

ТЕСТИРОВАНИЕ АЛГОРИТМОВ ДЛЯ ЦЕЛОЧИСЛЕННЫХ КВАЗИБЛОЧНЫХ ЗАДАЧ ОПТИМИЗАЦИИ

Д.В. Ковков¹

dmkovkov@gmail.com

Д.В. Лемтюжникова²

darabbt@gmail.com

¹ Вычислительный центр им. А.А. Дородницына ФИЦ «Информатика и управление» РАН, Москва, Российская Федерация

² Московский авиационный институт (национальный исследовательский университет), Москва, Российская Федерация

Аннотация

Рассмотрены алгоритмы для решения целочисленных квазиблочных задач оптимизации. Проанализированы современные методы декомпозиции, а также необходимость применения локального элиминационного алгоритма для задач большой размерности. Рассмотрены особенности применения параметрической оптимизации. Проведены тестовые эксперименты решения задач целочисленного линейного программирования большой размерности для точных, приближенных и эвристических алгоритмов. Получены результаты для различных модификаций локального элиминационного алгоритма. Приведены эксперименты для распараллеливания локального элиминационного алгоритма с помощью GRID-технологий. Показаны примеры задач, которые не могут быть решены без применения технологии распараллеливания

Ключевые слова

Декомпозиция, целочисленное программирование, квазиблочная структура, локальный элиминационный алгоритм

Поступила в редакцию 20.04.2017
© МГТУ им. Н.Э. Баумана, 2018

Введение. Рассмотрены оптимизационные задачи с матрицами квазиблочной структуры с целочисленными переменными. Эти задачи являются результатом применения критерия для локального элиминационного алгоритма (ЛЭА) О.А. Щербины [1] и имеют блочно-лестничную или блочно-древовидную структуру. Такие структуры формируются после соответствующих процедур перестановки столбцов [2] для задач с разреженными матрицами.

Особенность ЛЭА — использование локальной информации об окрестностях так называемых связывающих переменных, т. е. переменных, принадлежащих одновременно нескольким окрестностям. Поэтому ЛЭА позволяет получать глобальное решение задачи с помощью локальных вычислений, которые обычно являются решениями соответствующих подзадач. В общем виде ЛЭА представляется как последовательное исключение переменных с сохранением информации о них. Процедура ЛЭА состоит из прямой и обратной частей. В прямой части переменные исключаются в соответствии с заданным порядком, при этом вычисляемая информация о каждой переменной сохраняется в виде

локальных решений (чаще всего, таблиц), а в завершении процесса вся полученная информация представляется в виде значения критерия. В обратной части находится глобальное решение исходной задачи по определенным в прямой части таблицам с локальными решениями, которые обеспечивают достижение критерия, представленного в прямой части. В настоящей статье ЛЭА будет рассмотрено в основном как класс бинарных задач динамической оптимизации, а именно булевых задач линейного программирования:

$$\text{extr } \{cx \mid Ax \sim b, x_j \in \{0,1\}, j \in N\}.$$

Здесь c — вектор целевой функции; A — матрица ограничений; b — вектор ограничений; \sim — одно из соотношений $\{<, \leq, =, \geq, >\}$.

В работе представлены результаты численного тестирования, которые позволяют выявить специфику применения рассматриваемого подхода.

Обзор методов декомпозиции. Методы декомпозиции для блочных задач целочисленного программирования, использовавшиеся до 1980-х гг., изучены в работе [3]. Ниже предложен краткий обзор методов, применяющихся в настоящее время. Разумеется, он не претендует на полноту, на взгляд авторов представлены некоторые оригинальные подходы.

Для задач стохастического смешанного целочисленного программирования авторы работы [4] анализируют алгоритм двойственной декомпозиции Каро и Шульца с позиции возможностей параллельного решения. Предложена формулировка задачи, которая позволяет устранить существенное препятствие для эффективного распараллеливания главной задачи, для чего использованы решатели, основанные на методах внутренней точки, учитывающие структуру задачи. Результаты этой работы демонстрируют высокий потенциал для ускорения с помощью распараллеливания и важность регуляризации (стабилизации) для задач двойственной оптимизации. Неравномерность распределения нагрузки остается единственным препятствием для оптимальной масштабируемости подзадач. В работе [5] рассмотрено применение модели задачи смешанного целочисленного программирования для решения задач управления планированием использования рабочей силы и ее возможной транспортировки. В данной задаче множество исполнителей назначается на выполнение заданий, места выполнения которых распределены географически. Такая постановка возникает в сценариях планирования работы социальных работников в Великобритании. Авторы представляют задачу смешанного целочисленного программирования, которая отражает важнейшие особенности практической задачи, такие как наличие определенных географических районов и приспособляемость исполнителей (их доступность и различную мобильность). В работе показано, что на качество итогового решения влияет порядок, в котором решаются подзадачи. Поэтому авторы исследуют различные способы упорядочивания подзадач и показывают, что весьма перспективным является подход на основе метода декомпозиции, который позволяет получить решение высокого качества за разумное время, и точного метода оптимизации.

В работе [6] было обнаружено, что в задачах назначения и планирования программное обеспечение для решения задач на основе смешанного целочисленного программирования стало конкурировать или превосходить подходы, основанные на методе Бендерса (использующего логическую декомпозицию задачи). Приведенные в этой работе реализации метода декомпозиции Бендерса тем не менее отличаются от описанных в цитированной авторами литературе. Это приводит к тому, что производительность решения оказывается существенно ниже, чем указано в источниках. В работе [7] было установлено, что для задач большей размерности при корректной реализации метода Бендерса он остается на 2–3 порядка быстрее, чем самые новейшие коммерческие системы решения задач на основе подходов смешанного целочисленного программирования, что полностью делает противоположными выводы работы [6].

В работе [8] предложена архитектура для решения задач построения оптимальных траекторий, декомпозиция в которой устроена следующим образом: глобальная задача обхода препятствий разбивается на простые подзадачи, соответствующие гомотопиям различных путей. В классических подходах к планированию траекторий на основе гомотопий планирование траектории и идентификация гомотопий проводятся одновременно, что приводит к большой вычислительной сложности. В этой работе предложен метод, позволяющий перечислять и явно выражать различные классы гомотопий до шага планирования траектории или оптимизации, что дает возможность расщепить задачу на более простые независимые подзадачи. В работе изложено два интересных результата. Первый результат — описание метода, который использует известные методы ячеевидной декомпозиции (*cell decomposition*), чтобы перечислить и описать все созданные локальные задачи, которые могут быть решены эффективно и независимо. Кроме того, проанализировано отношение между предложенными представлениями ячейка–последовательность (*cell-sequence representation*) и классами гомотопий. Второй результат представляет собой новый, вычислительно эффективный метод решения задачи оптимизации траекторий в виде последовательности ячеек, основанный на смешанном целочисленном квадратичном программировании. С помощью симуляции показаны вычислительная эффективность и большое множество решений. Предложенная формулировка задачи смешанного целочисленного квадратичного программирования хорошо вписывается в класс подходов на основе предсказаний для линейных моделей с невыпуклыми ограничениями на отсутствие столкновений.

Класс двухэтапных задач стохастического целочисленного программирования, где на первом этапе переменные являются бинарными, а на втором — целочисленными общего вида, рассмотрен в работе [9]. Разработан алгоритм декомпозиции, аналогичный методам L -разбиения и Бендерса, где с помощью сечения Гомори итеративно получают все более и более точные приближения решений целочисленных задач второго этапа. Авторы показывают, что предложенная методология является гибкой, допуская различные способы реализации, каждый из которых даст алгоритм, сходящийся за конечное число шагов. Авторы описывают

свои алгоритмы, используя примеры из литературы. В работе также приведены вычислительные результаты для нескольких случаев стохастической задачи местонахождения сервера (*server location problem*), которые показывают, что указанный алгоритм, основанный на декомпозиции, лучше масштабируется по числу сценариев, чем другой современный алгоритм решения/решатель, использованный для решения эквивалентной детерминированной задачи.

Новый метод отсекающих плоскостей для двухэтапных задач стохастического смешанного целочисленного программирования, названный декомпозицией Фенхеля (*Fenchel decomposition, FD*), предложен в работе [10]. Декомпозиция Фенхеля использует класс выполняющихся неравенств, называемый отсечения Фенхеля, которые получаются с помощью отсекающих плоскостей Фенхеля в задачах целочисленного программирования. Сначала авторы получают отсечения Фенхеля, использующие переменные как первого, так и второго этапов, формулируют алгоритм на основе отсечений Фенхеля для задач стохастического смешанного целочисленного программирования и показывают сходимость за конечное время для случая бинарного первого этапа. В работе [10] также определены отсечения Фенхеля, содержащие только переменные второго этапа, и использованы идеи дизъюнктивного программирования (*disjunctive programming*) для расширения отсечений в пространстве большей размерности, включающего в себя переменные первого этапа. Затем дан альтернативный алгоритм (*FD-L*), основанный на расширенных отсечениях. В завершение приведены результаты вычислительных экспериментов для нескольких тестовых задач, известных из литературы, имеющих специальную структуру задачи о рюкзаке с неотрицательными коэффициентами в левой части. Результаты выглядят перспективно, в случае больших размерностей оба алгоритма превосходят прямой алгоритм решения и прямой алгоритм дизъюнктивного программирования (*disjunctive decomposition*). Кроме того, алгоритм *FD-L* в целом показывает лучшие результаты, чем алгоритм *FD*. Поскольку отсечения Фенхеля в общем случае могут быть вычислительно сложными и лучше всего приспособлены для задач со специальной структурой, оба алгоритма используют специальную структуру тестовых задач, уменьшая размерность задач генерации отсечений, основываясь на числе ненулевых компонент в нецелочисленном решении, которые необходимо отсечь.

В работе [11] речь идет о том, что объединение задач планирования выпуска продукции и динамической оптимизации может увеличить общую производительность многопродуктовых реакторов непрерывного действия с механическим перемешиванием. Однако такое объединение приводит к появлению смешанной целочисленной задачи динамической оптимизации большой размерности, которая может оказаться очень сложной в решении. Здесь предложено два эффективных метода решения на основе обобщенного метода декомпозиции Бендерса, который использует специальную структуру объединенной задачи. Первый метод основан на подходе с разделением по времени, согласно парадигме *Master-workers*. После декомпозиции главная задача представляет собой набор отделившихся задач динамической оптимизации (*worker problems*) и координирующую задачу

(*master problem*), которая является задачей смешанного целочисленного нелинейного дробного программирования. Затем координирующая задача решается до нахождения глобально оптимального решения методами дробного программирования, что гарантирует допустимость отсечений Бендерса. Вторым методом декомпозиции применяется к задаче вычисления оптимальной последовательности производства. Аналогично первому методу во втором используется алгоритм дробного программирования для решения координирующей задачи. По сравнению с одновременным методом в задаче оптимизации производственного процесса в многопродуктовых реакторах непрерывного действия с механическим перемешиванием предложенные алгоритмы декомпозиции могут уменьшить время вычисления на два порядка и более.

Диффузионные процессы в производстве полупроводников, представляющие собой последовательность процессов от очистки пластины до процесса отжига, рассмотрены в работе [12]. Большинство печей осуществляет обработку поступающих пластин крупными партиями за одну процедуру, и этот процесс характеризуется сравнительно большими затратами по сравнению с другими технологическими процессами полупроводниковых производств. Строгое ограничение по времени связывает процесс очистки пластины и процессы отжига для дальнейшего контроля качества. Эти производственные ограничения на диффузионные процессы делают задачи составления расписаний очень сложными. В работе [12] предложен улучшенный подход к задачам составления расписаний на основе понятия скользящего горизонта планирования. В силу комбинаторной природы задачи планирования сложность задачи экспоненциально возрастает при увеличении числа заданий и инструментов. Однако в практических случаях время вычислений должно быть ограничено, поскольку в большинстве полупроводниковых производств требуется, чтобы расписание обновлялось за короткий интервал времени. Авторы работы [12] предлагают модель оптимизации диффузионных процессов на основе задачи смешанного целочисленного линейного программирования, а также эффективный метод декомпозиции этой сложной задачи. Предлагаемый метод декомпозиции повторяет процесс составления расписания по мере того, как он постепенно увеличивает число запусков, что позволяет алгоритму планирования создавать расписания, близкие к оптимальным на ограниченных интервалах времени. Алгоритм планирования может существенно увеличить такие ключевые показатели производительности, как частоты нарушений ограничений по времени, размеры партий, пропускная способность. В этой работе также рассмотрена программная архитектура для реализации алгоритма планирования.

Новый подход для точного решения задач математического программирования с вероятностными ограничениями (*chance-constrained mathematical programs*), которые имеют дискретные распределения с конечным носителем и случайными ограничениями типа многогранника, предложен авторами работы [13]. Такие задачи известны своей чрезвычайной сложностью за счет невыпуклости области допустимых решений. Большинство известных методов может всего лишь найти

доказуемо хорошие решения в некоторых весьма узких частных случаях. Предлагаемый подход использует декомпозицию для получения подзадач, каждая из которых соответствует одному возможному исходу, и методы целочисленного программирования для объединения результатов решения подзадач, а затем получения детерминированных сильных выполняющихся неравенств. Вычислительные эксперименты для задачи математического программирования с вероятностными ограничениями относительно задачи планирования ресурсов в задаче оптимизации персонала контактного центра показывают, что предложенный подход работает значительно лучше, чем существующие постановки на основе смешанного целочисленного программирования, а также чем простой метод декомпозиции, который не использует детерминированных неравенств. Авторы также показывают, как данный подход может применяться для эффективного нахождения последовательности уровней риска в приложении к задаче нахождения эффективной зависимости между риском и стоимостью.

Метод объединения алгоритмов прогрессивного хеджирования (*Progressive Hedging, PH*) и двойственной декомпозиции (*Dual Decomposition, DD*, Каро и Шульц) для задач стохастического смешанного целочисленного программирования предложен в работе [14]. Используя соответствие между нижними оценками, полученными в алгоритмах *PH* и *DD*, найден метод преобразования весов из алгоритма *PH* в множители Лагранжа. Быстрый прогресс на начальных итерациях алгоритма *PH* ускоряет сходимость алгоритма *DD* к точному решению. В работе приведены результаты вычислительных экспериментов для задач о положении сервера или единичных обязательствах (*unit commitment*).

Алгоритм декомпозиции, который сочетает декомпозицию Бендерса и основанную на сценарии лагранжеву декомпозицию, в применении к двухэтапной задаче стохастического программирования о планировании инвестиции с полным регрессом описан в работе [15]. Переменные первого этапа являются смешанно-целочисленными, а второго — непрерывными. Алгоритм основан на схеме кросс-декомпозиции (перекрестной декомпозиции) и полностью совместно использует информацию из прямой и двойственной задач, в части добавленных прямодвойственных мультиотсечений (разрезов) в координирующие лагранжеву задачу и задачу Бендерса, для всех сценариев. Преимущества схемы перекрестной декомпозиции показаны на наглядном примере задачи о расположении производства в условиях риска сбоя.

Отметим также несколько работ по задачам транспортного типа, которые содержат разреженные матрицы [16–21].

Далее представлены результаты по тестовым вычислениям, которые получены вторым автором настоящей работы.

Эффективность использования локального элиминационного алгоритма. Формальный критерий задачи оптимизации имеет вид

$$f(X) = \sum_{j=1}^m c_j x_j \rightarrow \min \text{ с ограничениями;}$$

$$\sum_{j=1}^m a_{ij}x_j \leq b_i, \quad i = 1, 2, \dots, n;$$

$$x_j \in \{0; 1\}, \quad j = 1, 2, \dots, m,$$

где $x = (x_1, x_2, \dots, x_m) \in X$ — множество переменных.

Решить задачу, означает найти оптимальное значение целевой функции для заданных ограничений.

Далее будут представлены результаты численных расчетов, которые характеризуют эффективность ЛЭА, т. е. время работы этого алгоритма существенно меньше, чем если задача решается напрямую с помощью стандартного метода целочисленного линейного программирования (ЦЛП).

Таблица 1

Результаты сравнения времени решений, полученных с помощью алгоритмов *Symphony* без ЛЭА, *Symphony* с ЛЭА и *Symphony* с ЛЭА с применением постоптимального анализа в зависимости от различных параметров

| Число ограничений m | Число переменных задачи n | Число блоков соответствующей структуры k | Максимальный сепаратор s | Алгоритм | | |
|-----------------------|-----------------------------|--|----------------------------|-----------------|-----------------------|--|
| | | | | <i>Symphony</i> | <i>Symphony</i> + ЛЭА | <i>Symphony</i> + ЛЭА + постоптимальный анализ |
| 50 | 150 | 25 | 2 | 1,5722 | 0,0301 | 0,0316 |
| 100 | 300 | 50 | 4 | 3,3821 | 0,1588 | 0,1579 |
| 150 | 500 | 75 | 6 | – | 0,0765 | 0,0774 |
| 200 | 800 | 100 | 8 | – | 0,0367 | 0,0395 |
| 250 | 1000 | 125 | 10 | – | 0,5681 | 0,5772 |

В ходе эксперимента все тестовые задачи ЦЛП с бинарными переменными имели блочно-древовидную структуру, сгенерированную искусственно. Все блоки в отдельно взятой задаче имели одинаковое число переменных, а также сепараторы одинакового размера внутри каждой задачи. Размеры сепараторов для одних и тех же параметров задачи изменялись для получения оценки времени решения задачи при увеличении размера сепаратора. Тестовые задачи ЦЛП генерировались исходя из заданного общего числа переменных, связывающих переменных между блоками, а также числа ограничений. Размеры и число блоков вычислялись по числу переменных и ограничений исходной задачи. С помощью генератора случайных чисел задавались остальные компоненты задачи: коэффициенты целевой функции, коэффициенты матрицы ограничений и правых частей для каждого блока. Каждая тестовая задача решалась с использованием следующих алгоритмов. Первый алгоритм — базовый решатель *Symphony* [22] для задач частично-целочисленного линейного программирования (ЧЦЛП). Второй алгоритм — ЛЭА, который использовал решатель *Symphony* для решения подзадач, соответствующих блокам. Третий алгоритм —

вариация второго алгоритма, где решатель *Symphony* применен с поддержкой технологии постоптимального анализа (теплого старта (ТС)).

Результаты описанного выше эксперимента приведены в табл. 1. Прочерк ставился, если время решения задачи составляет более двух часов. Было установлено явное преимущество второго и третьего алгоритмов. Эксперимент показал, что второй алгоритм становился менее эффективным вследствие увеличения объема перебора при решении подзадач в блоках, если увеличивать число связывающих переменных в задачах с одинаковым числом переменных и размером блоков. В этом случае разумно использовать третий алгоритм, а именно ЛЭА в сочетании с решателем *Symphony*, когда применяются принципы параметрической оптимизации (технология ТС). Дело в том, что соответствующие одному и тому же блоку задачи ЦЛП отличаются одна от другой только правыми частями для различных значений связывающих переменных. Алгоритм получает информацию о решении и использует ее для анализа последующих задач, что позволяет решать каждую задачу не полностью, а частично при переборе значений связывающих переменных. Таким образом, с помощью параметрического программирования можно существенно увеличить производительность ЛЭА. Однако результат эксперимента оказался неоднозначным. Время решения большинства задач с использованием второго и третьего алгоритмов практически сопоставимо. Для некоторых тестовых задач параметрическое программирование оказалось неэффективным. Отмечено, что при малых размерностях эффективность ЛЭА отсутствует.

Приближенный и эвристический подходы. Один из недостатков ЛЭА — полный перебор по множествам связывающих переменных, что обуславливает возможность большого объема перебора при большом числе связывающих переменных. Преодолеть этот недостаток ЛЭА и способствовать успешному решению практически важных задач динамической оптимизации специальной структуры с большим числом связывающих переменных могло бы введение процедуры, позволяющей «предсказать» искомые оптимальные (или близкие к оптимальным) значения связывающих переменных. Зная значения связывающих переменных, задачи динамической оптимизации, соответствующие блокам (окрестностям), можно было бы решать отдельно, независимо друг от друга. Впервые эта идея была введена О.А. Щербиной [23]. Однако возможность узнать каким-то образом именно оптимальные значения связывающих переменных представляется сомнительной; тем не менее можно найти значения связывающих переменных, близкие к оптимальным, с помощью замены задач динамической оптимизации, соответствующих блокам, их релаксациями [24] и последующего решения построенной таким способом задачи динамической оптимизации со специальной структурой с помощью ЛЭА. Найденные для релаксированных блоков оптимальные значения связывающих переменных (все возможные наборы) перебираются полностью, однако трудоемкость выполнения каждого шага существенно снижена за счет решения не исходных, а релаксированных — более легко решаемых подзадач. В качестве релаксаций могут приме-

няться, например, линейная релаксация, когда условия $x_j = \{0;1\}$ заменяются неравенствами $0 \leq x \leq 1$ и в результате получается задача линейного программирования. Таким образом, имеет место задача частично-целочисленного программирования. Она решается итеративно, где имеется независимое решение подзадач для отдельных целочисленных блоков. Непрерывное решение затем округляется.

Цель вычислительного эксперимента — провести оценку эффективности использования эвристического ЛЭА (ЭЛЭА) и «жадного» алгоритма, а также сравнить перечисленные выше приближенные алгоритмы с точным алгоритмом. Все задачи для вычислительного эксперимента являются задачами ЦЛП блочно-лестничной структуры с булевыми переменными, все блоки в отдельно взятой задаче имеют одинаковое число переменных и ограничений, а также одинаковое число переменных в сепараторах между ними. Число переменных в сепараторах варьировалось для одних и тех же параметров задачи, что позволило оценить эффективность каждого приближенного алгоритма при увеличении размеров сепараторов. Тестовые задачи генерировались по заданному числу переменных, числу ограничений и размеру сепараторов между блоками. Исходя из числа переменных и ограничений определялись размеры блоков, а также их число. Далее, с помощью датчика случайных чисел генерировались коэффициенты целевой функции, а также коэффициенты матрицы ограничений и правых частей для каждого блока. Библиотека *LES (Local Elimination Solver)* версии v0.1.1 [25] используется для вычисления задач бинарного целочисленного линейного программирования (*BILP problem*), например задачи о рюкзаке (*Knapsack problem*). «Жадный» алгоритм реализован в функции *start* класса *GreedyDriver*.

Таблица 2

**Результаты сравнения эвристического ЛЭА (числитель)
и «жадного» алгоритма (знаменатель)**

| Число ограничений m | Число переменных задачи n | Число блоков соответствующей структуры k | Максимальный сепаратор s | Точность ¹ | Ускорение ² | Число ошибок ³ |
|-----------------------|-----------------------------|--|----------------------------|-----------------------|------------------------|---------------------------|
| 100 | 300 | 10 | 6 | 99,31/98,67 | 12,12/36,56 | 3/6 |
| 100 | 300 | 10 | 10 | 98,40/97,71 | 12,52/495,46 | 11/9 |
| 100 | 300 | 10 | 14 | 99,02/97,28 | 4,50/3262,37 | 11/14 |
| 100 | 600 | 10 | 6 | 99,85/98,99 | 11,34/23,59 | 1/7 |
| 100 | 600 | 10 | 10 | 99,51/98,16 | 64,26/1250,81 | 4/12 |
| 100 | 600 | 10 | 14 | 99,59/97,54 | 3,55/1267,05 | 4/10 |
| 200 | 500 | 10 | 6 | 99,49/98,26 | 557,13/1059,74 | 4/7 |
| 200 | 500 | 10 | 10 | 99,34/98,61 | 71,37/922,73 | 8/8 |
| 200 | 500 | 10 | 14 | 98,72/98,174 | 3,82/955,63 | 12/14 |

¹ Точность приближенных алгоритмов относительно точного ЛЭА.
² Во сколько раз приближенный алгоритм работает быстрее, чем ЛЭА.
³ Сколько переменных из сепараторов было определено неверно.

Результаты тестовых задач следующие. Задачи из первой группы содержали 300 переменных и 100 ограничений, задачи из второй группы — 600 переменных и 100 ограничений, задачи из третьей группы — 500 переменных и 200 ограничений. В каждой задаче было 10 блоков. Размеры сепараторов в каждой группе задач — 6, 10 и 14 переменных. Удалось получить ускорение вычислений относительно точного ЛЭА. Для первой группы эвристический алгоритм решает задачи с сепараторами из шести переменных в 12,12 раза быстрее, задачи с сепараторами из 10 переменных — в 12,52 раза быстрее, а задачи с сепараторами из 14 переменных — в 4,5 раза быстрее, чем точный ЛЭА. «Жадный» алгоритм решает задачи с сепараторами из шести переменных в 36,56 раза быстрее, задачи с сепараторами из 10 переменных — в 495,46 раза быстрее, а задачи с сепараторами из 14 переменных — в 3262,37 раза быстрее, чем точный ЛЭА. При этом из всех сепараторов для решения, полученного с помощью эвристического алгоритма, в первом случае не совпало три значения, во втором и в третьем — 11 значений. Для решения, полученного с помощью «жадного» алгоритма, в первом случае не совпало шесть значений, во втором — девять значений, а в третьем — 14 значений сепараторов. Для второй группы эвристический алгоритм решает задачи с сепараторами из шести переменных в 11,34 раза быстрее, задачи с сепараторами из 10 переменных — в 64,26 раза быстрее, а задачи с сепараторами из 14 переменных — в 3,55 раза быстрее, чем точный ЛЭА. «Жадный» алгоритм решает задачи с сепараторами из шести переменных в 23,59 раза быстрее, задачи с сепараторами из 10 переменных — в 1250,81 раза быстрее, а задачи с сепараторами из 14 переменных — в 1267,05 раза быстрее, чем точный ЛЭА. При этом из всех сепараторов для решения, полученного с помощью эвристического алгоритма, в первом случае не совпало одно значение, во втором и в третьем — четыре значения. В то время как для решения, полученного с помощью «жадного» алгоритма, в первом случае не совпало семь значений, во втором — 12 значений, а в третьем — 10 значений сепараторов. Для третьей группы эвристический алгоритм решает задачи с сепараторами из шести переменных в 557,13 раза быстрее, задачи с сепараторами из 10 переменных — в 71,37 раза быстрее, а задачи с сепараторами из 14 переменных — в 3,82 раза быстрее, чем точный ЛЭА. «Жадный» алгоритм решает задачи с сепараторами из шести переменных в 1059,74 раза быстрее, задачи с сепараторами из 10 переменных — в 922,73 раза быстрее, а задачи с сепараторами из 14 переменных — в 955,63 раза быстрее, чем точный ЛЭА. При этом из всех сепараторов для решения, полученного с помощью эвристического алгоритма, в первом случае не совпало четыре значения, во втором — восемь значений, а в третьем — 12 значений. Для решения, полученного с помощью «жадного» алгоритма, в первом случае не совпало семь значений, во втором — восемь значений, а в третьем — 14 значений сепараторов.

Анализ полученных результатов показал, что при достаточно больших сепараторах между блоками квазиблочной задачи ЦЛП приближенные ЛЭА в сочетании с современными решателями позволяют решать задачи быстрее, чем

используемые точные алгоритмы при решении всей задачи. Кроме того, с увеличением размерности задач ускорение «жадного» и эвристического алгоритма становится более сопоставимым. При этом эвристический алгоритм намного чаще «угадывает» значения сепараторов, чем «жадный» алгоритм, что позволяет точно решить соответствующую подзадачу. Следовательно, можно предположить, что для задач с многократно большими размерностями более эффективным для использования представляется ЭЛЭА.

Распараллеливание блочных задач. Наличие блоков в рассматриваемых постановках с блочно-древовидной и блочно-лестничной структурой и концепция ЛЭА о независимом решении оптимизационных задач для отдельных блоков позволяет осуществлять параллельные решения на различных процессорах одновременно. Для этой цели используют ГРИД-систему с гибким распределением свободных вычислительных ресурсов [26]. Указанная система работает по следующему принципу. Имеется центральный компьютер. С помощью некоторой операционной системы через интернет к нему подключаются ресурсы других компьютеров, число которых достигает полутора тысяч. В фиксированный момент несколько сотен компьютеров представляют свои вычислительные ресурсы. Это решает важнейшую проблему незагруженности компьютеров в данный момент времени (грубая оценка дает 90 % ресурсов).

Для параллельной реализации ЛЭА использовалась облачная платформа *Everest* [27], поддерживающая публикацию, выполнение и композицию вычислительных приложений в распределенной среде [28, 29]. Одна из отличительных черт платформы — возможность запуска приложений на произвольных комбинациях внешних вычислительных ресурсов, подключенных пользователями. Программирование ЛЭА для платформы *Everest* осуществлялось с помощью языка алгебраического программирования *AMPL* [30]. Поскольку *AMPL* не решает задачи непосредственно, в качестве соответствующего внешнего «решателя» использован пакет *SCIP* — широко применяемая система с открытым кодом [31]. При этом транслятор *AMPL* использован не только для формулировки основной задачи и всех вспомогательных подзадач, но и для управления сценарием расчетов в распределенной среде солверов, подключенных к *Everest*, с помощью подсистемы *AMPLX* [32], доступной в исходных кодах [33].

Сначала рассматривается задача булевого программирования с блочно-древовидной структурой (первый эксперимент). На верхнем уровне имеется блок 1, который содержит 3000 переменных. Он связан с блоками 2, 3 и 4, которые включают в себя 1996, 10 000 и 2000 переменных. Число связывающих переменных 2, 1 и 1. Блок 4 связан с блоком 8, который содержит 9996 переменных. Число связывающих переменных равно 4. Блок 2 связан с блоками 5 и 6, имеющих 7000 и 2996 переменных соответственно. Число связывающих переменных 3 и 1. Блок 5 связан с блоком 9, который состоит из 4996 переменных. Число связывающих переменных равно 4. Блок 3 связан с блоком 7, который имеет 4996 переменных. Число связывающих переменных составляет 4. Блок 7 связан с блоками 10 и 11, имеющих 1000 и 1996 переменных. Число связываю-

щих переменных 1 и 3. Блок 10 связан с блоками 12, 13 и 14, которые включают в себя 1000, 2000 и 996 переменных. Число связывающих переменных 1, 1 и 2. Блок 11 связан с блоком 15, который имеет 2996 переменных. Число связывающих переменных равно 4. Исходная задача с блочно-древовидной структурой содержала 50 000 переменных и 100 ограничений и решалась в течение 17 мин.

Во втором эксперименте решались задачи с блочно-лестничной структурой. Они содержали 100 000 переменных и 100 ограничений и решались параллельно в среднем за 6 ч. Задачи разбивались на пять подзадач с глубиной дерева. Каждая подзадача содержала 20 000 переменных и 20 ограничений, размер сепаратора равен 4.

Предложенные задачи с блочно-древовидной и блочно-лестничной структурой не могут быть решены точно только с помощью «решателя» *SCIP* без блочно-элиминационного алгоритма. Основным результатом этого эксперимента — реализация ЭЛЭА для решения задач динамической оптимизации со специальной структурой с помощью ГРИД-системы.

Заключение. Рассмотрен интересный класс задач целочисленного программирования с квазиблочной структурой. Он выделяется после применения критерия Щербина для задач с разреженными матрицами. В зависимости от размерности задач эффективны те или иные методы декомпозиции, которые и представлены в настоящей работе.

ЛИТЕРАТУРА

1. Щербина О.А. Локальные алгоритмы для блочно-древовидных задач дискретного программирования // Журн. вычисл. математики и матем. физики. 1985. № 8 (25). С. 1143–1154.
2. Лемтюжникова Д.В., Свириденко А.В., Щербина О.А. Алгоритм выделения блочно-древовидной структуры в разреженных задачах дискретной оптимизации // Таврический вестник информатики и математики. 2012. № 1. С. 44–55.
3. Авербах И.Л., Цурков В.И. Оптимизация в блочных задачах с целочисленными переменными. М.: Наука, 1995. 225 с.
4. Lubin M., Martin K., Petra C.G., Sandikçi B. On parallelizing dual decomposition in stochastic integer programming // Operations Research Letters. 2013. Vol. 41. Iss. 3. P. 252–258. DOI: 10.1016/j.orl.2013.02.003
5. Laesanklang W., Landa-Silva D., Castillo-Salazar J.A. Mixed integer programming with decomposition to solve a workforce scheduling and routing problem // Int. Conf. on Operations Research and Enterprise Systems (ICORES 2015). 2015. P. 283–293.
6. Heinz S., Beck J.C. Reconsidering mixed integer programming and MIP-based hybrids for scheduling // Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2012). 2012. P. 211–227.
7. Cire A., Coban E., Hooker J.N. Mixed integer programming vs. logic-based benders decomposition for planning and scheduling // Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2013). 2013. P. 325–331.

8. Park J., Karumanchi S., Iagnemma K. Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming // IEEE Transactions on Robotics. 2015. Vol. 31. No. 5. P. 1101–1115.
9. Gade D., Küçükyavuz S., Sen S. Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs // Mathematical Programming. 2014. Vol. 144. Iss. 1-2. P. 39–64. DOI: 10.1007/s10107-012-0615-y
10. Ntairo L. Fenchel decomposition for stochastic mixed-integer programming // Journal of Global Optimization January. 2013. Vol. 55. Iss. 1. P. 141–163. DOI: 10.1007/s10898-011-9817-8
11. Chu Y., You F. Integration of production scheduling and dynamic optimization for multi-product CSTRs: Generalized Benders decomposition coupled with global mixed-integer fractional programming // Computers & Chemical Engineering. 2013. Vol. 58. P. 315–333. DOI: 10.1016/j.compchemeng.2013.08.003
12. An effective problem decomposition method for scheduling of diffusion processes based on mixed integer linear programming / C. Jung, D. Pabst, M. Ham, M. Stehli, M. Rothe // IEEE Transactions on Semiconductor Manufacturing. 2014. Vol. 27. No. 3. P. 357–363.
13. Luedtke J. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support // Mathematical Programming. 2014. Vol. 146. Iss. 1-2. P. 219–244. DOI: 10.1007/s10107-013-0684-6
14. Integration of progressive hedging and dual decomposition in stochastic integer programs / G. Guo, G. Hackebeil, S.M. Ryan, J.P. Watson, D.L. Woodruff // Operations Research Letters. 2015. Vol. 43. Iss. 3. P. 311–316. DOI: 10.1016/j.orl.2015.03.008
15. Mitra S., Garcia-Herreros P., Grossmann I.E. A novel cross-decomposition multi-cut scheme for two-stage stochastic programming // Computer Aided Chemical Engineering. 2014. Vol. 33. P. 241–246. DOI: 10.1016/B978-0-444-63456-6.50041-7
16. Миронов А.А., Федорчук В.В., Цурков В.И. Минимум в моделях транспортного типа с интегральными ограничениями. I // Известия РАН. Теория и системы управления. 2003. № 4. С. 69–81.
17. Миронов А.А., Федорчук В.В., Цурков В.И. Минимум в моделях транспортного типа с интегральными ограничениями. II // Известия РАН. Теория и системы управления. 2005. № 5. С. 66–86.
18. Миронов А.А., Цурков В.И. Наследственно минимумные матрицы в моделях транспортного типа // Известия РАН. Теория и системы управления. 1998. № 6. С. 104–121.
19. Миронов А.А., Цурков В.И. Транспортные и сетевые задачи с минимумным критерием // Журн. вычисл. математики и матем. физики. 1995. Т. 35. № 1. С. 24–45.
20. Соколов А.А., Тизик А.П., Цурков В.И. Итеративный метод для транспортной задачи с дополнительными пунктами производства и потребления и квадратичным штрафом // Известия РАН. Теория и системы управления. 2013. № 4. С. 88–98.
21. Mironov A.A., Tsurkov V.I. Network models with fixed parameters at the communication nodes // Journal of Computer and Systems Sciences International. 1995. Vol. 33. No. 3. P. 107.
22. *Symphony* // Coin-or: веб-сайт. URL: <https://projects.coin-or.org/SYMPHONY> (дата обращения: 27.03.2017).
23. Щербина О.А. Локальные элиминационные алгоритмы для разреженных задач дискретной оптимизации. Дис. ... д-ра физ.-мат. наук. М., 2011. 361 с.
24. Gorry G.A., Shapiro J.F., Wolsey L.A. Relaxation methods for pure and mixed integer programming problems // Management Science. 1972. Vol. 18. No. 5. P. 229–239.

25. *Local elimination solver* // GitHub.com: веб-сайт. URL: <https://github.com/robionica/les> (дата обращения: 27.03.2017).
26. *Концепция* многозадачной ГРИД-системы с гибким распределением свободных вычислительных ресурсов суперкомпьютеров / А.П. Афанасьев, И.В. Бычков, О.С. Заикин и др. // Известия РАН. Теория и системы управления. 2017. № 4. С. 133–139.
27. *Everest: A cloud platform*. URL: <http://everest.distcomp.org> (дата обращения: 27.03.2017).
28. *Koutsopoulos I., Papaioannou T.G., Hatzi V.* Modeling and optimization of the smart grid eco-system // Foundations and Trends in Networking. 2016. Vol. 10. No. 2–3. P. 115–316. DOI: 10.1561/1300000042 URL: <http://www.nowpublishers.com/article/Details/NET-042>
29. *Salah A., Hart E.* A modified grid diversity operator for discrete optimization and its application to wind farm layout optimization problems // Companion Material Proceedings Genetic and Evolutionary Computation Conference. 2016. P. 977–982.
30. *A mathematical programming language* // AMPL: веб-сайт. URL: <https://ampl.com> (дата обращения: 27.03.2017).
31. *Achterberg T.* SCIP: Solving constraint integer programs // Mathematical Programming Computation. 2009. Vol. 1. Iss. 1. P. 1–41. DOI: 10.1007/s12532-008-0001-1
32. *Волошинов В.В., Смирнов С.А., Неверов В.С.* Интеграция программных средств оптимизационного моделирования в неоднородной вычислительной среде на основе системы AMPLX. URL: <http://distcomp.ru/~vladimirv/docs/nscf2015-amplx-VoloshinovVVSmirnovSANeverovVS.pdf> (дата обращения: 27.03.2017).
33. *AMPLX* // GitLab: веб-сайт. URL: <https://gitlab.com/ssmir/amplx> (дата обращения: 27.03.2017).

Ковков Дмитрий Валерьевич — канд. физ.-мат. наук, научный сотрудник Вычислительного центра им. А.А. Дородницына ФИЦ «Информатика и управление» РАН (Российская Федерация, 119333, Москва, ул. Вавилова, д. 40).

Лемтюжникова Дарья Владимировна — старший преподаватель Московского авиационного института (национального исследовательского университета) (Российская Федерация, 125993, Москва, Волоколамское шоссе, д. 4).

Просьба ссылаться на эту статью следующим образом:

Ковков Д.В., Лемтюжникова Д.В. Тестирование алгоритмов для целочисленных квази-блочных задач оптимизации // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. 2018. № 1. С. 59–75. DOI: 10.18698/0236-3933-2018-1-59-75

TESTING OF ALGORITHMS FOR INTEGER QUASIBLOCK OPTIMIZATION PROBLEMS

D.V. Kovkov¹

dmkovkov@gmail.com

D.V. Lemtyuzhnikova²

darabbt@gmail.com

¹ **Dorodnitsyn Computing Centre Federal Research Centre Computer Science and Control, Russian Academy of Sciences, Moscow, Russian Federation**

² **Moscow Aviation Institute (National Research University), Moscow, Russian Federation**

Abstract

The article focuses on the algorithms for solving integer quasi-block optimization problems. For this purpose we analyzed modern decomposition techniques, as well as the advantages of using the local elimination algorithm for large-scale problems. Moreover, we described the special issues in applying parametric optimization and carried out a series of computational experiments for solving large-scale integer linear programming problems by exact, approximate, and heuristic algorithms. The study gives the results obtained for different modifications of the local elimination algorithm. It also describes computational experiments with the local elimination algorithm parallelized by grid technologies. Finally, we provide some examples of the problems that cannot be solved without the paralleling approach

Keywords

Decomposition, integer programming, quasiblock structure, local elimination algorithm

Received 20.04.2017

© BMSTU, 2018

REFERENCES

- [1] Shcherbina O.A. Local algorithms for block-tree problems of discrete programming. *USSR Computational Mathematics and Mathematical Physics*, 1985, vol. 25, iss. 4, pp. 114–121. DOI: 10.1016/0041-5553(85)90154-5
- [2] Lemtyuzhnikova D.V., Sviridenko A.V., Shcherbina O.A. Discrimination algorithm for block-tree diagram in split tasks of discrete optimization. *Tavrisheskiy vestnik informatiki i matematiki*, 2012, no. 1, pp. 44–55 (in Russ.).
- [3] Averbakh I.L., Tsurkov V.I. Optimizatsiya v blochnykh zadachakh s tselochislennymi peremennymi [Optimization in block problems with integer variables]. Moscow, Nauka Publ., 1995. 225 p. (in Russ.).
- [4] Lubin M., Martin K., Petra C.G., Sandikçi B. On parallelizing dual decomposition in stochastic integer programming. *Operations Research Letters*, 2013, vol. 41, iss. 3, pp. 252–258. DOI: 10.1016/j.orl.2013.02.003
- [5] Laesanklang W., Landa-Silva D., Castillo-Salazar J.A. Mixed integer programming with decomposition to solve a workforce scheduling and routing problem. *Int. Conf. on Operations Research and Enterprise Systems (ICORES 2015)*, 2015, pp. 283–293.
- [6] Heinz S., Beck J.C. Reconsidering mixed integer programming and MIP-based hybrids for scheduling. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2012)*, 2012, pp. 211–227.
- [7] Cire A., Coban E., Hooker J.N. Mixed integer programming vs. logic-based benders decomposition for planning and scheduling. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR 2013)*, 2013, pp. 325–331.
- [8] Park J., Karumanchi S., Iagnemma K. Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming. *IEEE Transactions on Robotics*, 2015, vol. 31, no. 5, pp. 1101–1115.
- [9] Gade D., Küçükayavuz S., Sen S. Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Mathematical Programming*, 2014, vol. 144, iss. 1-2, pp. 39–64. DOI: 10.1007/s10107-012-0615-y

- [10] Ntaimo L. Fenchel decomposition for stochastic mixed-integer programming. *Journal of Global Optimization* January, 2013, vol. 55, iss. 1, pp. 141–163. DOI: 10.1007/s10898-011-9817-8
- [11] Chu Y., You F. Integration of production scheduling and dynamic optimization for multi-product CSTRs: Generalized Benders decomposition coupled with global mixed-integer fractional programming. *Computers & Chemical Engineering*, 2013, vol. 58, pp. 315–333. DOI: 10.1016/j.compchemeng.2013.08.003
- [12] Jung C., Pabst D., Ham M., Stehli M., Rothe M. An effective problem decomposition method for scheduling of diffusion processes based on mixed integer linear programming. *IEEE Transactions on Semiconductor Manufacturing*, 2014, vol. 27, no. 3, pp. 357–363.
- [13] Luedtke J. A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming*, 2014, vol. 146, iss. 1-2, pp. 219–244. DOI: 10.1007/s10107-013-0684-6
- [14] Guo G., Hackebeil G., Ryan S.M., Watson J.P., Woodruff D.L. Integration of progressive hedging and dual decomposition in stochastic integer programs. *Operations Research Letters*, 2015, vol. 43, iss. 3, pp. 311–316. DOI: 10.1016/j.orl.2015.03.008
- [15] Mitra S., Garcia-Herreros P., Grossmann I.E. A novel cross-decomposition multi-cut scheme for two-stage stochastic programming. *Computer Aided Chemical Engineering*, 2014, vol. 33, pp. 241–246. DOI: 10.1016/B978-0-444-63456-6.50041-7
- [16] Mironov A.A., Fedorchuk V.V., Tsurkov V.I. Minimax in transportation models with integral constraints: I. *Journal of Computer and Systems Sciences International*, 2003, vol. 42, no. 4, pp. 562–574.
- [17] Mironov A.A., Fedorchuk V.V., Tsurkov V.I. Minimax in transportation models with integral constraints: II. *Journal of Computer and Systems Sciences International*, 2005, vol. 44, no. 5, pp. 732–752.
- [18] Mironov A.A., Tsurkov V.I. Hereditary minimax matrixes in transportation models. *Izvestiya RAN. Teoriya i sistemy upravleniya*, 1998, no. 6, pp. 104–121 (in Russ.).
- [19] Mironov A.A., Tsurkov V.I. Transport and network problems with the minimax criterion. *Computational Mathematics and Mathematical Physics*, 1995, vol. 35, no. 1, pp. 15–30.
- [20] Sokolov A.A., Tizik A.P., Tsurkov V.I. Iterative method for the transportation problem with additional supply and consumption points and quadratic cost. *Journal of Computer and Systems Sciences International*, 2013, vol. 52, iss. 4, pp. 588–598. DOI: 10.1134/S106423071304014X
- [21] Mironov A.A., Tsurkov V.I. Network models with fixed parameters at the communication nodes. *Journal of Computer and Systems Sciences International*, 1995, vol. 33, no. 3, pp. 107.
- [22] Symphony. Coin-or: website. Available at: <https://projects.coin-or.org/SYMPHONY> (accessed: 27.03.2017).
- [23] Shcherbina O.A. Lokal'nye eliminatsionnye algoritmy dlya razrezhennykh zadach diskretnoy optimizatsii. Dis. d-ra fiz. mat. nauk [Local elimination algorithms for sparse problems of discrete optimization. Dr. phys.-mat. sci. diss.]. Moscow, 2011. 361 p. (in Russ.).
- [24] Gorry G.A., Shapiro J.F., Wolsey L.A. Relaxation methods for pure and mixed integer programming problems. *Management Science*, 1972, vol. 18, no. 5, pp. 229–239.
- [25] Local elimination solver. GitHub.com: website. Available at: <https://github.com/robionica/les> (accessed: 27.03.2017).
- [26] Afanasiev A.P., Bychkov I.V., Zaikin O.S., Manzyuk M.O., Posypkin M.A., Semenov A.A. Concept of a multitask grid system with a flexible allocation of idle computational resources

of supercomputers. *Journal of Computer and Systems Sciences International*, 2017, vol. 56, iss. 4, pp. 701–707. DOI: 10.1134/S1064230717040025

[27] Everest: A cloud platform. Available at: <http://everest.distcomp.org> (accessed: 27.03.2017).

[28] Koutsopoulos I., Papaioannou T.G., Hatzi V. Modeling and optimization of the smart grid ecosystem. *Foundations and Trends in Networking*, 2016, vol. 10, no. 2–3, pp. 115–316.

DOI: 10.1561/1300000042 Available at: <http://www.nowpublishers.com/article/Details/NET-042>

[29] Salah A., Hart E. A modified grid diversity operator for discrete optimization and its application to wind farm layout optimization problems. *Companion Material Proceedings Genetic and Evolutionary Computation Conference*, 2016, pp. 977–982.

[30] A mathematical programming language. AMPL: website. Available at: <https://ampl.com> (accessed: 27.03.2017).

[31] Achterberg T. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 2009, vol. 1, iss. 1, pp. 1–41. DOI: 10.1007/s12532-008-0001-1

[32] Voloshinov V.V., Smirnov S.A., Neverov V.S. Integratsiya programmnykh sredstv optimizatsionnogo modelirovaniya v neodnorodnoy vychislitel'noy srede na osnove sistemy AMPLX [Optimization simulation software integration in heterogeneous computing environment based on AMPLX system] (in Russ.).

Available at: <http://distcomp.ru/~vladimirv/docs/>

nscf2015-amplx-VoloshinovVV-SmirnovSA-NeverovVS.pdf (accessed: 27.03.2017).

[33] AMPLX. GitLab: website. Available at: <https://gitlab.com/ssmir/amplx> (accessed: 27.03.2017).

Kovkov D.V. — Cand. Sc. (Phys.-Math.), Research Fellow, Dorodnitsyn Computing Centre Federal Research Centre Computer Science and Control, Russian Academy of Sciences (Vavilova ul. 40, Moscow, 119333 Russian Federation).

Lemtyuzhnikova D.V. — Assist. Professor, Moscow Aviation Institute (National Research University) (Volokolamskoe shosse 4, Moscow, 125993 Russian Federation).

Please cite this article in English as:

Kovkov D.V., Lemtyuzhnikova D.V. Testing of Algorithms for Integer Quasiblock Optimization Problems. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Priborostr.* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2018, no. 1, pp. 59–75 (in Russ.).

DOI: 10.18698/0236-3933-2018-1-59-75