. д.). Маршрут не содержит никаких других внутренних маршрутов. По базе поступивших заказов формируется предварительный график доставки грузов от поставщиков к потребителям. Допускается частичное исполнение заказов. В распоряжении оператора имеются собственные вагоны разных типов, а также есть возможность брать вагоны в краткосрочную аренду. Тип вагона определяет допустимый к перевозке груз и разрешённые к использованию маршруты. Обеспечение заказов вагонами осуществляется путём перемещения порожних вагонов между станциями. Целью транспортного оператора является максимизация общей оперативной прибыли на конечном временном горизонте планирования.

Фактический размер рассматриваемой железнодорожной сети составляет около 1000 станций и 200 тысяч маршрутов. На практике рассматривается чуть более 30 типов вагонов и 100 типов грузов. Ежемесячный график доставки состоит из более чем 10 тысяч отправлений.

На каждый день планового периода по станциям известно количество готовых к отправке вагонов каждого типа. Поступившие к исполнению заказы характеризуются станциями отправления и прибытия, типом груза, максимальным количеством необходимых вагонов, выручкой с одного отправленного вагона и временем выполнения заказа (погрузка/проезд/разгрузка). График исполнения заказов включает дни подачи и количество вагонов на погрузку. Задаётся допустимое отклонение от запланированных дней погрузки. Затраты оператора на перемещение одного порожнего вагона зависят от его типа и типа последнего перевозимого в этом вагоне груза. Для каждого маршрута известна продолжительность порожнего проезда.

Оперативная прибыль определяется суммарной выручкой от исполнения заказов за вычетом расходов на перемещение порожних вагонов и краткосрочную аренду, плату за простой вагонов на станциях и т. п.

Транспортному оператору необходимо, во-первых, выбрать наиболее выгодные заказы для исполнения, во-вторых, определить цепочки порожних рейсов, обеспечивающих заказы вагонами, в-третьих, при необходимости, арендовать вагоны для исполнения заказов.

По окончанию текущего планового периода на станциях формируется новый вариант дислокаций вагонов, что, в свою очередь, определяет начальное состояние системы для следующего периода планирования и тем самым оказывает влияние на будущие доходы оператора. Предполагается, что оператор может прогнозировать будущие заказы с данными об их объёмах и ожидаемой выручки. Этот прогноз используется для эффективного возврата вагонов в логистику будущих перевозок.

2. Формализация проблемы

Математическая формализация рассматриваемой проблемы в силу её значительных размеров, разнообразия внутренних процессов и входных данных представляет существенную трудность. Отметим, что модель находится в постоянном развитии и реагирует на изменения в технологиях, структуре спроса и предложения, трансформацию транспортной сети и т. п. Приведённая ниже математическая модель описывает основные строительные блоки транспортной системы, её динамику и потоковые балансы, а также показывает масштаб и сложность проблемы.

2.1. Обозначения. Следуя традиции большинства языков алгебраического моделирования (см., например, AMPL [6,7], GAMS [8], GMPL [9]), разделим вводимые обозначения на три категории: множества, параметры и переменные. Множества определяют объекты задачи и их дискретные характеристики. Параметры задают числовые значения измеримых данных, связанных с различными комбинациями элементов из множеств. Переменные — это неизвестные величины, значения которых определяются в ходе решения задачи.

Множества:

T — дни планового периода ($t \in T$);

S — станции отправления-прибытия ($s \in S$);

S^0 — станции, на которых запрещено хранение порожних вагонов ($S^0 \subset S$);

$R = S \times S$ — железнодорожные маршруты ($(s_1, s_2) \in R$);

V — типы вагонов ($v \in V$);

K — типы грузов ($k \in K$);

O — заказы на перевозку ($o \in O$);

$\overline{O}$ — заказы, прогнозируемые на период, следующий за T ($\overline{o} \in \overline{O}$).

Параметры:

q^t_{svk} — количество готовых к использованию вагонов типа v после разгрузки груза k на станции s в день t;

χ^t_o — максимальное количество вагонов заказа o, назначенное к погрузке в день t;

p_{ov} — выручка за один вагон типа v, доставленный по заказу o;

τ_o — продолжительность исполнения заказа o (погрузка/проезд/разгрузка) (в днях);

σ_o — максимально допустимое отклонения (в днях) от графика исполнения заказа o (т. е. к погрузке вагоны могут быть поданы в дни интервала $[t-\sigma_o, t+\sigma_o]$);

ν_o — издержки на краткосрочную аренду вагонов для исполнения заказа o;

$\overline{\chi}_{\overline{o}}$ — прогноз максимального числа вагонов на будущий заказ $\overline{o}$;

$\overline{p}_{\overline{o}v}$ — прогноз выручки с одного вагона типа v будущего заказа $\overline{o}$;

$c_{s_1s_2vk}$ — издержки на перемещение одного порожнего вагона типа v после груза k от станции s_1 на станцию s_2;

$\theta_{s_1s_2}$ — продолжительность перегона одного порожнего вагона от станции s_1 на станцию s_2 (в днях);

λ_s — стоимость хранения одного вагона в сутки на станции s.

Каждый заказ o характеризуется станцией отправления $s_1(o)$, станцией назначения $s_2(o)$, типом груза к перевозке $k(o)$. График подачи вагонов на погрузку задаётся множеством $Q = \big\{(t,o) \in T \times O \mid \chi^t_o > 0\big\}$.

Переменные:

y^t_{svk} — количество порожних вагонов типа v после груза k, скопившихся на станции s в день t;

$x^t_{s_1s_2vk}$ — количество порожних вагонов типа v после груза k, отправленных со станции s_1 на станцию s_2 в день t;

z^t_{ovi} — количество гружёных вагонов типа v, отправленных в соответствии с записью $(t,o) \in Q$ графика исполнения заказов с отклонением $i \in [-\sigma_o, \sigma_o]$ дней;

ζ^t_o — количество вагонов в краткосрочной аренде, привлечённых для реализации отправки $(t,o) \in Q$;

$\overline{y}_{sv}$ — оценочное количество вагонов типа v на станции s в будущем периоде;

$\overline{x}_{s_1s_2v}$ — оценочное количество порожних вагонов типа v, отправленных со станции s_1 на станцию s_2 в будущем периоде;

$\overline{z}_{\overline{o}v}$ — оценочное количество гружёных вагонов типа v, отправленных по заказу $\overline{o}$ будущего периода.

2.2. Ограничения. Характерной особенностью транспортно-экспедиционных задач являются условия сохранения вагонопотоков — вагоны

никуда не исчезают и не появляются из ниоткуда. В контексте рассматриваемой проблемы эти условия формализуются как балансы по отправлениям и прибытиям вагонов на железнодорожные станции и являются обязательными для осуществления перевозок.

Условие сохранения вагонопотоков должно быть выполнено в каждый день планового периода t по каждой станции s для каждого типа вагона v, ранее перевозящего груз k:

$$y^t_{svk} = q^t_{svk} + y^{t-1}_{svk} + \sum_{s_1\in S} x^{t-\theta_{s_1 s}}_{s_1 svk} - \sum_{s_2\in S} x^t_{ss_2 vk} \\ + \sum_{\substack{(t-i-\tau_o,o)\in Q:\\ s_2(o)=s,\, k(o)=k,\\ i\in[-\sigma_o,\sigma_o]}} z^{t-i-\tau_o}_{ovi} - \sum_{\substack{(t-i,o)\in Q:\\ s_1(o)=s,\\ i\in[-\sigma_o,\sigma_o]}} z^{t-i}_{ovi}. \quad (2)$$

Для оценки числа вагонов на станциях в будущем периоде также опираемся на принцип сохранения потоков и используем

1) дислокации вагонов к концу текущего планового периода T;

2) вагоны, отправленные в дни планового периода, прибытие которых ожидается в будущем периоде;

3) прогнозируемые вагонопотоки будущего периода.

Тем самым будущий пул вагонов задаётся уравнением

$$\overline{y}_{sv} = \sum_{k\in K} y^T_{svk} + \sum_{\substack{s_1\in S,\, k\in K,\\ t:\, t+\theta_{s_1 s}>T}} x^t_{s_1 svk} + \sum_{\substack{(t-\tau_o,o)\in Q:\, s=s_2(o),\\ i\in[-\sigma_o,\sigma_o],\, t+i>T}} z^{t-\tau_o}_{ovi} \\ + \sum_{s_1\in S} \overline{x}_{s_1 sv} - \sum_{s_2\in S} \overline{x}_{ss_2 v} + + \sum_{\overline{o}\in\overline{O}:\, s=s_2(\overline{o})} \overline{z}_{\overline{o}v} - \sum_{\overline{o}\in\overline{O}:\, s=s_1(\overline{o})} \overline{z}_{\overline{o}vk}. \quad (3)$$

Для станций, на которых запрещено хранение вагонов, дополнительно накладываются ограничения $y^t_{svk} = 0$ и $\overline{y}_{svk} = 0$ для всех $s \in S^0$.

Общее количество гружёных вагонов, отправляемых согласно записи графика $(t, o) \in Q$, не должно превышать заданной величины χ^t_o:

$$\zeta^t_o + \sum_{v\in V,\, i\in[-\sigma_o,\sigma_o]} z^t_{ovi} \leqslant \chi^t_o, \quad (t, o) \in Q. \quad (4)$$

Аналогично ограничиваем выполнение заказов будущего периода:

$$\sum_{v\in V} \overline{z}_{\overline{o}v} \leqslant \overline{\chi}_{\overline{o}}, \quad \overline{o} \in \overline{O}. \quad (5)$$

Система (2)–(5) задаёт базовые ограничения модели. Существует ряд дополнительных условий, учитывающих специфику перевозок. В данной работе эти условия рассматриваться не будут.

2.3. Целевая функция. Доходы транспортного оператора за текущий плановый период обозначим через $\Phi(x, y, z)$. Значение $\Phi(x, y, z)$ зависит от

1) общей выручки от исполнения заказов собственным парком вагонов

$$G_1(z) = \sum_{\substack{(t,o)\in Q,\, v\in V,\\ i\in[-\sigma_o,\sigma_o]}} p_{ov} z^t_{ovi};$$

2) общей выручки от исполнения заказов вагонами краткосрочной аренды

$$G_2(\zeta) = \sum_{(t,o)\in Q} (\min_{v\in V}\{p_{ov}\} - \nu_o)\zeta^t_o;$$

3) суммарных издержек на порожние передвижения

$$F_1(x) = \sum_{\substack{t\in T,\, (s_1,s_2)\in R,\\ v\in V,\, k\in K}} c_{s_1s_2vk} x^t_{s_1s_2vk};$$

4) суммарных издержек на хранение простаивающих вагонов

$$F_2(y) = \sum_{\substack{t\in T,\, s\in S,\\ v\in V,\, k\in K}} \lambda_s y^t_{svk}.$$

Следовательно,

$$\Phi(x, y, z, \zeta) = G_1(z) + G_2(\zeta) - F_1(x) - F_2(y).$$

При оптимальном управлении парком вагонов в текущем периоде необходимо учитывать и потенциальный будущий доход. Введём следующие показатели:

1) прогнозируемая выручка от исполнения будущих заказов

$$\overline{G}_1(\overline{z}) = \sum_{\overline{o}\in\overline{O},\, v\in V} \overline{p}_{\overline{o}} \overline{z}_{ov};$$

2) прогнозируемые издержки на порожнее передвижение в будущем периоде

$$\overline{F}_1(\overline{x}) = \sum_{(s_1,s_2)\in R,\, v\in V} \overline{c}_{s_1,s_2v} \overline{x}_{s_1s_2v};$$

3) прогнозируемые издержки на хранение простаивающих вагонов

$$\overline{F}_2(\overline{x}, \overline{y}, \overline{z}) = \overline{T} \sum_{s\in S,\, v\in V} \lambda_s \overline{y}_{sv} - \overline{\lambda}\Big(\sum_{\overline{o}\in\overline{O},\, v\in V} \tau_{\overline{o}} \overline{z}_{\overline{o}v} + \sum_{(s_1,s_2)\in R,\, v\in V} \theta_{s_1s_2} \overline{x}_{rv} \Big),$$

где $\overline{c}_{s_1,s_2v}$ — средние (по типам перевозимых грузов) издержки порожнего проезда вагона типа v от станции s_1 до станции s_2, $\overline{T}$ — число дней, составляющих будущий период, $\overline{\lambda}$ — средняя по станциям стоимость хранения одного вагона в сутки. Тогда

$$\overline{\Phi}(\overline{x},\overline{y},\overline{z}) = \overline{G}_1(\overline{z}) - \overline{F}_1(\overline{x}) - \overline{F}_2(\overline{x},\overline{y},\overline{z})$$

можно рассматривать, как верхнюю оценку дохода на будущий период.

За целевой показатель качества управления парком железнодорожных вагонов примем линейную функцию

$$L(x,y,z,\zeta,\overline{x},\overline{y},\overline{z}) = \Phi(x,y,z) + \overline{\Phi}(\overline{x},\overline{y},\overline{z}), \tag{6}$$

значение которой будем максимизировать на множестве допустимых вагонопотоков, задаваемом линейной системой (2)–(5).

3. Программная реализация и структурные особенности

Для программной реализации сложных многоиндексных оптимизационных задач рынок ПО предлагает разнообразные высокоуровневые языки моделирования. В данной работе критерием выбора такого языка служило наличие следующих его возможностей:

1) гибкий и простой для понимания набор инструментов описания основных частей модели (переменных, параметров, ограничений, целевой функции);

2) удобные средства для наполнения модели входными данными;

3) наличие сервисов для отладки модели;

4) присутствие инструментов для анализа решений;

5) возможность экспорта задачи в иные стандарты описания модели для дополнительного анализа и исследований (как правило, это MPS-формат для задач линейной оптимизации).

Для программирования рассматриваемой задачи была получена лицензия на использование алгебраического языка моделирования AMPL [6, 7]. AMPL является не только мощным и удобным языком моделирования, но и стандартом, поддерживаемым многими современными промышленными решателями (например, CPLEX [2], Gurobi [1]), а также общедоступным ПО, которое, например, можно найти на сервисе оптимизации NEOS-Server [10].

Численные методы линейной оптимизации оперируют вещественнозначными матрицами и векторами, в которые трансформируются ограничения и целевая функция модели. Из-за огромной размерности реальных задач создание таких матриц в оперативной памяти может занять весьма ощутимое время, поэтому требует к себе особого внимания.

Первые AMPL-реализации рассматриваемой проблемы были точной копией её алгебраической формы записи, приведённой выше: сначала

вводились параметры и переменные, затем в их терминах описывались ограничения (2)–(5) и целевая функция (6). Такая реализация модели соответствует так называемому строчному представлению её матриц: каждая строка матрицы определяется списком элементов её столбцов. На момент тестирования задачи в распоряжении были данные для железнодорожной сети, состоящей из 990 станций и 135520 маршрутов; ресурс оператора включал 29923 вагонов 31 различного типа; на месяц к исполнению поступило 1282 заказа. Первая возникшая трудность заключалась в том, что даже для краткосрочного горизонта планирования формирование задачи в оперативной памяти потребовало слишком много времени — около получаса для недельного управления парком вагонов и при увеличении горизонта планирования стремительно росло, что неприемлемо для производственного использования.

Отметим, что матрица ограничений (2)–(5) имеет высокую степень разреженности. Так, например, переменная y^t_{svk} входит только в два балансовых уравнения (2), соответствующих индексам (t, s, v, k) и $(t+1, s, v, k)$; переменная $x^t_{s_1 s_2 vk}$ входит в уравнение (2) с индексом (t, s_1, v, k) и, если $t + \theta_{s_1 s_2} \in T$, то в уравнение с индексом $(t + \theta_{s_1 s_2}, s_2, v, k)$, иначе в балансы будущего периода (3) с индексом (s_2, v); переменная z^t_{ovi} входит в уравнение (2) с индексом $(t + i, s_1(o), v, k(o))$ и, если $t + i + \tau_o \in T$, то в уравнение с индексом $(t + i + \tau_o, s_2(o), v, k(o))$, иначе в балансы будущего периода (3) с индексом $(s_2(o), v)$. Подобная специфика задачи позволяет представить матрицу её ограничений в так называемой столбцовой форме записи [11]: сначала объявляются ограничения и целевая функция модели, затем вводятся переменные, для которых указываются включающие ограничения и соответствующие коэффициенты. Этот способ моделирования имитирует описание разреженных матриц с помощью столбцов и является естественным для симплексоподобных алгоритмов.

Таблица 1

Время формирование AMPL-модели в оперативной памяти

$\|T\|$	Время, с, стр. запись	Время, с, столб. запись	Кол-во переменных	Кол-во ограничений	Число ненулевых элементов
1	43,82	35,88	7 911 511	63 254	15 844 030
2	633,03	52,46	1 188 3012	94 455	23 827 873
3	877,92	68,5	15 860 277	125 775	31 835 993
4	1130,81	86,93	19 831 744	156 933	39 858 570
5	1407,22	102,56	23 801 582	188 144	47 958 557
6	1682,08	120,39	27 771 335	219 335	56 147 993
7	1937,29	138,67	31 741 619	250 541	64 508 807

AMPL предоставляет средства для описания модели как в строковой, так и в столбцовой форме записи. Для сравнения было реализовано столбцовое представление задачи (2)–(5), (6) и проведён анализ времени формирования данных в оперативной памяти для обеих форм записи. Результаты приведены в табл. 1. Видим, что время генерации задачи для недельного горизонта планирования при столбцовой реализации в 14 раз меньше, чем при её строковой записи. Более того, эта разница увеличивается с увеличением размерности задачи. С учётом такой значительной разницы в машинном времени в качестве рабочей была принята столбцовая реализация модели.

Относительно размерности задачи и потребляемых ресурсов можно отметить следующее. Для текущего планового периода на 15 дней и будущего прогноза на заказы в течении следующих 14 дней получили задачу линейной оптимизации с 68374100 переменными и 499996 ограничениями. При экспорте задачи в MPS-формат был создан файл, занимающий около 5,8 ГБ дискового пространства, генерация файла заняла около 7 минут. Конфигурация ПК для тестирования была следующей: процессор Intel Xeon Gold 6136 с частотой 3 ГГц, оперативная память 96 ГБ. Задача решалась пакетом CPLEX (с использованием до 8 потоков). Подключение опции предварительной обработки задачи (presolver) позволило свести задачу к меньшей размерности: 20819229 переменных, 188338 ограничений и 43762028 ненулевых элементов. Суммарное время формирования и решения задачи составило около 4 часов.

4. Математические аспекты решения транспортно-логистических задач большой размерности

Как можно видеть в общем описании модели в разд. 1, её основная часть состоит из многопродуктовых многоиндексных межпериодных балансов вида

$$\begin{aligned} \sum_{J\colon (I,J)\in W} A^e_{I,J} x_{I,J} = b^e_I, \quad I \in S_e, \\ \sum_{I\colon (I,J)\in W} B^e_{I,J} x_{I,J} = b^e_J, \quad J \in D_e, \end{aligned} \tag{7}$$

где $\{I, J\}$ — это составные индексы выбираемые из подмножества W декартового произведения множеств продуктов $\{P\}$, типов средств перевозок $\{V\}$, временных периодов $\{T\}$, станций отправления и получения $\{S\}$ и, может быть, чего-то ещё. Переменные $x_{I,J}, I, J \in W$, представляют собой планируемые объёмы вагонов, предназначенных для выполнения предписанных перевозок, осуществляемых от отправителя, закодированного индексом I, к потребителю, закодированному индексом J.

Эти соотношения меняются от одной рассматриваемой задачи к другой, часто зависят от решений, принимаемых в ручном режиме, состояния транспортной инфраструктуры и прочих факторов. Некоторые могут быть описаны как дополнительные ограничения-неравенства вида

$$\begin{aligned} \sum_{J:\,(I,J)\in W} A^l_{I,J} x_{I,J} \leqslant b^l_I, \quad I \in S_l, \\ \sum_{I:\,(I,J)\in W} B^l_{I,J} x_{I,J} \leqslant b^l_J, \quad I \in D_l, \end{aligned} \tag{8}$$

и практика показывает, что $|S^e \cup D_e| \ggg |S^l \cup D_l|$. Следует также отметить важную роль условий неотрицательности $x_{I,J} \geqslant 0$, на первый взгляд выглядящих обманчиво простыми.

4.1. Проекционный подход к задаче линейной оптимизации. Если иметь в виду планируемое развитие построенной модели, то стоит рассмотреть некоторые новые подходы к решению задач линейной оптимизации большой размерности, которые можно настроить на применение к задаче (2)–(5), (6). Такое развитие мотивируется существенным ростом и усложнением модели при увеличении горизонта планирования, более детального описания перевозимых грузов, типов вагонов, технологий перевозки и других факторов, что выводит задачу из поля проблем, решаемых стандартным оптимизационным аппаратом. В связи с этим и возникла идея использования проекционных алгоритмов, достаточно хорошо себя зарекомендовавших в других областях, но для решения задач линейной оптимизации пока серьёзно не применявшихся.

Для упрощения обозначений определим и обозначим оператор ортогональной проекции:

$$p \downarrow X = \operatorname{argmin} \min_{z \in X} \|p - x\|, \tag{9}$$

т. е. $\min\limits_{x\in X} \|p-x\| = \|p-p \downarrow X\|$, где $\|\cdot\|$ обозначает стандартную евклидову норму. Пользуясь случаем, обозначим также через $|N|$ количество элементов конечного множества N некоторых объектов. Обозначим через $\mathrm{Cone}(D)$ коническую оболочку конечного множества $D = \{z^1, z^2, \ldots, z^m\}$ векторов некоторого векторного пространства.

В [12] было показано, что при не слишком ограничительных предположениях разрешимая задача линейной оптимизации (1) может быть решена единственной операцией проекции на её допустимое множество X. Чтобы сформулировать один из вариантов таких предположений, напомним стандартные определения допустимых и двойственных конусов для допустимого множества задачи линейной оптимизации.

Определение 1. Для $x \in X$ в (1) определим допустимый конус

$$K(x) = \{z \mid x + \lambda z \in X \text{ для некоторого } \lambda > 0\}. \tag{10}$$

Существенным является то, что для задач линейной оптимизации $K(x)$ является выпуклым замкнутым множеством, если он непуст.

Определение 2. Для $x \in X$ в (1) определим двойственный конус

$$K^\star(x) = \{z \mid xz \leqslant 0 \text{ для всех } x \in K(x)\}. \tag{11}$$

Если $x^\star$ — решение задачи (1), то двойственный конус $K^\star(x^\star)$ будем обозначать $K^\star$.

Простейшим предположением, гарантирующим то, что решение задачи (1) может быть получено единственной проективной операцией, является непустота внутренности $K^\star$. Из этого предположения в первую очередь элементарно следует, что в этом случае оптимальное решение единственно. Справедливо и обратное, что требует некоторых рассуждений. Проблема заключается в том, что в этом случае двойственные множители могут быть неединственны. Следующая лемма решает эту проблему.

Лемма 1. *Если решение $x^\star_J$ задачи* (1) *единственно, то $K^\star$ имеет непустую внутренность.*

Для доказательства рассмотрим задачу (1) и её решение $x^\star_J$ более подробно. Обозначим через S_e подмножество множества ограничений S такое, что

$$\sum_{J\colon (I,J)\in W} A_{I,J} x^\star_J = b_I, \quad I \in S_e \subset S. \tag{12}$$

Из единственности оптимального решения следует, что $|S_e| \geqslant |J|$. В противном случае можем сместиться из точки $x^\star$ в нетривиальном ортогональном подпространстве линейной оболочки A_I, $I \in S_e$, где для сокращения обозначений A_I обозначает I-ю строку матрицы A, сохраняя ограничения-равенства и оптимальность, что противоречит единственности. Заметим, что также согласно замечательному результату из [13] существует строго комплементарное решение линейной оптимизационной задачи (1) с ограничениями (7), (8) со строго положительными двойственными множителями u_I, $I \in S_e$, такими, что

$$c = \sum_{I \in S_e} u_I A_I. \tag{13}$$

Далее покажем, что существует $S'_e \subset S_e$ такое, что $A_I,\ I \in S'_e$, линейно независимы, $|S'_e| = n$ и есть строго положительные $u_I,\ I \in S'_e$, (двойственные множители) такие, что

$$c = \sum_{I \in S'_e} u_I A_I, \tag{14}$$

что влечёт их двойственную оптимальность. Такое множество S'_e может быть построено при помощи процесса последовательного исключения, описанного ниже.

Действительно, если $|S_e| > n$, то $A_I,\ I \in S_e$, линейно зависимы и существуют такие $\mu_I,\ I \in S_e$, не равные нулю, что

$$0_J = \sum_{I \in S_e} \mu_I A_I, \tag{15}$$

где 0_J — нуль-вектор соответствующей размерности, следовательно,

$$c = \sum_{I \in S_e} (u_I + t\mu_I) A_I \tag{16}$$

для любого t.

В этом случае можем выбрать t таким, что $u_{I'} + t\mu_{I'},\ I' \in S_e$, обращается в нуль по крайней мере для одного I' и t' и остальные $u_I + t\mu_I,\ I \neq I',\ I \in S_e$, остаются неотрицательными. В результате I' можно исключить из множества S_e (как и другие $I'' \in S_e$ такие, что $u_{I''} + t\mu_{I''} = 0$, по случайному совпадению), переопределить u_I, S_e и продолжить до тех пор, пока не получим $|S_e| = n$ с линейной независимостью $A_I,\ I \in S_e$.

В свою очередь, это гарантирует, что для любого $c'_J,\ J \in J$, достаточно близкого к c, в любой метрике будут существовать $u'_I > 0,\ I \in S_e$, такие, что $c' = \sum\limits_{I \in S_e} u'_I A_I$, следовательно, $c' \in K^\star$, таким образом, $K^\star$ имеет непустую внутренность.

Как показано в [12], при этом решение проективной задачи нахождения $(x^0 - \tau c) \downarrow X$, где X обозначает допустимое множество (1), даёт решение (1) для произвольного x^0 и достаточно большого $\tau > 0$, зависящего, конечно, от x^0. Другими словами, $(x^0 - \tau c) \downarrow X = x^\star$.

4.2. Реализация и численные эксперименты. Проекционные алгоритмы, особенно их альтернирующие варианты, весьма популярны для решения выпуклых задач допустимости. Развитие этих алгоритмов началось в 1940-х гг. [14], подробную библиографию можно найти в [15]. Их развитие продолжается и сейчас (см., например, [16] для обзора современного состояния этого направления).

Проблема поиска минимального расстояния до выпуклого множества (9) получила значительно меньше внимания, за исключением, пожалуй,

наиболее знаменитого случая [14], когда множество X представимо в виде алгебраической суммы двух множеств $X = X_1 + X_2$ и альтернирующие проекции на X_1 и X_2 легко реализуемы. Другой случай, когда удалось доказать конечную сходимость, это когда X представляет собой (почти) канонический симплекс [17, 18].

К сожалению, мы имеем дело с полиэдром X практически общего вида, однако возможно применить ряд преобразований, которые помогут существенно уменьшить размер вспомогательных задач и сделают возможным решение практических задач.

Преобразование проективной задачи. Первый шаг заключается в сведении задачи $p \downarrow X$ к задаче о поиске элемента минимальной нормы. Так как $p \downarrow A = (p - a) \downarrow (A - a) + a$ для любого a, то

$$(x^0 - \tau c) \downarrow X = 0 \downarrow (X - x^0 + \tau p) + x^0 - \tau c$$

и основная часть проблемы справа — это задача на поиск вектора минимальной нормы

$$0 \downarrow (X - x^0 + \tau p) = 0 \downarrow Q \tag{17}$$

для $Q = X - x^0 + \tau p$, смещённого полиэдра X. Это не меняет тип ограничений, которые описывают X, так что $Q = Q_e \cap Q_l$, где

$$Q_e = \Big\{ x \mid \sum_{J\colon (I,J) \in W} A^e_{IJ} x_J = b'^e_I,\ I \in S_e \Big\}, \tag{18}$$

$$Q_l = \Big\{ x \mid \sum_{J\colon (I,J) \in W} A^l_{IJ} x_J = b'^l_I,\ I \in S_l \Big\}, \tag{19}$$

которые отличаются от (7) и (8) только правыми частями b'^e_I, $I \in S_e$, и b'^l_I, $I \in S_l$.

Следующий шаг заключается в замене задачи о минимальной норме $0 \downarrow Q$ на проекцию специальной точки p^0 на полиэдральный конус, генерируемый строками матрицы ограничений (18), (19):

$$\min_{z \in \mathrm{Cone}(\overline{Q}),} \|z - p^0\|^2 = \|p^0 \downarrow \overline{Q} - p^0\|^2, \tag{20}$$

где $\overline{Q} = \mathrm{Cone}(\overline{Q}_e) + \mathrm{Cone}(\overline{Q}_l)$. Матрицы $\mathrm{Cone}(\overline{Q}_e), \mathrm{Cone}(\overline{Q}_l)$ представляют собой матрицы Q_e, Q_l с дополнительными столбцами $-b'^e_I$, $I \in S_e$, и $-b'^l_I$, $I \in S_l$:

$$\overline{Q}_e = \|Q_e \quad -b'^e\|, \quad \overline{Q}_l = \|Q_l \quad -b'^l\|. \tag{21}$$

Детали этого преобразования описаны в [19], здесь рассмотрим лишь декомпозиционные возможности, предоставляемые (20) для проекции на полиэдральный конус.

Декомпозиция задач линейной оптимизации большой размерности. Первая идея для упрощения задачи проектирования (20) с помощью декомпозиции основана на том, что конус $\mathrm{Cone}(\overline{Q}_e) = L_e$ является линейным подпространством и проекция на него представляет собой ряд стандартных операций линейной алгебры, для которых существуют эффективные реализации [4], учитывающие традиционную для линейной оптимизации разреженность данных.

Тогда, как было показано в [20], задача (20) может быть сведена к операции проекции $p_l^0 \downarrow \overline{Q}_{l,e}$, где

$$p_e^0 = p^0 \downarrow \overline{Q}_e, \quad \overline{Q}_{l,e} = \overline{Q}_l \downarrow \overline{Q}_e = Q_l \downarrow L_e,$$
$$\overline{Q}_l \downarrow \overline{Q}_e = \{z = q \downarrow \overline{Q}_e \mid q \in \overline{Q}_l\} = \overline{Q}_l \downarrow L_e.$$

Таким образом, по сути дела исключаются ограничения-равенства, составляющие большинство в транспортно-логистических задачах, и существенно сокращается размер задачи. Конечно, это требует дополнительных вычислительных затрат на проектирование всех $q \in \overline{Q}_l$ на L_e, но так как каждый такой вектор q может быть представлен как неотрицательная комбинация генераторов $\overline{Q}_l$:

$$\overline{Q}_l \downarrow L_e = \Big(\sum_I \mu_I \overline{Q}_{l,I} \Big) \downarrow L_e = \sum_I \mu_I (\overline{Q}_{l,I} \downarrow L_e) = \sum_I \mu_I z_I,$$

это сводится к проектированию всех строк $\overline{Q}_l$ на L_e.

Конечно, это всё ещё довольно вычислительно затратная операция, но её стоимость может быть существенно снижена, если $\overline{Q}_l$ имеет подходящую структуру.

Для выявления особенностей такой структуры, которые целесообразно использовать для ускорения вычислений, можно использовать некоторые подходы, основанные на теории графов.

4.3. Структурный анализ задач линейной оптимизации. Практические приложения. Для рассмотрения задачи линейной оптимизации с точки зрения теории графов полезно в первую очередь рассмотреть матрицу смежности ограничений, т. е. рассмотреть строки матриц ограничений как вершины некоторого графа. Две вершины-строки можно считать соединёнными ребром, если эти две строки одновременно имеют в каком-либо столбце ненулевые коэффициенты. Такой граф связи вершин-ограничений может распадаться на несколько непересекающихся связанных компонент, тогда операция проекции на такую матрицу существенно упрощается. Несомненно, эффект от такой декомпозиции зависит от характера разреженности матриц. Реальные задачи слишком велики для теоретического анализа, однако задача поиска связных компонент полиномиально разрешима и, следовательно, может эффективно

Таблица 2

Характеристики тестовых задач

N	Число строк	Число столбцов	Число ненулевых коэф-тов	Плотность, %	Число компонент
1	28935	5794061	—	—	—
2	39932	51090	99608	$4{,}88 \cdot 10^{-3}$	535
3	122347	16675027	31192305	$1{,}53 \cdot 10^{-3}$	32
4	361024	53404781	106608366	$5{,}42 \cdot 10^{-4}$	233

применяться даже для больших матриц. Далее мы представим некоторые примеры такого анализа для задач транспортной логистики.

Для примера выбрано четыре транспортно-логистические задачи, которые отличаются в горизонте планирования, количестве транспортных средств, маршрутов и пр. Характеристики этих моделей представлены в табл. 2. Последний столбец даёт число связных компонент. С одной стороны, оно достаточно велико, и это позволяет надеяться на то, что размеры компонент будут существенно меньше, чем размеры всей задачи. С другой стороны, число связных компонент не слишком велико, что позволяет использовать стандартные средства для параллельных вычислений.

Подробности распределения числа связных компонент как функции их размера отражены в табл. 3. Полученные данные демонстрируют большую изменчивость в количестве связных компонент, хотя эта изменчивость в основном происходит за счёт большого числа небольших компонент. Число относительно больших компонент во всех экспериментах

Таблица 3

Размер компоненты	Число таких компонент	Средняя доля ненулевых элементов
25592	1	$3{,}907 \cdot 10^{-5}$
1743	1	$5{,}737 \cdot 10^{-4}$
1596	1	$6{,}266 \cdot 10^{-4}$
1162	1	$8{,}606 \cdot 10^{-4}$
872	1	$1{,}147 \cdot 10^{-3}$
727	2	$1{,}376 \cdot 10^{-3}$
726	1	$1{,}377 \cdot 10^{-3}$
581	2	0,721
437	2	$2{,}288 \cdot 10^{-3}$
436	4	$2{,}294 \cdot 10^{-3}$
291	7	$3{,}436 \cdot 10^{-3}$
1	970	1,000

составляет порядка 10% от общего числа ограничений, что позволяет надеяться на положительный эффект от крупнозернистой декомпозиции.

Заключение

В практике управления крупными компаниями расширяется использование всё более масштабных математических моделей, которые уже выходят за пределы возможностей стандартных алгоритмов и программ. Соответственно возникает и потребность в квалифицированном использовании средств описания подобных моделей и алгоритмах, способных учитывать специфику структуры задачи и предлагать новые подходы к её декомпозиции. В данной работе предложены эффективные приёмы использования широко распространённого языка моделирования AMPL и исследован алгоритмический подход к решению слабоструктурированных оптимизационных задач большой размерности, основанный на применении проекционного оператора. Как показано в работе, эффективное использование языка AMPL позволяет существенно ускорить подготовку модели к расчёту, а декомпозиция проективных операторов создаёт новые возможности для параллелизма.

ЛИТЕРАТУРА

1. Gurobi optimizer reference manual. Beaverton: Gurobi Optimization, 2021. Available at `www.gurobi.com/documentation/9.5/refman/index.html` (accessed Feb. 27, 2022).

2. IBM ILOG CPLEX Optimizer. Armonk: IBM, 2022. Available at `www.ibm.com/analytics/cplex-optimizer` (accessed Feb. 27, 2022).

3. **Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A., Sorensen D.** LAPACK users' guide. Philadelphia: SIAM, 1999. 404 p.

4. SuiteSparse: A suite of sparse matrix software. College Station: Texas A&M Univ., 2022. Available at `people.engr.tamu.edu/davis/suitesparse.html` (accessed Feb. 27, 2022).

5. **Dantzig G. B., Wolfe Ph.** The decomposition algorithm for linear programming // Econometrica. 1961. V. 9, No. 4. P. 767–778.

6. AMPL homepage. Mountain View, CA: AMPL Optimization, 2022. Available at `ampl.com` (accessed Feb. 27, 2022).

7. **Fourer R., Gay D. M., Kernighan B. W.** AMPL: A modeling language for mathematical programming. Boston: Cengage Learning, 2003. 517 p.

8. GAMS — A user's guide. Frechen: GAMS Software, 2022. Available at `www.gams.com/35/docs/UG_MAIN.html` (accessed Feb. 27, 2022).

9. GLPK (GNU linear programming kit). Boston, MA: Free Software Found., 2012. Available at `www.gnu.org/software/glpk/` (accessed Feb. 27, 2022).

10. NEOS Server. Madison, WI: Univ. Wisconsin, 2022. Available at `neos-server.org/neos/` (accessed Feb. 27, 2022).

11. **Kahan G.** Walking through a columnar approach to linear programming of a business // Interfaces. 1982. V. 12, No. 3. P. 32–39.
12. **Nurminski E. A.** Single-projection procedure for linear optimization // J. Glob. Optim. 2016. V. 66, No. 1. P. 95–110.
13. **Goldman A. J., Tucker A. W.** Theory of linear programming, linear inequalities and related systems. V. 38. Princeton, NJ: Princeton Univ. Press, 1956. P. 53–97.
14. **Von Neumann J.** On rings of operators. Reduction theory // Ann. Math. 1949. V. 50, No. 2. P. 401–485.
15. **Bauschke H. H., Borwein J. M.** On the convergence of von Neumann's alternating projection algorithm for two sets // Set-Valued Anal. 1993. No. 1. P. 185–212.
16. **Escalante R., Raydan M.** Alternating projection methods. Philadelphia: SIAM, 2011.
17. **Michelot C.** A finite algorithm for finding the projection of a point onto the canonical simplex of E^n // J. Optim. Theory Appl. 1986. V. 50, No. 1. P. 195–200.
18. **Малоземов В. Н., Тамасян Г. Ш.** Два быстрых алгоритма проектирования точки на стандартный симплекс // Журн. вычисл. математики и мат. физики. 2016. Т. 56, вып. 5. С. 742–755.
19. **Нурминский Е. А.** Проекция на внешне заданные полиэдры // Журн. вычисл. математики и мат. физики. 2008. Т. 48, вып. 3. С. 387–396.
20. **Nurminski E. A., Shamray N. B.** Row-oriented decomposition in large-scale linear optimization // Optimization and Applications. Proc. Int. Conf. OPTIMA 2021 (Petrovac, Montenegro, Sep. 27–Oct. 1, 2021). Heidelberg: Springer, 2021. P. 50–63. (Lect. Notes Comput. Sci.; V. 13078).
21. **Greenberg H. J., Lundgren J. R., Maybee J. S.** Graph theoretic methods for the qualitative analysis of rectangular matrices // SIAM J. Algebraic Discrete Methods. 1981. V. 2, No. 3. P. 227–239.

Нурминский Евгений Алексеевич
Шамрай Наталья Борисовна

Статья поступила
2 мая 2022 г.

После доработки —
2 мая 2022 г.

Принята к публикации
5 мая 2022 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
July–September 2022. Vol. 29, No. 3. P. 64–84

UDC 519.863 DOI 10.33048/daio.2022.29.736

MODELING AND OPTIMIZING LARGE-SCALE PRODUCTION-LEVEL TRANSPORTATION SYSTEMS

E. A. Nurminskiy[1, a] and *N. B. Shamray*[2, b]

[1] Far Eastern State University,
10 Ayaks Bay, 690922 Vladivostok, Russia

[2] Institute of Automation and Control Processes FEB RAS,
5 Radio Street, 690041 Vladivostok, Russia

E-mail: [a] `nurminskiy.ea@dvfu.ru`, [b] `shamray@dvo.ru`

Abstract. Large-scale economic modeling is becoming a reality for major businesses and it pushes their analytic and planning departments into very complicated areas of big data analytics and control. At the same time, it demands research communities in academia and else to develop adequate tools to operate models with millions of variables and gigabytes of data, where traditional off-the-shelf solutions fail. In this paper, we describe our experience with one rather common high-dimensional logistic problem and some of the mathematical and computational ideas we pursue to deal with it. Tab. 3, bibliogr. 21.

Keywords: large-scale economic modeling, production-level transport expedition system, linear optimization, projection algorithm.

REFERENCES

1. Gurobi Optimizer Reference Manual (Gurobi Optimization, Beaverton, 2021). Available at `www.gurobi.com/documentation/9.5/refman/index.html` (accessed Feb. 27, 2022).

2. IBM ILOG CPLEX Optimizer (IBM, Armonk, 2022). Available at `www.ibm.com/analytics/cplex-optimizer` (accessed Feb. 27, 2022).

3. **E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney,** and **D. Sorensen,** *LAPACK Users' Guide* (SIAM, Philadelphia, 1999).

This research is supported by the Ministry of Science and Higher Education of Russia (Project 075–02–2022–880).

English version: Journal of Applied and Industrial Mathematics **16** (3) (2022).

4. SuiteSparse: A Suite of Sparse Matrix Software (Texas A&M Univ., College Station, 2022). Available at `people.engr.tamu.edu/davis/suitesparse.html` (accessed Feb. 27, 2022).

5. **G. B. Dantzig** and **Ph. Wolfe,** The decomposition algorithm for linear programming, *Econometrica* **9** (4), 767–778 (1961).

6. AMPL Homepage (AMPL Optimization, Mountain View, CA, 2022). Available at `ampl.com` (accessed Feb. 27, 2022).

7. **R. Fourer, D. M. Gay,** and **B. W. Kernighan,** *AMPL: A Modeling Language for Mathematical Programming* (Cengage Learning, Boston, 2003).

8. GAMS — A User's Guide (GAMS Software, Frechen, 2022). Available at `www.gams.com/35/docs/UG_MAIN.html` (accessed Feb. 27, 2022).

9. GLPK (GNU Linear Programming Kit) (Free Software Found., Boston, MA, 2012). Available at `www.gnu.org/software/glpk/` (accessed Feb. 27, 2022).

10. NEOS Server (Univ. Wisconsin, Madison, WI, 2022). Available at `neos-server.org/neos/` (accessed Feb. 27, 2022).

11. **G. Kahan,** Walking through a columnar approach to linear programming of a business, *Interfaces* **12** (3), 32–39 (1982).

12. **E. A. Nurminski,** Single-projection procedure for linear optimization, *J. Glob. Optim.* **66** (1), 95–110 (2016).

13. **A. J. Goldman** and **A. W. Tucker,** *Theory of Linear Programming, Linear Inequalities and Related Systems. Vol. 38* (Princeton Univ. Press, Princeton, NJ, 1956). P. 53–97.

14. **J. von Neumann,** On rings of operators. Reduction theory, *Ann. Math.* **50** (2), 401–485 (1949).

15. **H. H. Bauschke** and **J. M. Borwein,** On the convergence of von Neumann's alternating projection algorithm for two sets, *Set-Valued Anal.* **1**, 185–212 (1993).

16. **R. Escalante** and **M. Raydan,** *Alternating Projection Methods* (SIAM, Philadelphia, 2011).

17. **C. Michelot,** A finite algorithm for finding the projection of a point onto the canonical simplex of E^n, *J. Optim. Theory Appl.* **50** (1), 195–200 (1986).

18. **V. N. Malozemov** and **G. Sh. Tamasyan,** Two fast algorithms for projecting a point onto the canonical simplex, *Zh. Vychisl. Mat. Mat. Fiz.* **56** (5), 742–755 (2016) [Russian] [*Comput. Math. Math. Phys.* **56** (5), 730–743 (2016)].

19. **E. A. Nurminski,** Projection onto polyhedra in outer representation, *Zh. Vychisl. Mat. Mat. Fiz.* **48** (3), 387–396 (2008) [Russian] [*Comput. Math. Math. Phys.* **48** (3), 367–375 (2008)].

20. **E. Nurminski** and **N. Shamray,** Row-oriented decomposition in large-scale linear optimization, in *Optimization and Applications* (Proc. Int. Conf. OPTIMA 2021, Petrovac, Montenegro, Sep. 27–Oct. 1, 2021) (Springer, Heidelberg, 2021), pp. 50–63 (Lect. Notes Comput. Sci., Vol. 13078).

21. H. J. Greenberg, J. R. Lundgren, and **J. S. Maybee,** Graph theoretic methods for the qualitative analysis of rectangular matrices, *SIAM J. Algebraic Discrete Methods* **2** (3), 227–239 (1981).

Evgeny A. Nurminskiy
Natalia B. Shamray

Received May 2, 2022
Revised May 2, 2022
Accepted May 5, 2022