УДК 004.832 **DOI** 10.15622/ia.23.5.1

А.А. Зуенко, Ю.А. Олейник

СОСТАВЛЕНИЕ РАСПИСАНИЙ КАК ЗАДАЧА УДОВЛЕТВОРЕНИЯ ОГРАНИЧЕНИЙ (НА ПРИМЕРЕ ПЛАНИРОВАНИЯ ОТКРЫТЫХ ГОРНЫХ РАБОТ)

Зуенко А.А., Олейник Ю.А. **Составление расписаний как задача удовлетворения ограничений (на примере планирования открытых горных работ).**

Аннотация. Описываемые в статье исследования направлены на развитие методов составления расписаний. Принципиальным недостатком существующих методов смешано-целочисленного линейного программирования в применении к рассматриваемым задачам является то, что они слишком требовательны к объемам оперативной памяти. Сложность же применения процедур локального поиска к подобным задачам высокой размерности состоит в разработке эффективного способа нахождения приемлемого первоначального приближения и определении функции перехода в соседнее состояние, которая бы позволила достаточно быстро достичь оптимума. В теории исследования операций добавление к задаче дополнительных условий может привести к принципиальному изменению используемой схемы решения задачи. Предлагаемые в статье методы реализованы в рамках парадигмы программирования в ограничениях, что позволяет более экономно с точки зрения оперативной памяти представлять зависимости предметной области, а также обеспечивает возможность поэтапного учета разнородных условий задачи без принципиального изменения схемы поиска решений. Существенная часть исследований посвящена использованию методов логического вывода на ограничениях для снижения размерности пространства поиска и ускорения процесса вычислений. Подход к составлению расписаний проиллюстрирован на задаче оптимизации планирования открытых горных работ, которую впервые предложено решать как задачу удовлетворения ограничений. Для нахождения первого допустимого решения предложен метод «жадного» поиска, результат применения которого затем может быть улучшен с помощью разработанного гибридного метода. Оба метода опираются на оригинальные процедуры вывода на ограничениях. Предложенный подход доказал свою эффективность для блочных моделей размерностью в десятки и сотни тысяч блоков.

Ключевые слова: задача удовлетворения ограничений, программирование в ограничениях, распространение ограничений, локальный поиск, смешано-целочисленная оптимизация, планирование открытых горных работ.

1. Введение. Методы составления расписаний в последнее время значительно продвинулись благодаря применению методов и инструментов удовлетворения ограничений [1].

Составление расписаний – это задача назначения множества операций (действий) множеству рабочих ресурсов, на которые (как на операции, так и на ресурсы) накладывается ряд ограничений. Совокупность действий заранее известна и требуется отыскать лишь их порядок и распределение по рабочим ресурсам. Многие задачи составления расписаний являются NP-трудными.

Принципиальным недостатком существующих методов смешано-целочисленного линейного программирования в применении к рассматриваемым задачам является то, что они слишком требовательны к объемам оперативной памяти [2]. Сложность же применения процедур локального поиска к подобным задачам высокой размерности состоит в разработке эффективного способа нахождения приемлемого первоначального приближения и определении функции перехода в соседнее состояние, которая бы позволила достаточно быстро достичь оптимума [3, 4].

Актуальность разработки новых методов решения задач составления расписаний, обуславливается возрастающей потребностью в средствах поддержки принятия решений, обеспечивающих снижение производственных затрат и максимизацию прибыли при анализе пространства альтернатив высокой размерности. В теории исследования операций основное внимание почти всегда уделяется методам решения задач составления расписаний конкретного типа или определению их сложности, а не разработке общего подхода к составлению расписаний. Иногда, добавление к задаче дополнительных условий может привести к принципиальному изменению используемой схемы решения задачи. В результате создано огромное количество алгоритмов для большого количества конкретных задач. Это сильно отличает составление расписаний в рамках традиционного подхода от подхода с использованием технологии программирования в ограничениях, где основное внимание уделяется разработке общих методов решения сложных комбинаторных задач.

Программирование в ограничениях обеспечивает хорошую основу для интеграции методов исследования операций и более общих схем решения комбинаторных задач. Подобная интеграция основана на понятии глобального ограничения. Глобальные ограничения инкапсулируют определенную часть задачи удовлетворения ограничений и, вместо того, чтобы использовать набор ограничений для моделирования конкретной подзадачи, применяется специальное «более крупное» ограничение, которое позволяет лучше анализировать структуру подзадачи. Глобальные ограничения реализуют эффективные алгоритмы для решения четко определенных подзадач. Их можно комбинировать с другими ограничениями, моделирующими некоторые дополнительные условия.

Парадигма программирования в ограничениях (ППО) является мощным средством для решения комбинаторных задач. В неё входит широкий спектр методов искусственного интеллекта, информатики и исследования операций [5, 6].

Главные принципы ППО, которые позволили авторам осуществить выбор в пользу данной парадигмы как методологии решения задач составления расписаний высокой размерности, заключаются в следующем:

– Для применения ППО решаемая задача должна быть представлена в декларативном виде как задача удовлетворения ограничений (ЗУО, *Constraint Satisfaction Problem*), то есть должны быть заданы множества: переменных, областей определения этих переменных (доменов), а также зависимостей на переменных, называемых ограничениями. *Решением* ЗУО является такое присваивание значений всем переменным, которое не нарушает ни одного ограничения [7].

– Порядок задания ограничений несущественен.

– Обеспечивается возможность унифицированной обработки разнородных ограничений. Ограничения могут задаваться с помощью арифметических выражений; логических формул; таблиц; выражений, формулируемых на языке специализированных теорий; а также с помощью уже упоминаемого механизма глобальных ограничений.

– Любой метод удовлетворения ограничений должен содержать две обязательных компоненты: а) компоненту, реализующую вывод (распространение); б) компоненту, реализующую поиск. Вывод (распространение) реализуется как целенаправленное сужение областей определения переменных. Эвристики, используемые в процедурах поиска, и методы вывода разрабатываются не под конкретную задачу, а под целый класс задач и зачастую являются универсальными.

– Обеспечивается возможность сопровождать модели, открытые для оперативных модификаций, то есть развивающиеся модели. При добавлении/удалении из модели ограничений нет необходимости принципиально изменять схему решения задачи.

В статье преимущества подхода к составлению расписаний на основе технологии программирования в ограничениях проиллюстрированы на таком классе задач исследования операций как оптимизация планирования открытых горных работ (Open Pit Mine Production Schedulling Problem). Впервые предложено решать задачи данного класса как задачи удовлетворения ограничений.

2. Постановка задачи ПОГР. Для моделирования и отображения процесса разработки карьера на компьютере часто используется блочная модель карьера (B), представляющая собой набор блоков одинакового размера ($b_i \in B$), на которые разбита порода, планируемая к выемке за заданное количество периодов (T) [8]. Каждый блок обладает набором свойств, таких как: координаты, процент содержания полезного ископаемого, а также базовая ценность блока. Значение базовой ценности позволяет отнести блок либо к блокам с полезными ископаемыми (ПИ), либо к блокам со вскрышными породами (ВП), на основе этого же параметра рассчитывается итоговая ценность блока, зависящая также от периода добычи блока [9].

Содержательная постановка задачи ПОГР 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 итоговая ценность блоков) от разработки карьера должна быть максимальной [10, 11].

Развитие карьера (его углубление и расширение) должно происходить с соблюдением требований (ограничений), представленных далее.

К таким ограничениям относятся ограничения на порядок извлечения блоков: для извлечения какого-то определенного блока необходимо, чтобы некоторые другие блоки были извлечены в тот же период или ранее. Эти ограничения носят технологический характер и определяются свойствами горных пород, составляющих карьер. Иногда подобные ограничения задаются с помощью так называемого шаблона выемки, в котором указывается некоторый «ключевой» блок и блоки непосредственно предшествующего ему уровня, подлежащие изъятию до него.

Помимо технологических в задаче следует также учитывать ряд экономических ограничений, которые обусловлены производительностью добывающей техники и перерабатывающих мощностей.

Приведем формализацию базовой задачи ПОГР с помощью линейных уравнений, как это предложено в [12]. Предполагается, что блочная модель задачи состоит из N блоков, планирование производится на T периодов добычи.

Для каждого блока b_i, где $i \in \{1, ..., N\}$ создается набор переменных $X_{ij} \in \{0, 1\}$, где $j \in \{1, ..., T\}$. $X_{ij} = 1$ обозначает, что блок i добыт в период j. Переменные X_{ij} должны удовлетворять ограничению:

$$\sum_{j=1}^{T} X_{ij} = 1, \quad \forall i \in \{1, ..., N\}. \tag{1}$$

Прибыль от добычи каждого блока i в определенный период задается явно, образуя множество конкретных значений C_{ij}, где $j \in \{1, ..., T\}$.

Целевая функция с учетом вышенаписанного представляется следующим образом:

$$\sum_{i=1}^{N} \sum_{j=1}^{T} (C_{ij} * X_{ij}). \tag{2}$$

Экономические ограничения на объемы добычи принимают вид:

$$\begin{array}{cc} W_j^{min} \leqslant \sum\limits_{i \in I^w} X_{ij} \leqslant W_j^{max}, & \forall j \in \{1, ..., T\}, \\ O_j^{min} \leqslant \sum\limits_{i \in I^o} X_{ij} \leqslant O_j^{max}, & I^w, I^o \subseteq \{1, ..., N\}, \end{array} \tag{3}$$

где I^w – множество индексов блоков с ВП, I^o – множество индексов блоков с ПИ, W_j^{min} и W_j^{max} – нижняя и верхняя границы нормы добычи вскрыши в период j, O_j^{min} и O_j^{max} – нижняя и верхняя границы нормы добычи руды в период j.

Для каждого блока b_i блочной модели определим множество $\{b_p\}$, где $p \in P_i$, как множество блоков, извлекаемых непосредственно до блока b_i согласно шаблону выемки. Технологические ограничения в таком представлении разворачиваются во множество линейных уравнений:

$$(\sum_{k=1}^{t} X_{pk} - X_{it}) \geqslant 0, \quad \forall i \in \{1, ..., N\}, \forall t \in \{1, ..., T\}, \forall p \in P_i. \tag{4}$$

Уравнения такого вида указывают, что блоки из множества P_i должны быть добыты не позже блока b_i. Если предположить, что шаблон добычи описывается с помощью M блоков, то для описания технологических ограничений потребуется в среднем $N * M * T$ уравнений. Так, например, для простейшего шаблона из 5 блоков на 5 периодов планирования, ограничение на порядок добычи породит в среднем 20 линейных уравнений на каждый блок, что при размерности задачи в 5000 блоков, даст уже 100000 уравнений только для описания технологических ограничений. С увеличением размерности задачи, количества периодов планирования и сложности шаблона добычи, одно только хранение порожденных линейных уравнений в памяти представляет существенную сложность, не говоря уже о дальнейшей процедуре решения.

Таким образом, использование методов целочисленного программирования не позволяет решать задачи требуемого уровня дискретизации блочной модели (сотни тысяч – миллионы блоков). Добавление дополнительных ограничений только усугубляет данную ситуацию.

В отличие от базовой, в рассматриваемой в настоящей работе постановке задачи ПОГР в качестве еще одного экономического ограничения выступает ограничение на среднее значение содержания полезного компонента в блоках с ПИ, извлекаемых за период. Данное ограничение представляется с помощью линейных уравнений следующим образом:

$$\begin{aligned} &\sum_{i=1}^{N}(g_i - G_j^{max})X_{ij} \leqslant 0, \\ &\sum_{i=1}^{N}(g_i - G_j^{min})X_{ij} \geqslant 0, \end{aligned} \qquad \forall j \in \{1, ..., T\}, \tag{7}$$

где:

g_i – содержание полезного компонента в блоке i;

G_j^{min} и G_j^{max} – нижняя и верхняя границы заданного диапазона допустимого среднего содержания полезного компонента в блоках с ПИ извлекаемых в период j.

3. Предлагаемый подход к решению задач ПОГР. В ходе исследований разработан новый подход к представлению и решению задач ПОГР на основе ППО. Особенность подхода состоит в применении процедур рассуждения на ограничениях на большинстве этапов решения рассматриваемого класса задач. Процедуры, реализующие подобные рассуждения, опираются на методы усечения доменов переменных.

Для представления задачи ПОГР как ЗУО каждому блоку b_i карьера сопоставляется соответствующая многозначная переменная ЗУО X_i. В качестве значений доменов переменных выступают номера возможных периодов разработки (от 1 до T). Рассуждения на ограничениях предложено реализовывать с использованием не только типовых, но и авторских методов логического вывода, предназначенных для повышения эффективности унифицированной обработки разнородных требований к проведению горных работ. В частности, для анализа ограничений на последовательность извлечения блоков (технологических ограничений) разработан метод выявления несовместных значений переменных «*Block sequencing*», который подробно описан ниже [13].

Экономические ограничения задачи формализуются в виде типовых глобальных ограничений *global cardinality constraint*, которые позволяют ограничивать количество переменных, принимающих определенное значение [14]. В рассматриваемой задаче используются 2 таких ограничения: $gcc(X^w, U, Y^w)$ и $gcc(X^o, U, Y^o)$, описывающих нормы добычи для блоков с ВП и ПИ соответственно, где:

X^w и X^o – множества переменных блоков с ВП и ПИ соответственно;

$U = \{1, ..., T\}$ – возможные номера периодов планирования;

Y^w и Y^o – наборы переменных, имеющих домены: $Y_j^w \in \{W_j^{min}, ..., W_j^{max}\}, Y_j^o \in \{O_j^{min}, ..., O_j^{max}\}, \quad \forall j \in \{1, ..., T\}$.

Управляя доменами переменных Y^w и Y^o, можно ограничивать количество переменных из X^w и X^o, принимающих определенные значения.

Особую сложность представляло моделирование требования на среднее значение полезного компонента за период добычи. В ходе исследований найдена такая форма представления данного ограничения, при которой не происходит снижение производительности вычислений по сравнению с вариантом задачи, не включающим это условие. Нормы среднего содержания полезного компонента описываются, парой ограничений $linear$ для каждого периода добычи [13]. Эти ограничения представляют по сути линейные уравнения, которые несколько отличаются от уравнений (7) из представленной ранее формализации задачи и имеют вид:

$$\begin{array}{l} \sum_{i=1}^{N}(g_i * X_{ij}) \leqslant N * G_j^{max}, \\ \sum_{i=1}^{N}(g_i * X_{ij}) \geqslant N * G_j^{min}, \end{array} \quad \forall j \in \{1, ..., T\}, \tag{8}$$

что записывается как $linear(g, X_j, «\leqslant», N * G_j^{max})$ и $linear(g, X_j, «\geqslant», N * G_j^{min})$ для каждого периода $j \in \{1, ..., T\}$.

Значение целевой функции в данном представлении рассчитывается как:

$$\sum_{i=1}^{N} FC_i, \tag{9}$$

где FC_i – итоговая ценность блока i, которая зависит от базовой стоимости блока и периода его добычи, т.е. может быть вычислена с помощью некоторой функции $cost(X_i, C_i)$. Значения FC_i используются достаточно часто в процессе поиска, поэтому каждый раз вычислять их нецелесообразно. Подобную функциональную зависимость можно описать с помощью табличного ограничения (таблица 1), где во втором столбце находятся значения функции $cost$, вычисляемые единожды при построении данного ограничения. Все шаги решения задачи можно разбить на три основных этапа, которые подробно излагаются ниже: a) предварительная обработка данных (предварительное упрощение задачи); b) поиск некоторого допустимого решения; c) улучшение первого полученного решения.

Таблица 1. Зависимость итоговой стоимости от периода разработки

X_i	FC_i
1	$cost(1, C_i)$
2	$cost(2, C_i)$
...	...
T	$cost(T, C_i)$

При формулировании задачи ПОГР в виде ЗУО применяется предварительный анализ блочной модели, позволяющий в некоторых случаях заранее сократить домены некоторых переменных до начала основного процесса решения, уменьшая, таким образом, общее пространство поиска.

Для нахождения за полиномиальное время одного из допустимых решений задачи ПОГР, то есть решения, удовлетворяющего одновременно всем ограничениям, предложен метод «жадного» поиска, использующий разработанные процедуры вывода на ограничениях. Обеспечена возможность одновременного поиска допустимого решения в различных вычислительных потоках. Дальнейшая оптимизация целевой функции реализуется разработанным гибридным методом [15], сочетающим преимущества систематического [16] и локального поиска [17]. Компонента, осуществляющая локальный поиск, на основе первоначально найденного допустимого решения выполняет формирование подпространств вариантов (альтернатив), где применяется уже систематический поиск решений задачи оптимизации ПОГР. В ходе систематического поиска также используются упомянутые выше методы рассуждений на ограничениях.

4. Предварительная обработка данных (априорное упрощение ЗУО). В ходе предварительного анализа для каждого блока b_i вычисляется минимальное количество рудных и вскрышных блоков, которые необходимо изъять из карьера перед ним. Затем производится уточнение диапазона периодов, когда возможна выемка блока b_i. Если для добычи какого-то блока необходимо предварительно изъять N_o рудных блоков и N_w вскрышных, при этом верхняя норма добычи за период по руде равна O_{max}, а по вскрыше – W_{max}, то нижняя граница домена соответствующего блока сократится до меньшего из значений $1 + (N_o/O_{max})$ и $1 + (N_w/W_{max})$ (деление целочисленное с округлением в меньшую сторону). Для получения для каждого блока b_i значений N_o и N_w необходимо рассмотреть блоки, зависящие друг от друга согласно шаблону, итеративно перебирая слой за слоем вверх.

Пример 1. Для иллюстрации упрощений ЗУО, реализуемых в рамках этапа предварительной обработки, рассмотрим модель, представленную на рисунке 1. Рудные блоки модели обозначены зеленым цветом, вскрышные – белым, внутри каждого блока указан домен соответсвующей переменной, ниже модели схематично представлен заданный в рамках примера шаблон добычи.

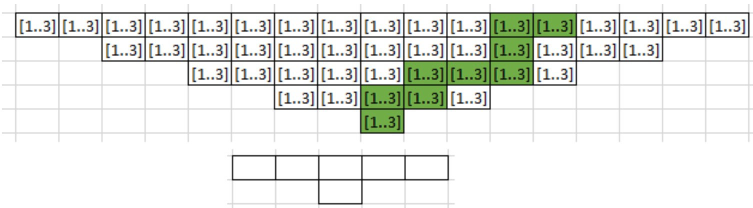


Рис. 1. Двумерная блочная модель карьера и шаблон извлечения

Нормы добычи для простоты будут заданы не в виде диапазона значений, а конкретными значениями: 3 рудных блока за период (O_{max}) и 12 вскрышных (W_{max}). Для каждого блока модели были вычислены значения N_w и N_o, результаты этих вычислений представлены на рисунке 2, где первое значение соответствует N_w, второе – N_o, сам блок также учитываются при подсчете этих значений.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0	1 0	0 1	0 1	1 0	1 0	1 0	1 0
B			6 0	6 0	6 0	6 0	6 0	6 0	6 0	5 1	4 2	3 3	4 2	4 2	5 1		
C					15 0	15 0	15 0	14 1	13 2	11 4	11 4	11 4	12 3				
D							26 2	24 4	22 6	21 7	22 6						
E									36 9								

Рис. 2. Расчет объемов добычи для извлечения каждого блока модели

Например, для блока B_3 множеством блоков, анализируемых согласно шаблону добычи блоков, является множество $S_{B_3} = \{B_3, A_1, A_2, A_3, A_4, A_5\}$, для блока B_4 – множество $S_{B_3} = \{B_4, A_2, A_3, A_4, A_5, A_6\}$ и т.д. Для блока C_5 множество анализируемых блоков можно получить следующим образом: $S_{C_5} = \{C_5\} \cup S_{B_3} \cup S_{B_4} \cup S_{B_5} \cup S_{B_6} \cup S_{B_7}$.

Тогда, на основе анализа множества S_{B_3}, ясно, что при извлечении блока B_3 минимальное количество вынимаемых вскрышных блоков равно 6, а рудных – 0. В результате аналогичных подсчетов для блока B_4 получаются числа 6 и 0, а для блока C_5 – числа 15 и 0. После вычисления значений N_w и N_o для каждого блока карьера, осуществляется сокращение доменов некоторых переменных задачи (рисунок 3).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]
B			[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]		
C					**[2..3]**	**[2..3]**	**[2..3]**	**[2..3]**	**[2..3]**	**[2..3]**	**[2..3]**	**[2..3]**	**[1..3]**				
D							**3**	**[2..3]**	**[2..3]**	**3**	**[2..3]**						
E									**3**								

Рис. 3. Сокращение доменов переменных в результате априорного упрощения ЗУО

Вычисления, реализуемые в ходе этапа предварительной обработки данных, позволили сократить начальное пространство поиска демонстрационной задачи с размерности в 3^{45} возможных присваиваний до размерности $3^{30} * 2^{12}$ присваиваний, т.е. более чем в 3500 раз. Для задач большей размерности с доменами переменных большей мощности эффект оказывается еще значительнее.

Далее начинается этап поиска первого решения задачи, за которым последует этап его улучшения. Оба этих этапа в значительной степени опираются на авторский метод вывода на ограничениях, названный «*Block sequencing*».

5. Глобальное ограничение «*Block sequencing*». В реальных задачах количество ограничений очень велико, и незначительное изменение домена распространителем какого-либо ограничения запускает целый каскад вызовов алгоритмов распространения для других ограничений. Поэтому целесообразно максимизировать сокращения доменов именно на этапе распространения отдельных ограничений (фильтрации) путем представления набора однотипных ограничений в виде одного глобального ограничения, для которого разрабатываются эффективные алгоритмы распространения. Так, например, одно глобальное ограничение ***alldifferent*** для N переменных, которое предписывает, чтобы все значения переменных были различными, заменяет $N(N-1)/2$ простейших неравенств. Аналогичная тенденция прослеживается и для других глобальных ограничений. Глобальные ограничения помимо ускорения процесса решения еще и упрощают описание ЗУО.

Для повышения эффективности решения комбинаторных задач ПОГР в рамках ППО разработан новый метод анализа несовместных значений переменных (метод вывода на ограничениях) *Block sequencing*. Метод позволяет совместно анализировать условия на последовательность извлечения для совокупности блоков, исключая, таким образом, большее количество неприемлемых значений из доменов переменных. Применение ограничения существенно снижает требования к оперативной памяти, что позволяет решать задачи с требуемым уровнем дискретизации модели месторождения.

Поясним используемые далее обозначения. Переменная X_j с доменом D_j – это переменная обозначающая «ключевой» блок, относительно которого выше по шаблону выемки лежат блоки X_{i_m}, где $i_m \in \{i_1, ..., i_p\}$, подлежащие выемке непосредственно до «ключевого», а ниже по шаблону находятся блоки X_{k_l}, где $k_l \in \{k_1, ..., k_r\}$. Очевидно, что $\{i_1, ..., i_p\} \cap \{k_1, ..., k_r\} = \varnothing$.

Алгоритм распространения *Block sequencing* состоит в следующем:

Шаг 0. Изначально все переменные модели, помещаются в очередь распространения Q. Затем циклически выполняются шаги 1-7.

Шаг 1. Если очередь пуста, то Шаг 8. Если очередь не пуста, то из нее извлекается переменная X_j.

Шаг 2. Для переменных X_{k_l}, где $k_l \in \{k_1, ..., k_r\}$, вычисляется $a = \min\limits_{k_l \in \{k_1,...,k_r\}} \{right\_bound(D_{k_l})\}$, то есть минимальная верхняя граница для всех их доменов D_{k_l}.

Шаг 3. В соответствии с найденным значением a урезается домен переменной X_j (если это возможно) – уточняется его верхняя граница: $D_j^* = \{t : (t \in D_j) \wedge (t \leqslant a)\}$. Вычисляется $a' = \min\{right\_bound(D_j^*), a\}$.

Шаг 4. Уточняются верхние границы доменов переменных $\{X_{i_m}\}$: $D^*_{i_m} = \{t : (t \in D_{i_m}) \wedge (t \leqslant a')\}$, где $i_m \in \{i_1, ..., i_p\}$. Те из переменных $\{X_{i_m}\}$, домены которых изменились, помещаются в очередь Q.

Шаг 5. Для переменных X_{i_m}, где $i_m \in \{i_1, ..., i_p\}$, вычисляется $e = \max\limits_{i_m \in \{i_1,...,i_p\}} \{left\_bound(D_{i_m})\}$, то есть максимальная нижняя граница для всех их доменов D_{i_m}.

Шаг 6. В соответствии с найденным значением e урезается домен переменной X_j (если это возможно) – уточняется его нижняя граница: $D^{**}_j = \{t : (t \in D_j) \wedge (t \geqslant e)\}$. Вычисляется $e' = \max\{left\_bound(D^{**}_j), e\}$.

Шаг 7. Уточняются нижние границы доменов переменных $\{X_{k_l}\}$: $D^*_{k_l} = \{t : (t \in D_{k_l}) \wedge (t \geqslant e')\}$, где $k_l \in \{k_1, ..., k_r\}$. Те из переменных $\{X_{k_l}\}$, домены которых изменились, помещаются в очередь Q. Сама переменная X_j удаляется из очереди Q. Возврат к шагу 1.

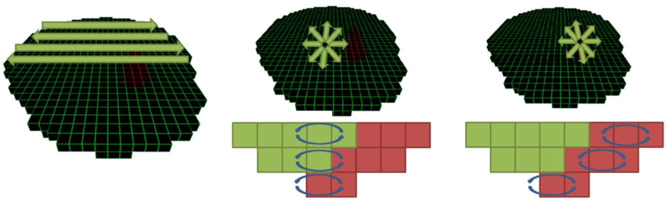
Шаг 8. Конец.

При попытке изменения домена любой переменной может возникнуть ситуация, что домен окажется пустым, тогда сообщается о противоречии, и распространитель завершает работу.

Если распространение завершилось без противоречий, производится проверка доменов переменных. Если встречена хотя бы одна переменная, в домене которой осталось больше 1 значения, то распространение достигло неподвижной точки, т.е. для дальнейшего решения задачи требуется запуск распространителей других ограничений или ветвление. Если домены всех переменных сократились до единственного значения, то ограничение считается удовлетворенным.

6. Этап поиска первого допустимого решения: «жадный» поиск решений задачи планирования открытых горных работ. Предлагаемый метод «жадного» поиска решения задачи ПОГР реализует целенаправленное (с использованием специализированных эвристик) движение по пространству поиска. Данное пространство сокращается в процессе движения с помощью механизма распространения ограничений. На каждом шаге осуществляется выбор некоторой переменной и присваивание ей определенного значения из соответствующего домена с последующим запуском распространителей ограничений задачи. Таким образом, из доменов переменных, которым еще предстоит присваивание, исключаются выявленные распространителями заведомо неприемлемые значения.

Далее опишем используемые эвристики выбора переменной и приписываемого ей значения. Значимым признаком упорядочивания блоков (соответствующих им переменных) является глубина залегания блока в карьере, т.е. координата Z (блоки рассматриваются горизонтальными слоями, начиная с нижнего слоя). Для обеспечения полного порядка на блоках, они упорядочиваются также и по двум остальным координатам в рамках одного горизонтального слоя модели. Были протестированы разные способы подобного упорядочивания. На ряде тестовых задач лучше всего показали следующие эвристики: a) обход блоков уровня зигзагом, b) выбор по расстоянию от проекции дна и c) выбор по расстоянию от среднего рудного блока в горизонтальном слое (рисунок 4).



a) b) c)

Рис. 4. Тестируемые эвристики «жадного» поиска решений: a) обход зигзагом; b) выбор по расстоянию от проекции дна; c) выбор по расстоянию от среднего рудного блока

При использовании всех трех эвристик, переменной присваивается максимальное из допустимых значений ее домена.

Описанная процедура «жадного» поиска потребляет сравнительно мало машинных ресурсов, поэтому для повышения вероятности нахождения допустимого решения она выполняется в параллельном режиме с использованием всех трех описанных выше эвристик одновременно. В случае нахождения допустимых решений несколькими эвристиками выбирается лучшее решение из найденных. Если ни одна из эвристик «жадного» поиска не позволяет найти допустимое решение, то запускается процедура систематического поиска. При этом для выбора переменной в ходе систематического поиска будет использована та из трех эвристик, с помощью которой «жадному» поиску удалось присвоить значения большему количеству переменных до обнаружения противоречий.

Для иллюстрации принципов работы «жадного» поиска вернемся к представленной ранее демонстрационной задаче из примера 1. Пусть для нее уже проведена предварительная обработка данных, результат которой был показан на рисунке 3. Поскольку блочная модель в примере является двумерной, то в качестве эвристики для выбора блоков (соответствующих переменных) будет применяться следующее правило: блоки выбираются слева направо, начиная с нижних слоев. В результате применения этой эвристики заполнение клеточек на рисунке 3 начнется снизу, причем сначала в клеточки будет записываться значение «3» (максимальное значение из доменов переменных). Распространение ограничений вызывается после каждого присваивания, однако значимые изменения в доменах станут появляться после присваивания значений всем рудным блокам в нижних двух рядах (D и E), когда норма добычи руды для третьего периода будет полностью выполнена. Это влечет удаление значения «3» из доменов переменных остальных рудных блоков, в результате чего в доменах рудных блоков ряда C остается только значение «2». Таким образом, полностью выбрана норма добычи руды и для второго периода, поэтому значения оставшихся рудных блоков в рядах A и B становятся равными «1». Далее на основании анализа ограничений на последовательность выемки блоков с помощью глобального ограничения «*Block sequencing*» уточняются домены переменных вскрышных блоков $A_6 - A_{11}$, $A_{14} - A_{16}$, $B_8 - B_{11}$, B_{13}, B_{14}, находящихся выше рудных согласно используемому шаблону (рисунок 5).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	**[1..2]**	**[1..2]**	**[1..2]**	**[1..2]**	**1**	**1**	**1**	**1**	**1**	**[1..2]**	**[1..2]**	[1..3]
B			[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	**[1..2]**	**[1..2]**	**[1..2]**	**[1..2]**	**1**	**[1..2]**	**[1..2]**	[1..3]		
C					[2..3]	[2..3]	[2..3]	[2..3]	[2..3]	**2**	**2**	**2**	[1..3]				
D							3	3	3	3	[2..3]						
E									3								

Рис. 5. Состояние задачи после набора рудных блоков для третьего периода и распространения ограничений

Распространение ограничений на текущем шаге больше не может быть выполнено, поэтому продолжается присваивание значений переменных согласно применяемой эвристике. Следующее значимое изменение произойдет, когда будет выбрана норма добычи вскрышных блоков за третий период. При этом значение «3» удаляется из доменов оставшихся переменных с еще не присвоенными значениями (рисунок 6).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	**[1..2]**	**[1..2]**	**[1..2]**	**[1..2]**	**[1..2]**	[1..2]	[1..2]	[1..2]	[1..2]	1	1	1	1	1	[1..2]	[1..2]	**[1..2]**
B			3	3	3	**[1..2]**	**[1..2]**	[1..2]	[1..2]	[1..2]	[1..2]	1	[1..2]	[1..2]	**[1..2]**		
C					3	3	3	3	3	2	2	2	3				
D							3	3	3	3	3						
E									3								

Рис. 6. Состояние задачи после набора всех блоков для третьего периода и распространения ограничений

Далее, согласно применяемой эвристике, вскрышным блокам будет назначаться значение «2» до момента, пока не будет выбрана норма вскрыши за второй период, после чего оставшимся блокам распространитель назначит значение «1». Полученное в результате этапа «жадного» поиска решение демонстрационной задачи представлено на рисунке 7.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B			3	3	3	2	2	2	2	2	2	1	2	2	2		
C					3	3	3	3	3	2	2	2	3				
D							3	3	3	3	3						
E									3								

Рис. 7. Полученное допустимое решение задачи

7. Этап улучшения полученного допустимого решения задачи. После получения допустимого решения запускается процедура гибридного поиска. В рамках данного этапа найденное решение задачи модифицируется путем ослабления некоторых его переменных, т.е. возврата их в начальное неприсвоенное состояние. Полученное таким образом новое сравнительно небольшое подпространство поиска затем исследуется систематическим методом с целью выявления новых решений с более высоким значением целевой функции. Систематический поиск в анализируемом подпространстве ограничивается по времени и/или по количеству найденных тупиковых веток дерева поиска. Фактически, параметр допустимого количества тупиковых ветвей задает степень «тщательности» исследования подпространства ослабленного решения и может варьироваться в зависимости от размерности задачи.

В случае нахождения нового решения с более высоким значением целевой функции, оно используется в качестве основы для дальнейшего поиска вместо первого найденного. Если такого решения не обнаружено, то исходное решение снова ослабляется иным образом, т.е. выбираются другие переменные для ослабления.

Признаком остановки процесса решения выступает отводимое на поиск время, а конечным решением задачи будет считаться последнее решение, найденное методом гибридного поиска.

Проиллюстрируем этот этап с помощью продолжения решения двумерной задачи из примера 1. В данном случае для ослабения первого допустимого решения, представленного на рисунке 7, будет выбрана часть переменных верхних двух рядов (ряды A и B), как это показано на рисунке 8.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	2	2	2	1	1	1	1	1	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]
B			3	3	3	2	2	2	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]	[1..3]		
C					3	3	3	3	3	2	2	2	3				
D							3	3	3	3	3						
E									3								

Рис. 8. Ослабление части переменных задачи

Теперь необходимо провести процедуру распространения ограничений, что уберет заведомо недопустимые значения из доменов ослабленных переменных. В данном случае, поскольку нормы добычи для третьего периода выбраны в неослабленной части модели, из доменов ослабленных переменных удаляется значение «3», также для второго периода выбраны нормы руды, поэтому ослабленным переменным рудных блоков остается только принять значение «1». Домены остальных переменных сокращаются с использованием глобального ограничения «*Block sequencing*». Состояние задачи после устранения несовместных значений переменных представлено на рисунке 9.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	2	2	2	1	1	1	1	1	[1..2]	1	1	1	1	1	[1..2]	[1..2]	[1..2]
B			3	3	3	2	2	2	[1..2]	[1..2]	[1..2]	1	[1..2]	[1..2]	[1..2]		
C					3	3	3	3	3	2	2	2	3				
D							3	3	3	3	3						
E									3								

Рис. 9. Результат распространения ограничений после ослабления части задачи

Подпространство поиска для состояния задачи, отображенного на рисунке 9, составляет 2^{10} присваиваний, что в $4 * 3^{30}$ раз меньше пространства поиска исходной задачи после предварительной обработки данных, что позволяет применять для исследования результирующего подпространства процедуры систематического поиска. В ходе исследования этого подпространства появляется возможность достаточно быстро найти новые решения задачи. При расчете целевой функции эти решения могут оказаться лучше исходного и стать основой для новой итерации.

8. Оценка эффективности предложенных методов. Для постановки задачи ПОГР, представленной выше в виде совокупности линейных уравнений, авторами проводились расчеты на тестовой машине с объемом оперативной памяти в 32 ГБ с использованием программного обеспечения Google OR-Tools. Машинных ресурсов не хватило даже для расчета задачи размерностью 2000 блоков. В литературе удалось найти пример решения задачи в близкой постановке методами целочисленного линейного программирования с помощью решателя IBM Cplex, где на схожей по параметрам тестовой машине для размерности уже порядка 20000 блоков не было получено ни одного решения за 48 часов расчетов [18]. Применяя предложенный подход, удалось получить решение для задачи размерностью 82000 блоков, что заняло около получаса, а за 5 часов было построено решение для задачи в 525000 блоков.

По сравнению с методами смешано-целочисленного программирования применение разработанного подхода позволяет существенно снизить требования к оперативной памяти за счет представления зависимостей предметной области с использованием глобальных ограничений, что позволяет решать задачи с требуемым уровнем дискретизации модели месторождения.

9. Заключение. Описываемые в статье исследования направлены на развитие методов составления расписаний. В статье предложено представлять и решать задачу оптимизации ПОГР как задачу удовлетворения ограничений. Разработанные методы реализованы в рамках парадигмы программирования в ограничениях, что позволяет более экономно с точки зрения оперативной памяти представлять зависимости предметной области, а также обеспечивает возможность поэтапного учета разнородных условий задачи без принципиального изменения схемы поиска решений. Существенная часть исследований посвящена использованию методов логического вывода на ограничениях для снижения размерности пространства поиска и ускорения процесса вычислений. Для нахождения допустимого решения предложен метод «жадного» поиска, результат работы которого может быть улучшен с помощью разработанного гибридного метода. Оба метода опираются на оригинальные процедуры вывода на ограничениях. Предложенный подход доказал свою эффективность для блочных моделей размерности в десятки и сотни тысяч блоков.

Литература

1. Baptiste Ph., Le Pape C., Nuijten W. Constraint-based scheduling: applying constraint programming to scheduling problems // Kluwer Academic Publishers. 2001. 198 p.
2. Patidar M., Bhardwaj R., Choudhary S. The study of linear programming approach for optimal scheduling of work in a corporation with different models // Materials Today: Proceedings. 2020. vol. 29. no. 2. pp. 661–667.
3. Ulaga L., Durasevic M., Jakobovic D. Local search based methods for scheduling in the unrelated parallel machines environment // Expert Systems with Applications. 2022. vol. 199.
4. Chen, Y., Lu J., He R., Ou J. An efficient local search heuristic for earth observation satellite integrated scheduling // Applied Sciences. 2020. vol. 10. no. 16.

5. Belaid M., Bessiere C., Lazaar N. Constraint Programming for Association Rules // Proceedings of the International Conference on Data Mining. Society for Industrial and Applied Mathematics. 2019. pp. 127–135.
6. Russell S., Norvig P., Davis E. Artificial intelligence: a modern approach –3 ed. // Upper Saddle River: Prentice Hall. 2010. 1132 p.
7. Narvaez D. Constraint Satisfaction Techniques for Combinatorial Problems // Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. 2018. vol. 32. no. 1. pp. 8028–8029.
8. Alipour A., Khodaiari A., Jafari A., Tavakkoli-Moghaddam R. An integrated approach to open-pit mines production scheduling // Resources Policy. 2022. vol. 75.
9. Novitasari R., Rosyidi C., Aisyati A. A Cut-off Grade Optimization Model in Multi Product Open Pit Mining Considering Reclamation and Valuable Waste Materials // IOP Conference Series: Materials Science and Engineering. 2021. vol. 1096. no. 1. DOI: 10.1088/1757-899X/1096/1/012021.
10. Tolouei K., Moosavi E., Tabrizi A., Afzal P., Bazzazi A. An optimization approach for uncertainty-based long-term production scheduling in open-pit mines using meta-heuristic algorithms // International Journal of Mining, Reclamation and Environment. 2021. vol. 35. no. 2. pp. 115–140.
11. Caccetta L. Application of Optimization Techniques in Open Pit Mining // Handbook of Operations Research in Natural Resources. Boston: Springer US, 2007. vol. 99. pp. 547–559.
12. Espinoza D., Goycoolea M., Moreno E., Newman A. MineLib: a library of open pit mining problems // Annals of Operations Research. 2013. vol. 206. no. 1. pp. 93–114.
13. Oleynik Y., Zuenko A. Open Pit Mine Production Scheduling using New Global Constraint // Proceedings of IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). 2022. pp. 1700–1704.
14. Regin J. Global Constraints: A Survey // Hybrid Optimization: The Ten Years of CPAIOR. 2011. pp. 63–134.
15. Mara S., Norcahyo R., Jodiawan P., Lusiantoro L., Rifai A. A survey of adaptive large neighborhood search algorithms and applications // Computers Operations Research. 2022. vol. 146.
16. Audemard G., Lecoutre C., Prudhomme C. Guiding Backtrack Search by Tracking Variables during Constraint Propagation // Proceedings of the 29th International Conference on Principles and Practice of Constraint Programming (CP 2023). Leibniz International Proceedings in Informatics (LIPIcs). 2023. vol. 280. pp. 9:1–9:17.
17. Kozik M. Solving CSPs using weak local consistency // SIAM Journal on Computing. 2021. vol. 50. no. 4. pp. 1263–1286.
18. Fathollahzadeh K., Mardaneh E., Cigla M., Waqar Ali Asad M. A mathematical model for open pit mine production scheduling with Grade Engineering and stockpiling // International Journal of Mining Science and Technology. 2021. vol. 31(4). pp. 717–728.

Зуенко Александр Анатольевич — канд. техн. наук, ведущий научный сотрудник, лаборатория информационных технологий управления промышленно-природными системами, ИИММ КНЦ РАН. Область научных интересов: программирование в ограничениях, моделирование слабо формализованных предметных областей. Число научных публикаций — 150. zuenko@iimm.ru; улица Ферсмана, 24А, 184209, Апатиты, Россия; р.т.: +7(815-55)79-782.

Олейник Юрий Андреевич — научный сотрудник, лаборатория информационных технологий управления промышленно-природными системами, ИИММ КНЦ РАН. Область научных интересов: программирование в ограничениях, моделирование природно-промышленных комплексов. Число научных публикаций — 19. yoleynik@iimm.ru; улица Ферсмана, 24А, 184209, Апатиты, Россия; р.т.: +7(815-55)79-248.

Поддержка исследований. Работа выполнена в рамках текущей темы НИР «Разработка теоретических и организационно-технических основ информационной поддержки управления жизнеспособностью региональных критических инфраструктур Арктической зоны Российской Федерации» (регистрационный номер 122022800547-3).

DOI 10.15622/ia.23.5.1

A. Zuenko , Yu. Oleynik

SCHEDULING AS A CONSTRAINT SATISFACTION PROBLEM (USING THE EXAMPLE OF OPEN-PIT MINE PRODUCTION SCHEDULING PROBLEM)

Zuenko A., Oleynik Yu. **Scheduling as a Constraint Satisfaction Problem (Using the Example of Open-Pit Mine Production Scheduling Problem).**

Abstract. The research described in the work is aimed at developing methods for Scheduling. The fundamental disadvantage of the existing methods of Mixed-Integer Linear Programming in application to the problems under consideration is the fact that they are too demanding on the amount of RAM. The difficulty of applying local search procedures to such high-dimensional problems is to develop an effective way to find an acceptable initial approximation and determine the neighboring state transition function, which would allow achieving the optimum fast enough. In the Operations Research Theory, adding additional conditions to a problem can lead to a fundamental change in the problem-solving scheme. The methods proposed in the study are implemented within the framework of the Constraint Programming Paradigm which makes it possible to represent the subject domain dependencies saving RAM, as well as to provide the ability step-by-step take into account heterogeneous problem conditions without essentially changing the scheme of finding solutions. A significant part of the research deals with methods of logical inference on constraints to reduce the search space and speed up the computational process. The approach to scheduling is illustrated by the Open-Pit Mine Production Scheduling Problem, which was first proposed to be solved as a Constraint Satisfaction Problem. In order to find the first feasible solution, a «greedy» search method is proposed, the result of which can be improved using the developed hybrid method. Both methods rely on original procedures of inference on constraints. The proposed approach has proven its efficiency for block models with sizes of tens and hundreds of thousands of blocks.

Keywords: constraint satisfaction problem, constraint programming, constraint propagation, local search, mixed-integer optimization, open pit mine production scheduling.

References

1. Baptiste Ph., Le Pape C., Nuijten W. Constraint-based scheduling: applying constraint programming to scheduling problems. Kluwer Academic Publishers. 2001. 198 p.
2. Patidar M., Bhardwaj R., Choudhary S. The study of linear programming approach for optimal scheduling of work in a corporation with different models. Materials Today: Proceedings. 2020. vol. 29. no. 2. pp. 661–667.
3. Ulaga L., Durasevic M., Jakobovic D. Local search based methods for scheduling in the unrelated parallel machines environment. Expert Systems with Applications. 2022. vol. 199.
4. Chen, Y., Lu J., He R., Ou J. An efficient local search heuristic for earth observation satellite integrated scheduling. Applied Sciences. 2020. vol. 10. no. 16.
5. Belaid M., Bessiere C., Lazaar N. Constraint Programming for Association Rules. Proceedings of the International Conference on Data Mining. Society for Industrial and Applied Mathematics. 2019. pp. 127–135.
6. Russell S., Norvig P., Davis E. Artificial intelligence: a modern approach –3 ed. Upper Saddle River: Prentice Hall. 2010. 1132 p.

7. Narvaez D. Constraint Satisfaction Techniques for Combinatorial Problems. Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. 2018. vol. 32. no. 1. pp. 8028–8029.
8. Alipour A., Khodaiari A., Jafari A., Tavakkoli-Moghaddam R. An integrated approach to open-pit mines production scheduling. Resources Policy. 2022. vol. 75.
9. Novitasari R., Rosyidi C., Aisyati A. A Cut-off Grade Optimization Model in Multi Product Open Pit Mining Considering Reclamation and Valuable Waste Materials. IOP Conference Series: Materials Science and Engineering. 2021. vol. 1096. no. 1. DOI: 10.1088/1757-899X/1096/1/012021.
10. Tolouei K., Moosavi E., Tabrizi A., Afzal P., Bazzazi A. An optimization approach for uncertainty-based long-term production scheduling in open-pit mines using meta-heuristic algorithms. International Journal of Mining, Reclamation and Environment. 2021. vol. 35. no. 2. pp. 115–140.
11. Caccetta L. Application of Optimization Techniques in Open Pit Mining. Handbook of Operations Research in Natural Resources. Boston: Springer US, 2007. vol. 99. pp. 547–559.
12. Espinoza D., Goycoolea M., Moreno E., Newman A. MineLib: a library of open pit mining problems. Annals of Operations Research. 2013. vol. 206. no. 1. pp. 93–114.
13. Oleynik Y., Zuenko A. Open Pit Mine Production Scheduling using New Global Constraint. Proceedings of IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). 2022. pp. 1700–1704.
14. Regin J. Global Constraints: A Survey. Hybrid Optimization: The Ten Years of CPAIOR. 2011. pp. 63–134.
15. Mara S., Norcahyo R., Jodiawan P., Lusiantoro L., Rifai A. A survey of adaptive large neighborhood search algorithms and applications. Computers Operations Research. 2022. vol. 146.
16. Audemard G., Lecoutre C., Prudhomme C. Guiding Backtrack Search by Tracking Variables during Constraint Propagation. Proceedings of the 29th International Conference on Principles and Practice of Constraint Programming (CP 2023). Leibniz International Proceedings in Informatics (LIPIcs). 2023. vol. 280. pp. 9:1–9:17.
17. Kozik M. Solving CSPs using weak local consistency. SIAM Journal on Computing. 2021. vol. 50. no. 4. pp. 1263–1286.
18. Fathollahzadeh K., Mardaneh E., Cigla M., Waqar Ali Asad M. A mathematical model for open pit mine production scheduling with Grade Engineering and stockpiling. International Journal of Mining Science and Technology. 2021. vol. 31(4). pp. 717–728.

Zuenko Alexander — Ph.D., Leading researcher, Laboratory of information technologies for control of industrial-natural systems, IIMM KSC RAS. Research interests: constraint programming, modeling in poorly formalized subject domains. The number of publications — 150. zuenko@iimm.ru; 24A, Fersmana St., 184209, Apatity, Russia; office phone: +7(815-55)79-782.

Oleynik Yurii — Researcher, Laboratory of information technologies for control of industrial-natural systems, IIMM KSC RAS. Research interests: constraint programming, modeling of nature-industrial complexes. The number of publications — 19. yoleynik@iimm.ru; 24A, Fersmana St., 184209, Apatity, Russia; office phone: +7(815-55)79-248.

Acknowledgements. The work was carried out within the framework of the current research topic «Development of theoretical and organizational and technical foundations of information support for managing the viability of regional critical infrastructures of the Arctic zone of the Russian Federation» (registration number 122022800547-3).