

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

DOI 10.17223/20710410/19/9

УДК 519.8

ИССЛЕДОВАНИЕ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ С ЛОГИЧЕСКИМИ ОГРАНИЧЕНИЯМИ НА ОСНОВЕ МЕТОДА РЕГУЛЯРНЫХ РАЗБИЕНИЙ¹

А. А. Колоколов*, А. В. Адельшин*, Д. И. Ягофарова**

** Омский филиал Института математики им. С. Л. Соболева СО РАН, г. Омск, Россия**** Омский государственный университет им. Ф. М. Достоевского, г. Омск, Россия***E-mail:** kolo@ofim.oscsbras.ru, adelshin@ofim.oscsbras.ru, ydarya@mail.ru

Рассматриваются различные постановки задач дискретной оптимизации с логическими ограничениями и возможности их применения для решения некоторых задач проектирования сложных изделий. Представлен обзор результатов исследования указанных задач на основе моделей целочисленного программирования и метода регулярных разбиений. С использованием данного подхода проведен анализ структуры и сложности задач, предложены алгоритмы решения задач выполнимости и максимальной выполнимости.

Ключевые слова: задача выполнимости, логические ограничения, целочисленное программирование, перебор L -классов.

Введение

Значительное число исследований в области дискретной оптимизации посвящено задаче выполнимости логической формулы и её обобщениям. Широкое практическое применение этих задач в экономике, управлении, проектировании и других областях является стимулом для разработки алгоритмов их решения. Алгоритмы решения задач, связанных с выполнимостью логической формулы, представлены, например, в [1 – 4]. Одним из подходов к анализу и решению задач выполнимости и максимальной выполнимости является использование аппарата целочисленного линейного программирования (ЦЛП).

Для анализа задач целочисленного программирования (ЦП), разработки и исследования алгоритмов, основанных на релаксации условия целочисленности, А. А. Колоколовым предложен метод регулярных разбиений [5]. Ранее с помощью этого подхода была исследована сложность решения ряда задач, изучена структура релаксационных множеств, введены новые классы отсечений, построены оценки числа итераций для некоторых известных алгоритмов, проведён анализ устойчивости задач и алгоритмов, разработаны новые алгоритмы.

В работах [6, 7] получен ряд теоретических результатов для задач с логическими условиями. Исследована L -структура многогранников задач выполнимости и максимальной выполнимости. Выделены семейства логических формул, для которых мощности L -накрытий указанных задач растут экспоненциально с увеличением числа пе-

¹Работа поддержана грантом РФФИ № 10-01-00598.

ременных. Получены оценки числа итераций некоторых алгоритмов целочисленного линейного программирования при решении задач из построенных семейств.

В данной работе рассматриваются некоторые из перечисленных результатов при исследовании задач выполнимости и максимальной выполнимости. Кроме того, строятся модели ЦЛП для задач проектирования на основе задач дискретной оптимизации с логическими ограничениями и приводятся алгоритмы решения задач выполнимости и максимальной выполнимости с использованием метода регулярных разбиений.

1. Задачи с логическими ограничениями

Рассмотрим постановку задачи выполнимости (SAT). Пусть $x_1, \dots, x_n$ — переменные, принимающие значение **истина** или **ложь**. Под литералом понимается либо переменная x_j , либо ее отрицание. Пусть логическая формула F представляет собой конъюнкцию формул (скобок) $C_i, i = 1, \dots, m$, каждая из которых является дизъюнкцией литералов. Требуется определить, выполнима ли формула F , т. е. существует ли такой набор значений переменных, при котором F принимает значение **истина**. Известно, что задача SAT является NP-полной, а в случае, когда каждая скобка содержит не более двух литералов, задача полиномиально разрешима [8]. Многие известные задачи теории графов, построения расписаний, криптографии могут быть сформулированы в виде задачи выполнимости некоторой логической формулы в конъюнктивной нормальной форме.

Важным обобщением задачи SAT является задача максимальной выполнимости (MAX SAT). Предположим, что каждой скобке C_i соответствует неотрицательный вес c_i , тогда задача MAX SAT состоит в отыскании набора значений логических переменных, при котором суммарный вес выполненных скобок будет наибольшим. Актуальными также являются постановки так называемых смешанных задач [4, 9], в которых присутствуют два вида логических ограничений — «жёсткие» и «мягкие», причем веса приписаны только «мягким». При этом требуется найти набор значений логических переменных, при котором выполнены все «жёсткие» ограничения, а суммарный вес выполненных «мягких» является наибольшим. Указанные задачи находят применение при решении многих практических задач, в частности в проектировании изделий сложной структуры [9–11].

Перейдём к построению моделей ЦЛП рассматриваемых задач. Введём булевы переменные $y_1, \dots, y_n$, такие, что y_j соответствует переменной x_j , а $(1 - y_j)$ — её отрицанию, т. е. $y_j = 1$ тогда и только тогда, когда переменная x_j принимает значение **истина**. Нетрудно показать, что условие выполнимости логической формулы F эквивалентно существованию решения системы

$$\sum_{j \in C_i^+} y_j - \sum_{j \in C_i^-} y_j \geq 1 - |C_i^-|, \quad i = 1, \dots, m; \quad (1)$$

$$0 \leq y_j \leq 1, \quad j = 1, \dots, n; \quad (2)$$

$$y_j \in \mathbb{Z}, \quad j = 1, \dots, n, \quad (3)$$

где C_i^- и C_i^+ — множества индексов переменных, входящих в скобку C_i с отрицанием и без него соответственно; $|C_i^-|$ — мощность множества C_i^- .

Модель ЦЛП для задачи выполнимости будет получена, если к последней системе добавить целевую функцию. При этом может быть выбрана, например, функция $f(y) = y_1 \rightarrow \max$ или $f(y) = \sum_{j=1}^n y_j \rightarrow \max$, где $y = (y_1, \dots, y_n)$. Здесь рассматри-

вается лексикографическая постановка задачи ЦЛП, т. е. ищется лексикографически максимальный вектор y , удовлетворяющий системе ограничений (1) — (3).

Модель ЦЛП для задачи максимальной выполнимости выглядит следующим образом:

$$y_0 = \sum_{i=1}^m c_i z_i \rightarrow \max; \quad (4)$$

$$\sum_{j \in C_i^-} y_j - \sum_{j \in C_i^+} y_j + z_i \leq |C_i^-|, \quad i = 1, \dots, m; \quad (5)$$

$$0 \leq y_j \leq 1, \quad y_j \in \mathbb{Z}, \quad j = 1, \dots, n; \quad (6)$$

$$0 \leq z_i \leq 1, \quad z_i \in \mathbb{Z}, \quad i = 1, \dots, m. \quad (7)$$

Здесь каждой скобке C_i поставлена в соответствие переменная z_i , причем в любом допустимом целочисленном решении $z_i = 1$ только в том случае, когда C_i выполнена. Таким образом, оптимальное значение целевой функции равно наибольшему суммарному весу выполненных скобок. Отметим, что если к (4) — (7) добавить ограничения (1), то получим модель ЦЛП для смешанной задачи с «жесткими» ограничениями (1) и «мягкими» ограничениями (5).

2. Прикладные постановки

Постановки задач с логическими ограничениями могут использоваться при анализе и решении многих прикладных задач. Отметим некоторые из них. В [9, 10] исследуются и решаются задачи проектирования сложных изделий из набора составляющих, работа [12] посвящена задаче формирования производственных групп с учетом межличностных отношений, в [13, 14] решаются задачи логического криптоанализа. Кроме того, большое число приложений задачи выполнимости приведено в обзоре [2].

Остановимся подробнее на задачах проектирования. На основе моделей задач дискретной оптимизации с логическими условиями в последнее время успешно развивается подход к автоматизации проектирования сложных изделий. Отметим, что вопросам автоматизации проектирования уделяется большое внимание, что обусловлено актуальностью этого направления в различных отраслях промышленности. К примеру, в [9, 11] отмечается применимость рассматриваемых задач для проектирования одежды, в [10] решаются задачи проектирования химического состава резин специального назначения.

Рассмотрим процесс проектирования сложных изделий, которые формируются из множества составляющих. В случае проектирования одежды к ним относятся различные крои рукавов, виды воротников, карманов, застёжек и т. д. Естественно, что не все составляющие входят в изделие одновременно, поскольку помимо функциональной нецелесообразности определённых сочетаний некоторые из них приводят к визуальной дисгармонии. Такие условия могут отражаться с помощью логических ограничений и приводить к задачам выполнимости и максимальной выполнимости логической формулы. При проектировании химического состава резин возникают ограничения на включение различных химических элементов в состав смесей, которые также могут описываться с помощью логических условий. Кроме того, наряду с логическими ограничениями учитываются конструкторские, технологические, экономические и др.

Приведём математическую модель для задачи проектирования сложных изделий. Введём необходимые обозначения:

J — множество номеров составляющих изделия, $J = \{1, \dots, n\}$;

- v_j — составляющая изделия с номером $j \in J$;
 x_j — логическая переменная, принимающая значение **истина**, если v_j входит в состав изделия, и значение **ложь** в противном случае, $j \in J$;
 s_j — вес составляющей v_j , характеризующий степень целесообразности её включения в изделие;
 p — нижняя граница для суммарного веса составляющих v_j , включённых в изделие;
 I — множество номеров логических формул, используемых в задаче, $I = \{1, \dots, m\}$;
 I_1 — множество номеров логических формул, которые должны быть обязательно выполнены, $I_1 \subseteq I$;
 C_i — логическая формула, соответствующая i -му логическому ограничению, которая представляет собой дизъюнкцию переменных x_j или их отрицаний, $i \in I$;
 c_i — вес формулы C_i , характеризующий степень необходимости её выполнения, $i \in I \setminus I_1$;
 K — множество номеров используемых ресурсов, $K = \{1, \dots, q\}$;
 a_{kj} — объём k -го ресурса, требуемый для изготовления j -й составляющей изделия, $k \in K$;
 b_k — имеющийся объём k -го ресурса, $k \in K$.

Задача состоит в отыскании значений логических переменных, при которых выполняются формулы C_i с номерами $i \in I_1$, ограничения по ресурсам и по суммарному весу включённых в изделие составляющих v_j , а вес выполненных формул C_i для $i \in I \setminus I_1$ максимален. Если поставленная задача разрешима, то в результате может быть получен один или несколько наборов составляющих, из которых формируются проектируемые изделия.

Модель ЦЛП рассматриваемой задачи имеет вид

$$f(z) = \sum_{i \in I \setminus I_1} c_i z_i \rightarrow \max; \quad (8)$$

$$\sum_{j \in C_i^-} y_j - \sum_{j \in C_i^+} y_j + z_i \leq |C_i^-|, \quad i \in I \setminus I_1; \quad (9)$$

$$\sum_{j \in C_i^-} y_j - \sum_{j \in C_i^+} y_j \leq |C_i^-| - 1, \quad i \in I_1; \quad (10)$$

$$\sum_{j \in J} a_{kj} y_j \leq b_k, \quad k \in K; \quad (11)$$

$$\sum_{j \in J} s_j y_j \geq p; \quad (12)$$

$$0 \leq y_j \leq 1, \quad y_j \in \mathbb{Z}, \quad j \in J; \quad (13)$$

$$0 \leq z_i \leq 1, \quad z_i \in \mathbb{Z}, \quad i \in I. \quad (14)$$

Здесь система (9) представляет собой набор «мягких» ограничений, а система (10) — «жёстких». Условия (11) описывают ресурсные ограничения, (12) — возможные требования проектировщика. В [9, 11] приводятся также более сложные модели, в которых учитываются группы составляющих и другие ограничения.

Рассмотрим задачу проектирования, состоящую в разработке рецептуры огнестойкой резины на основе смеси каучуков [10]. В дополнение к введённым обозначениям, через J_1 обозначим множество номеров антипиренов (ингредиентов, обладающих свойством огнестойкости), $J_1 \subset J$, а через r_j — вес, соответствующий ингредиенту v_j , который характеризует его огнестойкость, $j \in J_1$. Двухкритериальная модель ЦЛП этой

задачи может быть записана в виде системы (9), (10), (13), (14) с критериями

$$f(z) = \sum_{i \in I \setminus I_1} c_i z_i \rightarrow \max; \quad g(y) = \sum_{j \in J_1} r_j y_j \rightarrow \max,$$

в которых максимизируются общий вес «мягких» ограничений и суммарная огнестойкость включённых в состав ингредиентов.

В работе [10] представлены результаты решения указанной задачи. Все ингредиенты были разбиты на несколько групп в зависимости от их свойств. Разработаны логические ограничения, которые отражают недопустимость или желательность определённых комбинаций составляющих. Проведён вычислительный эксперимент с использованием алгоритма перебора L -классов и показана перспективность данного подхода.

3. О методе регулярных разбиений

Рассмотрим кратко подход, идея которого заключается в выделении семейства специальных разбиений релаксационного множества задачи. Особенности данного подхода подробно описаны в [5]. Приведём необходимые для дальнейшего изложения определения и обозначения.

Введём понятие лексикографического порядка. Для этого рассмотрим функцию

$$\eta(x, y) = \min\{i : x_i \neq y_i, i = 1, \dots, n\}, \quad x, y \in \mathbb{R}^n, \quad x \neq y.$$

Вектор x лексикографически больше (меньше) вектора y , т. е. $x \succ y$ ($x \prec y$), если $x \neq y$ и $x_w > y_w$ ($x_w < y_w$) для $w = \eta(x, y)$. Отношение $\succ$ представляет собой отношение строгого линейного порядка. Запись $x \succeq y$ означает, что либо $x \succ y$, либо $x = y$. Аналогично понимается $x \preceq y$.

Для множеств $X, Y \subset \mathbb{R}^n$ полагаем $X \succ Y$ ($X \prec Y$), если $x \succ y$ ($x \prec y$) при любых $x \in X$ и $y \in Y$.

Пусть $x, y \in \mathbb{R}^n$ и $x \succ y$. Будем говорить, что точки x и y *отделимы*, если найдётся точка $z \in \mathbb{Z}^n$, для которой выполняется $x \succeq z \succeq y$. Точку z назовем *отделяющей*.

Дадим определение L -разбиения. Точки $x, y \in \mathbb{R}^n$ ($x \succ y$) называются *L -эквивалентными*, если не существует отделяющей их точки $z \in \mathbb{Z}^n$. Это отношение эквивалентности порождает разбиение любого множества $X \subset \mathbb{R}^n$ на непересекающиеся *L -классы*. Фактор-множество X/L называется *L -разбиением* множества X . Указанное разбиение обладает рядом полезных свойств, применяемых при разработке и исследовании алгоритмов целочисленного программирования, в частности алгоритмов перебора L -классов. Отметим некоторые из них:

- 1) каждая точка $z \in \mathbb{Z}^n$ образует отдельный L -класс, остальные классы состоят из нецелочисленных точек и называются *дробными*;
- 2) если X — ограниченное множество, то фактор-множество X/L конечно;
- 3) любой дробный L -класс $V \in X/L$ можно представить в виде

$$V = X \cap \{x : x_1 = a_1, \dots, x_{r-1} = a_{r-1}, a_r < x_r < a_r + 1\},$$

где $a_i \in \mathbb{Z}$ ($i = 1, \dots, r$), $1 \leq r \leq n$.

Рассмотрим лексикографическую задачу ЦЛП следующего вида: найти лексикографически максимальную точку y^* множества $(M \cap \mathbb{Z}^n)$, т. е.

$$\text{найти } y^* = \text{lexmax}(M \cap \mathbb{Z}^n), \tag{15}$$

L -классов для решения задачи выполнимости предложен в работе [15]. Подробное его изложение и некоторые модификации, а также экспериментальные исследования содержатся в [1]. Первое описание алгоритма точного решения для невзвешенного варианта задачи максимальной выполнимости представлено в [16], во взвешенном случае — в [17].

4.1. Алгоритм перебора L -классов для задачи выполнимости

Рассмотрим лексикографическую постановку задачи ЦЛП (15), где M — многогранник, определяемый системой ограничений (1), (2).

Опишем идею метода перебора L -классов для задачи (15). Пусть L -накрытие задачи состоит из классов $V_1, V_2, \dots, V_p$, причем $V_1 \succ V_2 \succ \dots \succ V_p$.

Основной шаг метода заключается в переходе от одного класса из M_*/L к другому в порядке лексикографического убывания, начиная с лексикографически максимального L -класса. Алгоритм порождает последовательность точек $y^{(t)} \in M_*$, обладающую следующими свойствами:

- $y^{(t)} \succ y^{(t+1)}$, $t = 1, 2, \dots, s-1$; $s \leq p$;
- все точки принадлежат различным L -классам.

В общем случае для задачи ЦЛП переход от одного L -класса к другому выполняется путём решения некоторого числа задач линейного программирования. Для решения задачи выполнимости поиск представителей $y^{(t)}$ указанных классов можно осуществлять, не решая задач линейного программирования, что заметно повышает эффективность алгоритма. Это основано на следующем свойстве: в каждом L -классе многогранника M содержится *полуцелочисленная точка*, т. е. точка со значениями координат из множества $\{0, 1/2, 1\}$. Поиск представителей осуществляется с помощью специальной комбинаторной *процедуры* W , в процессе выполнения которой строится лексикографически убывающая последовательность недопустимых целочисленных точек с координатами, равными 0, 1 или 2. Отметим, что значение 2 используется для обеспечения лексикографической монотонности указанной последовательности. С этой целью можно также выбрать любое число, большее 1.

Приведём алгоритм перебора L -классов для задачи выполнимости (алгоритм LCE), который за конечное число шагов либо находит лексикографически максимальное решение в случае выполнимости формулы, либо указывает на её невыполнимость.

Предполагается, что перед началом работы алгоритма формула не содержит единичных скобок, т. е. скобок вида x_j или $\bar{x}_j$. В противном случае число переменных и скобок можно сократить путём присвоения таким переменным соответствующих значений. Если в ходе работы алгоритма в формуле появляется единичная скобка x_j , то будем считать, что переменной y_j значение 1 присвоено «автоматически» (аналогично для скобки $\bar{x}_j$ — значение 0).

Алгоритм завершает работу в двух случаях: если получено оптимальное решение задачи (15) или если не удаётся найти очередной L -класс. Показано, что трудоёмкость алгоритма LCE составляет $O(n^3 m \cdot |M_*/L|)$.

Алгоритм LCE

Шаг 0 (*поиск представителя первого L -класса*). Выполняем *процедуру* W (она описана ниже), начиная с недопустимой точки $w^{(0)1} = (2, \dots, 2)$. Если находим полуцелочисленную точку $y^{(1)} \in V_1$, то переходим к шагу 1 первой итерации.

Итерация t ($t \geq 1$).

Шаг 1. Начинаем с точки $y^{(t)} = (a_1^{(t)}, \dots, a_{s-1}^{(t)}, 1/2, y_{s+1}^{(t)}, \dots, y_n^{(t)})$, где $a_i^{(t)} \in \{0, 1\}$, $i = 1, \dots, s-1$; $y_j^{(t)} \in \{0, 1/2, 1\}$, $j = s+1, \dots, n$. Первую дробную координату $y_s^{(t)}$ округляем до 0. Выполняем *процедуру упрощения формулы*. В ходе этого преобразования некоторые координаты $y_j^{(t)}$ ($j = s+1, \dots, n$) «автоматически» примут значения 0 или 1. Обозначим полученную точку $\tilde{y}^{(t)}$. После упрощения формулы возникает один из следующих случаев.

- *Все скобки исключены.* Рассматривая точку $\tilde{y}^{(t)}$, переходим на шаг 4.
- *Скобки остались, противоречие не обнаружено.* Из вектора $\tilde{y}^{(t)}$ строим недопустимую точку $w^{(t)1} = (a_1^{(t)1}, \dots, a_{s-1}^{(t)1}, 0, w_{s+1}^{(t)1}, \dots, w_n^{(t)1})$, где $w_j^{(t)1} \in \{0, 1, 2\}$, причём если $\tilde{y}_j^{(t)} = 1/2$, то $w_j^{(t)1} = 2$, $j = s+1, \dots, n$. Переходим на шаг 2.
- *Обнаружено противоречие.* Возвращаемся к точке $y^{(t)}$, переходим на шаг 3.

Шаг 2. Выполняем *процедуру W*, начиная с недопустимой точки $w^{(t)1}$. Если найдём полуцелочисленную точку $y^{(t+1)} \in V_{t+1}$, то переходим к шагу 1 итерации $(t+1)$.

Шаг 3. Среди координат вектора $y^{(t)}$ ищем максимальную по номеру координату $y_p^{(t)} = 1$, которой это значение не было присвоено «автоматически» ($p < s$). Возможны следующие случаи.

- Такая координата *существует*, тогда продолжаем просматривать вектор $y^{(t)}$. Находим все переменные $y_j^{(t)}$ ($p < j \leq n$), которым значение 0 или 1 присвоено после назначения $a_p^{(t)} = 1$. Изменяем их значения на 1/2. Значение координаты с номером p также уменьшаем до 1/2. В результате получаем точку $y^{(t+1)} \in V_{t+1}$ и переходим на шаг 1 итерации $(t+1)$.
- Такой координаты *не существует*. В этом случае алгоритм завершает работу: *формула невыполнима*.

Шаг 4. Алгоритм завершает работу: *формула выполнима*. Оптимальное решение y^* задачи (15) строим из текущего вектора $\tilde{y}^{(t)}$ следующим образом:

$$y_j^* = \begin{cases} \tilde{y}_j^{(t)}, & \text{если } \tilde{y}_j^{(t)} \in \{0, 1\}, \\ 1, & \text{если } \tilde{y}_j^{(t)} = 1/2, \end{cases}$$

для $j = 1, \dots, n$.

Отметим, что на шаге 4 координатам со значением 1/2 можно произвольным образом присвоить булевы значения. Это даёт множество различных допустимых решений задачи (15).

Приведём описание процедуры W , используемой в алгоритме LCE.

Процедура W

Полагаем $\gamma = 1$.

Итерация γ .

Начинаем с вектора $w^{(t)\gamma} = (a_1^{(t)\gamma}, \dots, a_{s-1}^{(t)\gamma}, 0, w_{s+1}^{(t)\gamma}, \dots, w_n^{(t)\gamma})$, где $a_i^{(t)\gamma} \in \{0, 1\}$, $i = 1, \dots, s-1$; $w_j^{(t)\gamma} \in \{0, 1, 2\}$, $j = s+1, \dots, n$. Для получения представителя первого L -класса начинаем с вектора $w^{(0)1} = (2, \dots, 2)$. Среди координат этого вектора ищем первую по номеру координату $w_q^{(t)\gamma}$, значение которой равно 2, и изменяем его на 1. Выполняем *процедуру упрощения формулы*. В результате некоторые компоненты вектора «автоматически» примут значения 0 или 1. Обозначим полученный вектор $\tilde{w}$. После упрощения формулы возникает один из следующих случаев:

- Все скобки исключены. В векторе $\tilde{w}$ заменяем все координаты, равные 2, на $1/2$. Полученную точку обозначаем $\tilde{y}^{(t)}$ и переходим на шаг 4 алгоритма LCE.
- Скобки остались, противоречия не обнаружено. Вектор $\tilde{w}$ обозначаем $w^{(t)\gamma+1}$ и переходим на $(\gamma + 1)$ -ю итерацию.
- Обнаружено противоречие. Из вектора $w^{(t)\gamma}$ строим точку $y^{(t+1)}$: если $w_j^{(t)\gamma} = 2$, то полагаем $y_j^{(t+1)} = 1/2$, $j = s + 1, \dots, n$. Таким образом, значение q -й координаты меняется на $1/2$. Возвращаемся на шаг 2 алгоритма LCE.

4.2. Алгоритм для задачи максимальной выполнимости

Приведём описание точного алгоритма решения задачи максимальной выполнимости, который за конечное число итераций находит оптимальное решение задачи. Алгоритм основан на лексикографическом переборе целочисленных векторов и сведении к последовательности задач выполнимости, для решения которых используется алгоритм LCE. По сравнению с алгоритмом из работы [16] для невзвешенной задачи, в данном алгоритме содержатся дополнительные процедуры, оптимизирующие перебор указанных векторов с целью сокращения числа решаемых подзадач выполнимости.

Пусть формуле $F = C_1 \wedge \dots \wedge C_m$ соответствует вектор $Z = (Z_1, \dots, Z_m)$, координаты которого принадлежат множеству $\{0, 1, 2\}$. Основные процедуры алгоритма заключаются в направленном переборе указанных векторов, каждому из которых соответствует следующая задача: найти такой набор значений переменных, что скобки C_p , для которых $Z_p = 1$, становятся истинными, а скобки C_q , где $Z_q = 0$, — ложными. Если задача выполнимости, соответствующая вектору Z , разрешима, то будем называть этот вектор выполнимым, иначе — невыполнимым. Отметим, что координаты $Z_i = 2$ используются для обеспечения монотонности лексикографического перебора векторов, с этой целью можно использовать любое значение, большее 1. Пусть все скобки, входящие в формулу, предварительно упорядочены по невозрастанию их весов. При описании алгоритма будем пользоваться следующими обозначениями:

$W_0(Z)$ — суммарный вес всех скобок C_i формулы F , для которых $Z_i = 0$;

$W_1(Z)$ — суммарный вес всех скобок C_i формулы F , для которых $Z_i = 1$;

$subSAT(F, Z)$ — функция для решения специальной задачи выполнимости с помощью алгоритма LCE. Её особенность заключается в следующем. Если находится такой выполняющий набор, при котором скобки C_p (для координат $Z_p = 1$) становятся истинными, а скобки C_q (для $Z_q = 0$) — ложными, то функция принимает значение 0. В противном случае функция принимает такое значение s , что если положить $Z_i = 2, i = s, \dots, m$, то вектор Z останется невыполнимым.

Схема точного алгоритма для MAX SAT

Шаг 0. Получаем приближённое решение $\tilde{x}$ задачи каким-нибудь эвристическим алгоритмом. Пусть $\tilde{f} = f(\tilde{x})$ — суммарный вес выполненных скобок формулы F , соответствующий набору $\tilde{x}$. Полагаем $Z_i = 2, i = 1, \dots, m$, и переходим на шаг 1.

Шаг 1. Если $\tilde{f} = \sum_{i=1}^m w_i$, алгоритм завершает работу ($\tilde{x}$ — оптимальное решение задачи); в противном случае переходим на шаг 2.

Шаг 2. Если выполняется $W_1(Z) \leq \tilde{f}$, то переходим на шаг 3; в противном случае — на шаг 7.

Шаг 3. Если имеет место $\sum_{i=1}^m w_i - W_0(Z) \leq \tilde{f}$, то переходим на шаг 4; в противном случае — на шаг 6.

Шаг 4. Пусть $\mu = \min\{i : Z_i = 0, i = 1, \dots, m\}$. Если среди координат Z_i , $i = \mu, \dots, m$, не более одной, равной 0, то алгоритм завершает работу ($\hat{x}$ — оптимальное решение задачи); в противном случае — на шаг 5.

Шаг 5. Находим $\nu = \max\{i : Z_i = 0, i = 1, \dots, m\}$, $k = \max\{i : Z_i = 0, i = 1, \dots, \nu - 1\}$ и $l = \max\{i : Z_i = 1, i = 1, \dots, k - 1\}$. Полагаем $Z_l = 0$, $Z_i = 2$, $i = l + 1, \dots, m$, и переходим на шаг 2.

Шаг 6. Пусть $r = \max\{i : Z_i \neq 2, i = 1, \dots, m\}$ при $Z_1 \neq 2$ и $r = 0$ в противном случае. Полагаем $Z_{r+1} = 1$ и переходим на шаг 2.

Шаг 7. Пусть $s = \text{subSAT}(F, Z)$. Если найден выполняющий набор x (в этом случае $s = 0$), то обновляем рекордное значение целевой функции и переходим на шаг 1; в противном случае переходим на шаг 8.

Шаг 8. Если $\{i : Z_i = 1, i = 1, \dots, s\} = \emptyset$, алгоритм завершает работу. В противном случае находим номер $l = \max\{i : Z_i = 1, i = 1, \dots, s\}$ и полагаем $Z_l = 0$, $Z_i = 2$, $i = l + 1, \dots, m$. Переходим на шаг 2.

Предложенные алгоритмы реализованы и проведены вычислительные эксперименты как на случайно сгенерированных задачах, так и на семействах логических формул из электронных библиотек тестовых задач. В [1, 17] приведены результаты расчётов для различных серий задач и сравнение алгоритмов с другими известными алгоритмами для задач выполнимости и максимальной выполнимости. В частности, в работе [17] алгоритм для задачи максимальной выполнимости сравнивается с основанным на методе ветвей и границ алгоритмом WMaxSatz-2.5 из библиотеки MAX-SAT Evaluation 2009. Кроме того, в указанной работе описывается возможность использования предложенного алгоритма для приближённого решения задачи максимальной выполнимости. Результаты эксперимента показали эффективность используемого подхода на ряде серий задач и целесообразность его дальнейшего развития. В настоящее время ведётся работа по применению разработанных схем алгоритмов для эффективного решения смешанных задач с логическими ограничениями.

Заключение

Метод регулярных разбиений, предложенный для анализа и решения задач целочисленного программирования, получил значительное развитие при исследовании задач о покрытии и упаковке множества, размещения, задачи о рюкзаке и других. В данной работе описаны результаты последних лет по его использованию для исследования задач выполнимости и максимальной выполнимости, которые находят широкое практическое применение во многих областях. В дальнейшем представляется целесообразным продолжение исследований задач дискретной оптимизации с логическими ограничениями на основе указанного подхода.

ЛИТЕРАТУРА

1. Колоколов А. А., Адельшин А. В., Ягофарова Д. И. Решение задачи выполнимости с использованием метода перебора L -классов // Информационные технологии. 2009. № 2. С. 54–59.
2. Gu J., Purdom P., Franco J., and Wah B. Algorithms for the satisfiability (SAT) problem: a survey // DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Princeton: AMS, 1996. P. 19–152.
3. Hansen P. and Jaumard B. Algorithms for the maximum satisfiability problem // Computing. 1990. No. 4. P. 279–303.

4. Thornton J., Bain S., Sattar A., and Pham D.N. A two level local search for MAX-SAT problems with hard and soft constraints // Proc. 15th Australian Joint Conference on Artificial Intelligence, AustAI-2002. Canberra, Australia, 2002. P. 603–614.
5. Колоколов А. А. Регулярные разбиения и отсеечения в целочисленном программировании // Сиб. журнал исследования операций. 1994. № 2. С. 18–39.
6. Адельшин А. В. Исследование задач максимальной и минимальной выполнимости с использованием L -разбиения // Автоматика и телемеханика. 2004. № 1. С. 35–42.
7. Колоколов А. А., Адельшин А. В., Чередова Ю. Н. Применение L -разбиения к исследованию некоторых задач выполнимости // Труды XII Байкальской междунар. конф. «Методы оптимизации и их приложения». Иркутск, 2001. Т. 1. С. 166–172.
8. Cook S. A. The complexity of theorem-proving procedure // Proc. 3rd Annual ACM Symposium on the Theory of Computing. New York: ACM, 1971. P. 151–159.
9. Колоколов А. А., Ярош А. В. Автоматизация проектирования сложных изделий с использованием дискретной оптимизации и информационных технологий // Омский научный вестник. 2010. № 2(90). С. 234–238.
10. Адельшин А. В., Жовнер Е. Н. Применение задач выполнимости логической формулы для проектирования химического состава резин // Вестник Омского университета. 2011. № 2. С. 14–18.
11. Гуселетова О. Н., Колоколов А. А. Решение задач дискретной оптимизации с логическими ограничениями при проектировании сложных изделий // Автоматика и телемеханика. 2008. № 10. С. 176–182.
12. Колоколов А. А., Серышева Ю. С., Шулепова Л. Д. Решение задач формирования малых групп с учетом межличностных отношений // Труды XV Байкальской междунар. школы-семинара «Методы оптимизации и их приложения». Иркутск, 2011. Т. 5. С. 61–66.
13. Посыпкин М. А., Заикин О. С., Беспалов Д. В., Семенов А. А. Решение задач криптоанализа поточных шифров в распределенных вычислительных средах // Труды ИСА. 2009. Т. 46. С. 119–137.
14. Massacci F. and Marraro L. Logical cryptanalysis as a SAT problem // J. Automated Reasoning. 2000. No. 24. P. 165–203.
15. Колоколов А. А., Чередова Ю. Н. Исследование и решение задачи о выполнимости с использованием L -разбиения // Материалы междунар. конф. «Дискретный анализ и исследование операций». Новосибирск, 2000. С. 150.
16. Колоколов А. А., Адельшин А. В., Ягофарова Д. И. Алгоритмы лексикографического перебора для решения задачи выполнимости и некоторых её обобщений // Труды XIII Байкальской междунар. школы-семинара «Методы оптимизации и их приложения». Иркутск, 2005. Т. 1. С. 503–508.
17. Адельшин А. В., Кучин А. К. Алгоритмы точного и приближенного решения задачи максимальной выполнимости // Омский научный вестник. 2011. № 1(97). С. 5–9.