ISSN 2782-2001
Системы анализа
и обработки данных
том 83, № 3, 2021, с. 99–114

http://journals.nstu.ru/vestnik
Analysis and data
processing systems
Vol. 83, No. 3, 2021, pp. 99–114

ИНФОРМАТИКА,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И УПРАВЛЕНИЕ

INFORMATICS,
COMPPUTER ENGINEERING
AND CONTROL

УДК 004.832, 622.2 DOI: 10.17212/2782-2001-2021-3-99-114

Интеллектуальный поиск точных решений задачи планирования открытых горных работ*

А.А. ЗУЕНКО[a], Р.А. МАКЕДОНОВ[b], Ю.А. ОЛЕЙНИК[c]

184209, РФ, г. Апатиты Мурманской обл., ул. Ферсмана, 24А, Институт информатики и математического моделирования – обособленное подразделение Федерального исследовательского центра «Кольский научный центр Российской академии наук» (ИИММ КНЦ РАН)

[a] *zuenko@iimm.ru* [b] *makedonov@iimm.ru* [c] *yoleynik@iimm.ru*

Разработан метод интеллектуального поиска точных решений задачи планирования открытых горных работ. Метод реализован в рамках парадигмы программирования в ограничениях, что позволяет совместно обрабатывать разнородные количественные и качественные ограничения (в частности, экономические, технологические и др.), а также поддерживать развивающуюся модель предметной области, открытую для добавления новых или удаления имеющихся ограничений. В модель можно добавлять различные ограничения, включая те, для которых сложно найти подходящее аналитическое выражение. В отличие от существующих методов локального поиска, предложенный метод позволяет исследовать пространство поиска. Он позволяет находить глобальный оптимум в пространствах большой размерности, описывающих практически значимые задачи, которые возникают на производстве. В настоящее время для решения поставленной задачи широко применяются методы целочисленного линейного программирования, принципиальным недостатком которых является необходимость представления всех ограничений в форме линейных уравнений и неравенств. На практике же некоторые задачи комбинаторной оптимизации невозможно линеаризовать и решить с помощью традиционных методов математического программирования. Разработанный метод проиллюстрирован на примере трехмерной задачи поиска положений рабочего борта карьера по периодам отработки с учетом заданной производительности по полезному ископаемому и вскрышным породам и целевой функции максимизации прибыли с учетом дисконтирования. Выделены типы ограничений, необходимые для моделирования рассматриваемой задачи. Для них рассмотрена возможность применения существующих процедур вывода на ограничениях. Предложенный метод позволяет получать точные решения за счет интеллектуализации процесса решения путем использования высокоэффективных алгоритмов редукции пространства поиска

* *Статья получена 01 марта 2021 г.*

Исследование выполнено при финансовой поддержке РФФИ в рамках научных проектов № 20-07-00708-а, 19-07-00359-а.

для каждого из типов ограничений и специализированных эвристик отсечения неперспективных альтернатив в дереве поиска.

Ключевые слова: программирование в ограничениях, задача удовлетворения ограничений, распространение ограничений, вывод на ограничениях, интеллектуальное планирование, открытые горные работы, целочисленное линейное программирование, развивающаяся модель предметной области

ВВЕДЕНИЕ

Открытые горные работы – это способ добычи полезных ископаемых с поверхности земли с помощью горных выработок, находящихся под открытым небом. При создании модели рудное тело обычно дискретизируется и моделируется как набор блоков. Существуют различные варианты формулировки данной задачи. За последние 30 лет были предложены различные подходы к ее решению: классическое линейное и математическое программирование [1–6], а также метаэвристическое программирование [7–9].

Наиболее развитым подходом к решению задачи планирования открытых горных работ следует признать подход, основанный на применении методов целочисленного линейного программирования [1, 4]. К его достоинствам можно отнести то, что он позволяет находить глобальный оптимум для задачи планирования открытых горных работ. При использовании же метаэвристического программирования зачастую приходится довольствоваться лишь нахождением локального оптимума. Принципиальным недостатком методов целочисленного линейного программирования является то, что все ограничения должны быть представлены в форме линейных уравнений и неравенств. К сожалению, при решении практических задач этот недостаток является непреодолимым препятствием для применения рассматриваемого подхода. Технология разработки каждого конкретного карьера определяется тем, какое полезное ископаемое будет извлечено, а также экономическими, технологическими и другими ограничениями, определяемыми заказчиком при проведении открытых горных работ. При этом требования заказчика могут уточняться, т. е. модель предметной области постоянно развивается и должна быть открыта для добавления новых или удаления имеющихся ограничений. В этих условиях применение классических методов теории исследования операций представляется затруднительным.

В ходе исследований разработан новый метод интеллектуального поиска точных решений задачи планирования открытых горных работ, который далее проиллюстрирован на примере задачи поиска положений рабочего борта карьера по периодам отработки с учетом заданной производительности по полезному ископаемому (ПИ) и вскрышным породам (ВП). В настоящих исследованиях для точного решения задачи планирования открытых горных работ предлагается применять технологию программирования в ограничениях (Constraint Programming) [10, 12]. Идея возможности применения технологии программирования в ограничениях для решения рассматриваемого класса задач была впервые выдвинута в [13], однако, по мнению авторов, не получила должного развития. Благодаря архитектуре библиотек программирования в ограничениях появляется возможность совместно обрабатывать разнородные ограничения и по мере необходимости вводить новые типы ограничений. При этом не меняется общий алгоритм поиска решения задачи: для новых типов ограничений требуется только описать специализированные методы-распространители, осуществляющие вывод на ограничениях данного типа.

Далее приведем постановку задачи на содержательном уровне.

1. СОДЕРЖАТЕЛЬНАЯ ПОСТАНОВКА ЗАДАЧИ

Начальное (на момент начала планирования) и конечное положение борта карьера ограничивают пространство геологической среды. В качестве исходных данных задачи выступает модель геологической среды, которая представляет собой набор блоков одинакового размера (рис. 1). Для каждого блока определены его размеры, положение в пространстве, процент содержания в нем ПИ и ценность блока – разницу между стоимостью работ по извлечению ПИ из блока (выемка блока из карьера и дальнейшая переработка блока) и выгодой от реализации извлеченного ПИ.

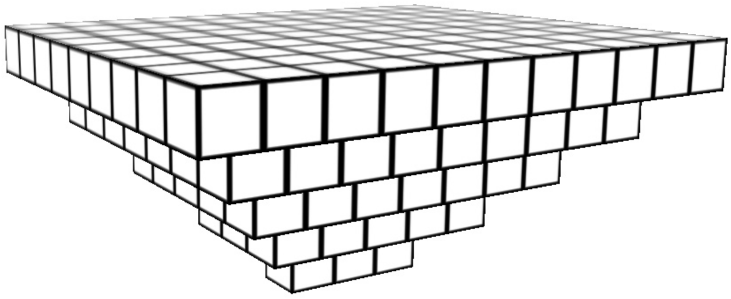

Рис. 1. Блочная модель карьера

Fig. 1. An open-pit block model

Содержательная постановка задачи календарного планирования состоит в том, чтобы определить такие положения рабочих бортов по периодам отработки, чтобы объемы ПИ и ВП, заключенные между последовательными положениями, соответствовали заданным с точностью до допустимой погрешности, при этом прибыль (суммарная ценность блоков) от разработки карьера должна быть максимальной. Развитие карьера – его углубление и расширение – происходит с соблюдением технических ограничений (ширина рабочей площадки) и технологических (какие блоки должны быть извлечены перед извлечением заданного блока). Исходная ценность каждого блока меняется в зависимости от года его добычи. Далее условимся называть конечное положение борта карьера конечным карьером, а текущую границу разработки – рабочим карьером или просто карьером.

Структура исходных данных может быть представлена в виде таблицы в *Excel.* Каждая строка данной таблицы описывает один блок карьера и содержит информацию о координатах блока, размерах блока, ценности блока и процентном содержании ПИ в блоке. Объем блока вычисляется из его размеров. Этот объем, в свою очередь, разделяется на объем ВП и объем ПИ в соответствии с заданным для блока содержанием ПИ.

Прежде чем перейти к описанию предлагаемого метода планирования, кратко рассмотрим особенности технологии программирования в ограничениях.

2. ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ В ОГРАНИЧЕНИЯХ

На настоящий момент технология программирования в ограничениях является мощным инструментом решения задач комбинаторного поиска и комбинаторной оптимизации. Для применения данной технологии любая задача должна быть сформулирована как задача удовлетворения ограничений (CSP – Constraint Satisfaction Problem), состоящая из трех компонент – $<X, D, C>$ [14]:

X – множество переменных $\{X_1, X_2, \ldots, X_n\}$;

D – множество доменов переменных $\{D_1, D_2, \ldots, D_n\}$, где D_i является доменом (областью определения) переменной X_i;

C – множество ограничений $\{C_1, C_2, \ldots, C_m\}$, которые предписывают допустимые комбинации значений переменных.

Каждый домен D_i описывает множество допустимых значений $\{v_1, \ldots, v_k\}$ для переменной X_i. Под ограничением понимается любое соотношение между переменными предметной области. В качестве ограничений могут выступать арифметические выражения, логические формулы, таблицы, а также выражения, формулируемые на языке специализированных теорий. С точки зрения конечного пользователя задача CSP формулируется в декларативном виде на языке, близком к языку математики.

Решением задачи CSP является полное присваивание, которое удовлетворяет всем ограничениям. В некоторых случаях необходимо получить все решения. Иногда требуется найти такое решение, в котором значения переменных оптимизировали бы некоторый заданный функционал.

Технология программирования в ограничениях предоставляет мощные и гибкие методы решения задач комбинаторного поиска. Алгоритмы удовлетворения ограничений содержат две обязательные компоненты: а) компоненту, реализующую вывод на ограничениях (распространение ограничений, фильтрация ограничений); б) компоненту, реализующую стратегию поиска. Обе эти компоненты нацелены на интеллектуализацию процесса поиска, т. е. на то, чтобы избежать полного перебора альтернатив путем быстрого обнаружения заведомо неперспективных комбинаций значений переменных. Термин «интеллектуальный поиск» является антонимом термина «слепой поиск» [12].

Сначала рассмотрим некоторые стратегии интеллектуального поиска. В рамках программирования в ограничениях традиционная стратегия поиска состоит в совместном применении методов распространения ограничений с методами поиска в глубину с возвратами, при этом применяются специализированные эвристики для выбора переменной и ее значения, а также используются разумные стратегии возврата к состоянию, являющемуся причиной возникновения недопустимого присваивания, такие как бэкмаркинг (backmarking), бэкджампинг (backjumping), динамический бэктрекинг и др. [12]. Особенность методов поиска с возвратами заключается в том, что частичное допустимое решение пошагово расширяется до полного. Если частичное решение окажется недопустимым, то осуществляется возврат и расширяется в альтернативном направлении предыдущее частичное решение. Методы поиска в глубину с возвратами называются также методами систематического (конструктивного) поиска, поскольку они гарантируют нахождение оптимума целевой функции задачи CSP в случае его существования и выдают негативный ответ в случае отсутствия решения задачи.

Также для организации эффективного поиска часто используются методы локального поиска. Идея локального поиска заключается в генерировании начального кандидата решения задачи CSP (полного присваивания), пошаговом его улучшении, например, путем уменьшения количества неудовлетворенных ограничений. Алгоритмы локального поиска отличаются методами нахождения этого улучшения и способами преодоления «ловушек» локальных оптимумов. Главным недостатком алгоритмов локального поиска в контексте решения рассматриваемых задач оптимизации является их неполнота, т. е. поиск может остановиться в локальном минимуме, который в действительности не является глобальным решением. К методам локального поиска относят поиск с восхождением к вершине, имитацию отжига (simulated annealing), поиск с запретами, роевой интеллект, генетические алгоритмы и т. д.

Теперь перейдем к рассмотрению методов вывода на ограничениях (распространения ограничений). Вывод на ограничениях реализуется как целенаправленное сужение областей определения одних переменных на основе усеченных на данном шаге вывода доменов других переменных. Для каждого типа ограничений разрабатываются собственные методы распространения. Например, для ограничения $alldiff(x_1,\ldots, x_n)$, предписывающего, что все переменные $x_1,\ldots, x_n$ должны принимать различные значения, соответствующий метод распространения опирается на теорию графов, а именно на алгоритм обнаружения паросочетаний в графе [15]. Чтобы процедуры вывода на ограничениях имели низкую вычислительную сложность, для их разработки применяются развитые научные теории (например, теория исследования операций, теория графов и т. п.). При использовании стратегий локального поиска процесс распространения ограничения, по сути, сводится к подстановке всех переменных конкретных значений и проверке того, удовлетворено ли данное ограничение.

Особую важность эффективность процедур вывода на ограничениях приобретает при совместном применении с методами систематического поиска, поскольку позволяет отсекать несовместные значения переменных, редуцируя пространство поиска. В реальных задачах количество ограничений очень велико и любое самое незначительное изменение домена на этапе распространения в каком-либо ограничении запускает целый каскад вызовов алгоритмов распространения для других ограничений. Причем прослеживается закономерность: чем проще ограничение, тем меньше возможностей у алгоритма распространения для сокращения доменов участвующих в нем переменных. Таким образом, набор однотипных ограничений целесообразно объединять в одно ограничение, разрабатывая для данного набора эффективные алгоритмы распространения. Такие составные ограничения называют глобальными. В [11] глобальное ограничение трактуется как ограничение, описывающее отношение, арность которого заранее не ограничена. Ранее уже упоминалось глобальное ограничение *alldiff*, которое в зависимости от того, какие конкретно переменные оно затрагивает, может иметь различную арность.

В рамках рассматриваемой в работе задачи планирования открытых горных работ используются глобальные ограничения двух основных типов: ограничение выбора максимального значения (*max constraint*) и ограничение, описывающее задачу об упаковке в контейнеры (*bin-packing constraint*).

Ограничение max [16]. Это ограничение предписывает, что значение одной указанной переменной из набора должно быть не меньше значений всех остальных переменных рассматриваемого набора. В рамках рассматриваемой задачи оно используется для описания технологических ограничений на порядок выемки блоков.

Пример 1. Пусть $X_1 \in \{2,\ 3\}$, $X_2 \in \{2,\ 3\}$, $X_3 \in \{1,\ 2\}$, $X_4 \in \{4,\ 5\}$, $X_5 \in \{2,\ 3,\ 4\}$, тогда некоторыми решениями, удовлетворяющими ограничению $\max(X_5,\ \{X_1,\ X_2,\ X_3,\ X_4,\ X_5\})$ будут следующие:

1) $X_1 = 2,\ X_2 = 2,\ X_3 = 1,\ X_4 = 4,\ X_5 = 4;$

2) $X_1 = 2,\ X_2 = 2,\ X_3 = 2,\ X_4 = 4,\ X_5 = 4;$

3) $X_1 = 2,\ X_2 = 3,\ X_3 = 2,\ X_4 = 4,\ X_5 = 4.$

Ограничение bin-packing [17]. Задача *bin-packing* описывается следующим образом: существует *n* предметов и *m* контейнеров, каждый предмет имеет свой объем и у каждого контейнера есть своя вместимость. Цель состоит в распределении всех предметов по контейнерам таким образом, чтобы вместимость ни одного из контейнеров не была превышена. Это ограничение описывается с помощью следующих массивов:

• массива переменных $X_1,\ldots,X_n$, каждая из которых имеет домен [1, …, *m*] и показывает, что предмет под номером *i* будет помещен в контейнер под номером X_i;

• массива объемов $vol_1,\ldots,vol_n$, определяющего, что предмет под номером *i* имеет объем vol_i;

• массива вместимостей $cap_1,\ldots,cap_m$, определяющего, что контейнер под номером *j* имеет вместимость cap_j.

Пример 2. Пусть $X_1 \in \{1,\ 2\}$, $X_2 \in \{1,\ 2\}$, $X_3 \in \{1,\ 2\}$, $vol = \{4,\ 3,\ 1\}$, $cap = \{4,\ 5\}$, тогда всеми решениями, удовлетворяющими ограничению $bin\_packing(\{X_1,\ X_2,\ X_3,\ X_4\},\ vol,\ cap,)$ будут:

1) $X_1 = 1,\ X_2 = 2,\ X_3 = 2;$

2) $X_1 = 2,\ X_2 = 1,\ X_3 = 1;$

3) $X_1 = 2,\ X_2 = 1,\ X_3 = 2.$

Для программирования в ограничениях используются как отдельные среды программирования с поддержкой специализированных языков ограничений (например, MiniZinc, IBM ILOG CPLEX, ECLiPSe Constraint Programming System), так и подключаемые библиотеки на обычных языках программирования (например, Google Optimization Tools (C++, Python, .Net, Java), Choco (Java), Gecode (C++)). В настоящей работе выбор в пользу подключаемых библиотек, а именно Choco [18], обусловлен тем, что в специализированных средах программирования в ограничениях затруднена разработка методов-распространителей для пользовательских ограничений, отличных от встроенных. Кроме того, разработанные с использованием подключаемых библиотек методы планирования могут быть легко интегрированы в специализированные программные комплексы для автоматизации горных производственных работ.

Наконец, приведем некоторые достоинства технологии программирования в ограничениях, которые играют важную роль в настоящей работе.

- Благодаря архитектуре библиотек программирования в ограничениях появляется возможность совместно обрабатывать количественные и качественные ограничения.
- Обеспечивается возможность сопровождать модели, открытые для оперативных модификаций, т. е. развивающиеся модели. При добавлении в модель новых типов ограничений зачастую требуется лишь разработать соответствующий метод-распространитель, а не придумывать принципиально иную схему поиска.
- Библиотеки программирования в ограничениях содержат высокоэффективные алгоритмы вывода для обработки так называемых глобальных ограничений. Применение подобных ограничений позволяет конструировать решаемую задачу.

Ниже подробно описаны типы ограничений, используемые для формализации задачи планирования открытых горных работ, а также методы вывода на данных ограничениях и разработанные эвристики для организации интеллектуального поиска.

3. ФОРМАЛИЗАЦИЯ ЗАДАЧИ В ПАРАДИГМЕ ПРОГРАММИРОВАНИЯ В ОГРАНИЧЕНИЯХ

В парадигме программирования в ограничениях задача планирования открытых горных работ ставится следующим образом: каждому блоку карьера сопоставляется целочисленная переменная $X_{i,j,k}$, которая может принимать значения от 1 до N, где N – номер последнего возможного периода разработки карьера. Присваивание $X_{i,j,k} = a$, где $a \in [1, N]$, означает, что блок с координатами (i, j, k) извлекается в период a.

Базовыми для рассматриваемой постановки задачи являются следующие ограничения: а) ограничения на порядок извлечения блоков; б) ограничения на заданную производительность по полезному ископаемому и вскрышным породам. В качестве целевой функции рассматривается максимизация прибыли от разработки карьера. Однако в рамках предложенного метода нет принципиальных запретов на то, чтобы обрабатывать ограничения и других типов.

Решение задачи состоит в том, чтобы назначить всем переменным $X_{i,j,k}$ целочисленные значения таким образом, чтобы одновременно выполнялись все перечисленные выше ограничения и функция прибыли принимала максимальное значение. Разберем семантику используемых ограничений более подробно.

Ограничения на порядок извлечения блоков (отношения предшествования-следования) обеспечивают последовательную разработку карьера, исключая технологически невозможные варианты выемки блоков. Правила извлечения блоков задаются в виде некоторого шаблона (схемы) выемки блоков и описывают принципы заглубления и расширения карьера. В качестве примера рассмотрим следующий шаблон (рис. 2): для выемки блока $X_{i,j,k}$, где i, j, k – координаты блока, необходимо выкопать блоки $X_{i,j,k+1}$, $X_{i-1,j,k+1}$, $X_{i+1,j,k+1}$, $X_{i,j-1,k+1}$, $X_{i,j+1,k+1}$, другими словами, необходимо извлечь пять вышележащих блоков при их наличии.

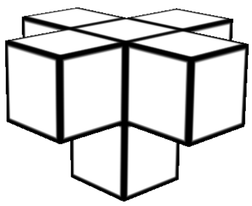

Рис. 2. Пример шаблона (схемы) извлечения блоков

Fig. 2. An example of a block mining template (scheme)

Этот шаблон можно задать с помощью пяти неравенств:

$$X_{i,j,k} \ge X_{i,j,k+1},\ X_{i,j,k} \ge X_{i-1,j,k+1},\ X_{i,j,k} \ge X_{i+1,j,k+1},$$

$$X_{i,j,k} \ge X_{i,j-1,k+1},\ X_{i,j,k} \ge X_{i,j+1,k+1}.$$

Теперь рассмотрим альтернативный способ формализации этого шаблона, предполагающий использование глобального ограничения *max*.

Все пять перечисленных выше неравенств можно описать с помощью одного глобального ограничения *max*:

$$\max(X_{i,j,k}, \{X_{i,j,k}, X_{i,j,k+1}, X_{i-1,j,k+1}, X_{i+1,j,k+1}, X_{i,j-1,k+1}, X_{i,j+1,k+1}\}).$$

В настоящей работе предлагается с помощью ограничения *max* описывать связи рассматриваемого блока не только с вышележащими непосредственными соседями, а со всеми блоками, которые необходимо извлечь для извлечения заданного блока. Таким образом, с увеличением глубины залегания блока ограничение *max* будет охватывать всё больше переменных, позволяя эффективнее сокращать их домены в процессе решения.

Ограничения на заданную производительность по полезному ископаемому и вскрышным породам моделируются линейными неравенствами. Пусть $o_{i,j,k}$ – это константа, показывающая содержание руды в блоке с индексами i, j, k; $w_{i,j,k}$ – это константа, показывающая содержание вскрышных пород в блоке с индексами i, j, k. Введем следующие обозначения: O – заданная производительность по полезному ископаемому; ΔO – допустимая погрешность O; W – заданная производительность по вскрышным породам; ΔW – допустимая погрешность W. Тогда ограничения на заданную производительность по полезному ископаемому и вскрышным породам можно выразить в виде неравенств:

$$O - \Delta O \le \sum_{i,j,k} o_{ijk} (X_{ijk} = y) \le O + \Delta O \text{ для каждого } y \in [1 \ldots N],$$

$$W - \Delta W \le \sum_{i,j,k} w_{ijk} (X_{ijk} = y) \le W + \Delta W \text{ для каждого } y \in [1 \ldots N].$$

Эти ограничения могут быть сформулированы в виде задачи об упаковке в N контейнеров (bin-packing) вместимостью от $O - \Delta O$ до $O + \Delta O$ и от $W - \Delta W$ до $W + \Delta W$ соответственно. Далее кратко рассмотрим, каким образом ограничение на то, что за каждый период ведения работ объем добытого ПИ должен принадлежать заданному интервалу $[O - \Delta O;\ O + \Delta O]$, сводится к ограничению *bin-packing*. Массив $X_1, \ldots, X_n$, являющийся одним из параметров глобального ограничения *bin-packing*, должен состоять из переменных, каждая из которых соотносится с блоком карьера, содержащим ПИ (n – число таких блоков). Домен этих переменных будет $[1, \ldots, m]$, где m – номер последнего периода ведения работ. Период ведения работ можно интерпретировать как «контейнер», тогда массив $vol_1, \ldots, vol_n$ будет содержать информацию об объеме ПИ в каждом из соответствующих блоков, массив $cap_1, \ldots, cap_m$ будет содержать переменные с доменами $[O - \Delta O;\ O + \Delta O]$.

Рассмотрим *целевую функцию* (*функцию максимизации прибыли от разработки карьера*). Для описания зависимости ценности блока от периода ведения работ зададим функцию $Fr(X_{i,j,k})$ вида

$$Fr(X_{i,j,k}) = V_{i,j,k} (0.9)^{X_{i,j,k} - 1},$$

где $V_{i,j,k}$ – начальная ценность блока с индексами i, j, k. Тогда общая оптимизационная функция будет иметь вид $\sum_{i,j,k} Fr(X_{i,j,k}) \to \max$. Ограничение определяет, что каждое новое найденное решение должно давать значение оптимизационной функции бóльшее, чем предыдущие решения.

4. СТРАТЕГИЯ ПОИСКА РЕШЕНИЙ ЗАДАЧИ CSP

На этапе формирования (генерации) ограничений для каждого блока осуществляется построение множества всех вышележащих блоков, а не только его непосредственных соседей, которые, согласно ограничению на порядок выемки блоков, необходимо извлечь перед тем, как приступать к разработке рассматриваемого блока. Данное множество используется для построения глобального ограничения *max*, а также для оценки минимальных объемов ВП и ПИ, которые необходимо извлечь для выемки рассматриваемого блока. Сравнивая эти объемы с верхними границами заданных диапазонов норм выработки ВП и ПИ за один период разработки, можно прийти к выводу, что некоторые блоки невозможно извлечь раньше определенного периода. Исходя из этой информации производится предварительное «прореживание» доменов переменных, соответствующих блокам модели.

Рассмотрим небольшой пример. Пусть для некоторого блока модели доменом сопоставленной ему переменной X является множество $\{1, \ldots, 5\}$. Допустим, что объем ПИ в блоках, которые располагаются выше рассматриваемого, равен 10 000, а верхняя граница нормы выработки ПИ за один период равна 3000. Очевидно, что при заданной норме рассматриваемый блок будет извлечен не ранее четвертого периода разработки, поэтому домен переменной X следует сократить до множества $\{4, 5\}$.

В результате такого «прореживания» блокам на нижних уровнях будут соответствовать переменные с доменами меньшей мощности, чем блокам на уровнях, расположенных выше.

В качестве стратегии поиска предлагается использовать поиск в глубину с возвратами. Для выбора переменной в процессе поиска используется следующая эвристика: в первую очередь выбираются переменные с наименьшими мощностями доменов; при наличии нескольких переменных с доменами одинаковой мощности выбирается переменная, которой соответствует самый глубоко залегающий блок. Принимая во внимание вид целевой функции, можно заключить, что рудные блоки (блоки с положительной ценностью) выгодно извлекать как можно раньше, а вскрышные (блоки с отрицательной ценностью) – как можно позже. В силу приведенных соображений предлагается следующая эвристика для выбора значения переменной: для переменной вскрышного блока в первую очередь выбирается наибольшее значение в домене, а у рудного блока – наименьшее.

Данные эвристики позволяют эффективно сокращать домены переменных в процессе поиска, а также быстрее максимизировать значение целевой функции.

5. АНАЛИЗ НЕСТАНДАРТНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ОГРАНИЧЕНИЙ

В рамках предложенного подхода можно добавлять в задачу самые различные ограничения, и даже те, для которых сложно найти подходящее аналитическое выражение (формулу). В качестве примера рассмотрим ограничение, смысл которого заключается в том, что в карьере не допускается наличие сразу нескольких локальных заглублений (допускается только одно заглубление). Для простоты возьмем плоский (двумерный) карьер.

На рис. 3 приведен пример конфигурации карьера с несколькими локальными заглублениями.

Все блоки карьера можно отнести к тому или иному горизонтальному слою. Каждый горизонтальный слой может быть представлен в виде одномерного массива переменных задачи CSP. При решении задачи CSP каждой переменной приписывается временная метка, показывающая, в какой период разработки карьера была осуществлена выемка соответствующего блока. При этом расстановка меток в пределах каждого такого одномерного массива (горизонтального слоя блоков) согласно рассматриваемому ограничению должна удовлетворять следующим требованиям: все значения одномерного массива, стоящие перед минимальным значением, должны образовывать монотонно невозрастающую последовательность, а после минимального – монотонно неубывающую последовательность. При этом допускается ситуация, когда минимальное значение стоит в начале и/или в конце массива. На рис. 4 приведены примеры правильной (слева) и неправильной (справа) расстановки временных меток для горизонтального слоя карьера. Расстановка меток справа на рис. 4 соответствует ситуации, когда на некотором промежуточном этапе разработки карьера имеется два локальных заглубления.

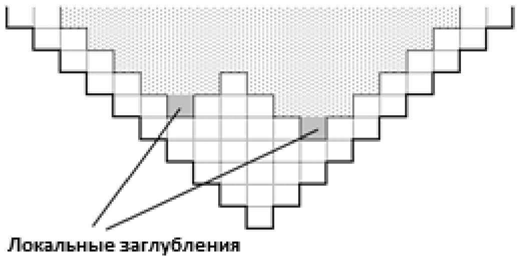

Рис. 3. Пример конфигурации карьера
с несколькими локальными заглублениями

*Fig. 3.*An example of the pit configuration
with several local depths

3	3	2	1	1	3	3	3

3	~~2~~	3	1	1	2	3	3

Рис. 4. Примеры правильной (слева) и неправильной (справа)
расстановки временных меток для горизонтального слоя карьера

Fig. 4. An examples of regular (left) and irregular (right) time
stamp assignments for a horizontal layer of the pit

Поскольку стандартных процедур-распространителей, реализующих требуемую логику, авторам обнаружить не удалось, то был разработан собственный метод *linepropagator*. Для того чтобы дно карьера всегда в ходе выемки было требуемой формы, нужно задать ограничение *linepropagator* для всех горизонтальных слоев карьера.

Кратко опишем работу распространителя *linepropagator*, которая заключается в следующем. На первом этапе в массиве ищутся последовательно расположенные переменные, значения которых еще не установлены. Эти последовательности переменных образуют своеобразные «промежутки неопределенности». Причем перед каждым их таких промежутков и после него должны стоять переменные с уже конкретизированными значениями. Домены переменных, принадлежащих «промежутку неопределенности», редуцируются таким образом, чтобы эти переменные не могли принимать значения больше, чем максимальное из граничащих с ними значений конкретизированных переменных на концах этого промежутка. На рис. 5 показан первый этап работы распространителя.

Рис. 5. Первый этап работы распространителя

Fig. 5. The first stage of propagation

На втором этапе проводится проверка переменных, которым уже присвоено конкретное значение, на соответствие условию упорядоченности. На рис. 6 приведен пример прошедшего проверку массива и не прошедшего.

Рис. 6. Пример массивов, прошедшего (слева) и не прошедшего (справа) проверку на втором этапе работы распространителя

*Fig. 6.*An examples of arrays that passed (left) and failed (right) testing during the second stage of propagation

Таким образом, интересующее пользователя ограничение может быть выражено не только аналитически, но и путем описания логики работы соответствующей процедуры-распространителя, что удобно в случае необходимости учета и анализа нестандартных пользовательских ограничений.

ЗАКЛЮЧЕНИЕ

Предложенный подход на основе парадигмы программирования в ограничениях продемонстрировал свою применимость для решения практически значимых задач планирования открытых горных работ, характеризующихся высокой размерностью пространства поиска. Дальнейшие направления исследований связаны с разработкой гибридных методов, интегрирующих методы целочисленного программирования и методы программирования в ограничениях для получения точных решений рассматриваемого класса задач в пространствах еще более высокой размерности.

СПИСОК ЛИТЕРАТУРЫ

1. LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity / N. Boland, I. Dumitrescu, G. Froyland, A.M. Gleixner // Computers and Operations Research. – 2009. – Vol. 36, N 4. – P. 1064–1089.

2. *Caccetta L.* Application of optimization techniques in open pit mining // Handbook of Operations Research in Natural Resources. – Boston, MA: Springer, 2007. – P. 547–559. – (International Series in Operations Research).

3. A new algorithm for the open-pit mine production scheduling problem / R. Chicoisne, D.G. Espinoza, M. Goycoolea, E. Moreno, E. Rubio // Operations Research. – 2012. – Vol. 60, N 3. – P. 517–528.

4. *Dagdelen K.* A new linear programing approach to determining risk-quantified open pit mine production schedules incorporating mineral resource classification categories // Proceedings of the

38th International Symposium "Application of Computers and Operations Research in the Mineral Industry" (APCOM 2017). – Colorado, USA, 2017. – P. 451–460.

5. *Molina E.* Analytical properties of the feasible and optimal profiles in the binary programming formulation of open pit // Proceedings of the 38th International Symposium "Application of Computers and Operations Research in the Mineral Industry" (APCOM 2017). – Colorado, USA, 2017. – P. 603–610.

6. *Ramazan S.* The new Fundamental Tree Algorithm for production scheduling of open pit mines // European Journal of Operational Research. – 2007. – Vol. 177, N 2. – P. 1153–1166.

7. *Denby B., Schofield D.* Open-pit design and scheduling by use of genetic algorithms // Transactions of the Institution of Mining and Metallurgy, Section A: Mining Industry. – 1994. – Vol. 103. – P. A21–A26.

8. *Hazra T.* Genetic algorithm based approach for simultaneous optimization of open pit mine planning parameters // Proceedings of the 38th International Symposium "Application of Computers and Operations Research in the Mineral Industry" (APCOM 2017). – Colorado, USA, 2017. – P. 595–602.

9. *Zhang M.* Combining genetic algorithms and topological sort to optimize open-pit mine plans // Proceedings of the 15th Mine Planning and Equipment Selection. – Torino, Italy, 2006. – P. 1234–1239.

10. *Bartak R.* Constraint programming: in pursuit of the Holy Grail // Proceedings of the Week of Doctoral Students (WDS 99). – Prague: MatFyzPress, 1999. – Pt. 4. – P. 555–564.

11. *Rossi F., Beek P. van, Walsh T.* Handbook of constraint programming. – Boston, MA: Elsevier, 2006. – 955 p.

12. *Russel S., Norvig P.* Artificial intelligence: a modern approach. – 3rd ed. – Prentice Hall, 2010. – 1132 p.

13. Easy modeling of open pit mining problems via constraint programming / B. Crawford, R. Soto, C. Zec, E. Monfroy, F. Paredes // Communications in Computer and Information Science. – Cham: Springer, 2014. – Vol. 434. – P. 519–522.

14. *Régin J.* global constraints: a survey // Hybrid Optimization. Springer Optimization and Its Applications. – 2011. – Vol. 45. – P. 63–134.

15. *Régin J.* A filtering algorithm for constraints of difference in CSPs // Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI'94). – Menlo Park, CA, 1994. – P. 362–367.

16. *Beldiceanu N.* Pruning for the minimum Constraint Family and for the number of distinct values Constraint Family // Principles and Practice of Constraint Programming (CP'2001). – Paphos, Cyprus, 2001. – P. 211–224.

17. *Aggoun A., Beldiceanu N.* Extending CHIP in order to solve complex scheduling and placement problems // Mathematical and Computer Modelling. – 1993. – Vol. 17 (7). – P. 57–73.

18. *Jussien N., Rochart G., Lorca X.* Choco documentation. – URL: https://choco-solver.org/docs/ (accessed: 01.09.2021).

Зуенко Александр Анатольевич, кандидат технических наук, ведущий научный сотрудник, Институт информатики и математического моделирования ФИЦ КНЦ РАН. Основное направление научных исследований – искусственный интеллект, программирование в ограничениях, моделирование слабо формализованных предметных областей. Имеет более 100 печатных работ и учебных пособий. E-mail: zuenko@iimm.ru

Македонов Роман Александрович, младший научный сотрудник, Институт информатики и математического моделирования ФИЦ КНЦ РАН. Основное направление научных исследований – ситуационное моделирование, моделирование технических и бизнес-процессов. Имеет более 10 печатных работ и учебных пособий. E-mail: makedonov@iimm.ru

Олейник Юрий Андреевич, младший научный сотрудник, Институт информатики и математического моделирования ФИЦ КНЦ РАН. Основное направление научных исследо-

ваний – программирование в ограничениях, интеллектуальное планирование. Имеет более 20 печатных работ. E-mail: yoleinik@iimm.ru

Zuenko Aleksandr A., PhD (Eng.), leading researcher, IIMM KSC RAS. His research interests are currently focused on artificial intelligence, constraint programming, modeling of poorly formalized subject domains. He has more than 100 publications and teaching manuals. E-mail: zuenko@iimm.ru

Makedonov Roman A., junior researcher, IIMM KSC RAS. His research interests are currently focused on situational modeling, modeling of technical and business processes. He has more than 10 publications and teaching manuals. E-mail: makedonov@iimm.ru

Oleynik Yuri A., junior researcher, IIMM KSC RAS. His research interests are currently focused on constraint programming and AI planning. He has more than 20 publications. E-mail: yoleinik@iimm.ru

DOI: 10.17212/2782-2001-2021-3-99-114

*Intelligent search for accurate solutions to the planning open-pit mining**

A.A. ZUENKO[a], R.A. MAKEDONOV[b], Y.A. OLEYNIK[c]

Institute for Informatics and Mathematical Modeling, a subdivision of the Federal Research Centre "Kola Science Centre of the Russian Academy of Sciences" (IIMM KSC RAS), 24A Fersman Street, Apatity, 184209, Russian Federation

[a] zuenko@iimm.ru [b] makedonov@iimm.ru [c] yoleynik@iimm.ru

Abstract

A method of intelligent search for accurate solutions to the planning of open-pit mining has been developed. The method is implemented within the framework of the Constraint Programming Paradigm that allows us to process heterogeneous qualitative and quantitative constraints (in particular, economic, technological, etc.) simultaneously, as well as to maintain the model of subject domain being developed which is open to adding new constraints or deleting existing constraints. Various constraints can be added to the model, including those for which it is difficult to find a suitable analytical expression. In contrast to existing methods of local search the proposed method systematically explores the search space. The method allows us to find a global optimum in large dimensional spaces that describe practically significant problems arising in production. Currently, to solve this problem, the methods of integer linear programming are widely used. But its fundamental disadvantage is the need to represent all the constraints in the form of linear equalities and inequalities. However, in practice, some combinatorial optimization problems cannot be linearized and solved using traditional methods of mathematical programming. The developed method is illustrated by the example of a three-dimensional problem of finding the position of an intermediate pit wall by the processing periods taking into account the specified performance for mineral and overburden rocks and the objective profit function taking into account discounting. The types of constraints necessary for modeling the problem under consideration are identified. The possibility of applying the existing inference procedures on constraints is considered for these types. The method proposed makes it possible to obtain accurate solutions due to the intellectualization of the solution process by using highly efficient algorithms of reducing the search space for each type of constraints and specialized heuristics for pruning unpromising alternatives in the search tree.

---

* *Received 01 March 2021.*

The study was carried out with the financial support of the Russian Foundation for Basic Research within the framework of scientific projects No. 20-07-00708-a, 19-07-00359-a.

Keywords: Constraint programming, constraint satisfaction problem, constraint propagation, inference on constraints, AI (Artificial Intelligence) planning, open-pit mining, integer linear programming, developing model of subject domain

REFERENCES

1. Boland N., Dumitrescu I., Froyland G., Gleixner A.M. LP-based disaggregation approaches to solving the open pit mining production scheduling problem with block processing selectivity. *Computers and Operations Research*, 2009, vol. 36, no. 4, pp. 1064–1089.

2. Caccetta L. Application of optimization techniques in open pit mining. *Handbook of Operations Research In Natural Resources. International Series in Operations Research.* Boston, MA, Springer, 2007, pp. 547–559.

3. Chicoisne R., Espinoza D.G., Goycoolea M., Moreno E., Rubio E. A new algorithm for the open-pit mine production scheduling problem. *Operations Research*, 2012, vol. 60, no. 3, pp. 517–528.

4. Dagdelen K. A new linear programing approach to determining risk-quantified open pit mine production schedules incorporating mineral resource classification categories. *Proceedings of the 38th International Symposium* "*Application of Computers and Operations Research in the Mineral Industry*" (*APCOM 2017*), Colorado, USA, 2017, pp. 451–460.

5. Molina E. Analytical properties of the feasible and optimal profiles in the binary programming formulation of open pit. *Proceedings of the 38th International Symposium* "*Application of Computers and Operations Research in the Mineral Industry*" (*APCOM 2017*), Colorado, USA, 2017, pp. 603–610.

6. Ramazan S. The new Fundamental Tree Algorithm for production scheduling of open pit mines. *European Journal of Operational Research*, 2007, vol. 177, no. 2, pp. 1153–1166.

7. Denby B., Schofield D. Open-pit design and scheduling by use of genetic algorithms. *Transactions of the Institution of Mining and Metallurgy*, *Section A*: *Mining Industry*, 1994, vol. 103, pp. A21–A26.

8. Hazra T. Genetic algorithm based approach for simultaneous optimization of open pit mine planning parameters. *Proceedings of the 38th International Symposium* "*Application of Computers and Operations Research in the Mineral Industry*" (*APCOM 2017*), Colorado, USA, 2017, pp. 595–602.

9. Zhang M. Combining genetic algorithms and topological sort to optimize open-pit mine plans. *Proceedings of the 15th Mine Planning and Equipment Selection*, Torino, Italy, 2006, pp. 1234–1239.

10. Bartak R. Constraint programming: in pursuit of the Holy Grail. *Proceedings of the Week of Doctoral Students* (*WDS 99*), Prague, MatFyzPress, 1999, pt. 4, pp. 555–564.

11. Rossi F., Beek P. van, Walsh T. *Handbook of constraint programming*. Boston, MA, Elsevier, 2006. 955 p.

12. Russel S., Norvig P. *Artificial intelligence*: *a modern approach*. 3rd ed. Prentice Hall, 2010. 1132 p.

13. Crawford B., Soto R., Zec C., Monfroy E., Paredes F. Easy modeling of open pit mining problems via constraint programming. *Communications in Computer and Information Science*, Cham, Springer, 2014, vol. 434, pp. 519–522.

14. Régin J. Global constraints: a survey. *Hybrid Optimization. Springer Optimization and Its Applications*, 2011, vol. 45, pp. 63–134.

15. Régin J. A filtering algorithm for constraints of difference in CSPs. *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI'94)*, Menlo Park, CA, 1994, pp. 362–367.

16. Beldiceanu N. Pruning for the minimum Constraint Family and for the number of distinct values Con-straint Family. *Principles and Practice of Constraint Programming (CP'2001)*, Paphos, Cyprus, 2001, pp. 211–224.

17. Aggoun A., Beldiceanu N. Extending CHIP in order to solve complex scheduling and placement problems. *Mathematical and Computer Modelling*, 1993, vol. 17 (7), pp. 57–73.

18. Jussien N., Rochart G., Lorca X. *Choco documentation*. Available at: https://choco-solver.org/docs/ (accessed: 01.09.2021).

Для цитирования:

Зуенко А.А., Македонов Р.А., Олейник Ю.А. Интеллектуальный поиск точных решений задачи планирования открытых горных работ // Системы анализа и обработки данных. – 2021. – № 3 (83). – С. 99–114. – DOI: 10.17212/2782-2001-2021-3-99-114.

For citation:

Zuenko A.A., Makedonov R.A., Oleynik Yu.A. Intellektual'nyi poisk tochnykh reshenii zadachi planirovaniya otkrytykh gornykh rabot [Intelligent search for accurate solutions to the planning open-pit mining]. *Sistemy analiza i obrabotki dannykh = Analysis and Data Processing Systems*, 2021, no. 3 (83), pp. 99–114. DOI: 10.17212/2782-2001-2021-3-99-114.