УДК 519.68

ЛОКАЛЬНЫЕ ЭЛИМИНАЦИОННЫЕ АЛГОРИТМЫ ДЛЯ ЗАДАЧ УДОВЛЕТВОРЕНИЯ ОГРАНИЧЕНИЙ

© О.А. Щербина

University of Vienna,
Institut für Mathematik
Nordbergstr. 15, Vienna 1090, Austria
e-mail: *oleg.shcherbina@univie.ac.at*

In this paper local elimination algorithms of decomposition of constraint satisfaction problems are considered.

Введение

Использование подходов и алгоритмов искусственного интеллекта (ИИ) позволяет решать многие прикладные задачи, такие, как задачи теории расписаний [10], задачи проектирования экспертных систем и систем поддержки принятия решений [3], доказательство теорем, задачи тестирования электронных схем, обработка изображений. Одной из важных задач ИИ является задача удовлетворения ограничений (constraint satisfaction problem) [14], [23]. К сожалению, большинство интересных задач ИИ являются NP-трудными и решение их в худшем случае может требовать перебора экспоненциального числа решений. Многие практические задачи содержат огромное число переменных и/или ограничений, что создает сложности при попытке решения этих задач с помощью современных решателей.

Перспективными декомпозиционными подходами, использующими структуру разреженных графов, описывающих задачи ИИ, являются графовые декомпозиционные методы [18], интерес к которым возрос в последнее время, что обусловлено результатами Courcelle [13], Arnborg et al. [9], доказавших, что ряд NP-трудных задач, поставленных в монадической логике второго порядка, могут быть решены за полиномиальное время с помощью методов динамического программирования на графах, описывающих структуру задачи, с ограниченной древовидной шириной.

К графовым декомпозиционным подходам относится класс локальных элиминационных алгоритмов (ЛЭА) вычисления информации [4], включающий локальные алгоритмы декомпозиции [1, 6], алгоритмы несериального динамического программирования (НСДП) ([11, 5, 20], алгоритмы сегментной элиминации [15], методы древовидной декомпозиции [18].

В [5, 20] рассмотрено применение ЛЭА для решения задач дискретной оптимизации (ДО). Успехи графовых декомпозиционных схем, позволяющих справиться с решением NP-трудных задач с помощью алгоритмов динамического программирования [18], вызвали интерес к применению этих методов и в областях, отличных от оптимизации. ЛЭА может быть использовано и для решения неоптимизационных задач, которые можно разбить на подзадачи и использовать полученные решения меньших подзадач при решении больших. Связь НСДП с локальными алгоритмами ([1, 6, 5]) обусловлена тем, что, как и локальные алгоритмы, НСДП решает задачи, преобразуя локальную информации в глобальную. Одной из общих для этих методов графовых интерпретаций является *элиминационная игра* (Parter [21]).

Для решения ряда разреженных дискретных задач, возникающих в самых разных областях, были предложены обобщения принципа динамического программирования, такие как обобщение Dechter [15] (bucket elimination), Shenoy [22] и др. Интересно заметить, что все прикладные задачи, для которых были предложены отмеченные обобщения, характеризовались возможностью находить решения всей задачи на базе локальных решений, анализируя окрестности элементов задачи. Автор считает важным подчеркнуть именно возможность вычисления и дальнейшего использования *локальной информации* (т.е. информации об элементах окрестности) при решении этих задач и предложить общий класс локальных элиминационных алгоритмов вычисления информации для их решения, позволяющих осуществлять вычисление *глобальной информации* с помощью *локальных вычислений.*

В настоящей статье рассмотрен класс локальных элиминационных алгоритмов вычисления информации и их применение при решении задач удовлетворения ограничений.

1. Задачи удовлетворения ограничений

1.1. **Основные понятия.** Задачи удовлетворения ограничений (УО), известные в англоязычной литературе как constraint satisfaction problems (CSP) [14, 23], широко используются при решении ряда практически важных задач ИИ, таких как составление расписаний, проектирование электронных схем, поддержка принятия решений. В данной статье рассматриваются задачи УО с дискретными переменными. При задании ограничений используются *отношения.*

Определение 1. Для данных множества переменных $X = \{x_1, \ldots, x_n\}$ и соответствующих их областей значений $D_1, \ldots, D_n$ *отношением* R на множестве переменных называется любое подмножество декартова произведения их областей значений. Множество переменных, на котором определено отношение R, называется *диапазоном* отношения и обозначается $scope(R)$.

Если $R = D_1 \times \ldots \times D_n$, то R называется *универсальным отношением.*

Отношения могут задаваться с помощью таблиц, описывающих допустимые сочетания значений переменных (табл. 1).

Таблица 1. Отношение R

x_1	x_2	x_3
a	b	c
b	b	c
c	b	c
c	b	s

Определение 2. Задача УО (ЗУО) определяется множеством дискретных переменных $x_1, \ldots, x_n$, для каждой из которых задана область определения или домен $D_j = \{d_j^{(1)}, \ldots, d_j^{(p_j)}\}$ $(j = 1, \ldots, n)$, и множеством ограничений. *Ограничением* называется пара (R, S), где R – отношение, определенное на диапазоне S. *Решением* ЗУО называется присвоение значений всем переменным, которое удовлетворяет всем ограничениям. *Целью* решения ЗУО может быть нахождение одного или всех решений.

Действия над отношениями. Пусть R_1, R_2 – два отношения с одинаковым диапазоном. Тогда *пересечением* $R_1 \cap R_2$ называется отношение, содержащее все наборы значений переменных, которые имеются одновременно в обоих отношениях R_1 и R_2.

Объединение $R_1 \cup R_2$ – отношение, содержащее все наборы значений переменных, которые имеются либо в R_1, либо в R_2, или в обоих отношениях.

Разностью $R_1 - R_2$ называется отношение, содержащее наборы значений переменных, содержащиеся в R_1, но не содержащиеся в R_2.

Проекция $\Pi_Y(R)$ отношения R на множество переменных $Y \subseteq scope(R)$ является отношением, содержащим наборы значений только переменных, содержащихся в Y. В табличном представлении отношений проекция выбирает подмножество столбцов, соответствующих множеству Y.

Соединение отношений. Оператор соединения $R_S \bowtie R_T$ из двух отношений R_S с диапазоном S и R_T с диапазоном T строит новое отношение, состоящее из их общих переменных в S и T. Набор r из соединения $R_S \bowtie R_T$ отношений R_S и R_T можно построить, используя следующие шаги: (1) взять набор s из R_S; (2) выбрать набор t из R_T такой, что компоненты s и t согласуются по переменным из $S \cap T$, общим для R_S и R_T; (3) образовать новый набор r, комбинируя компоненты s и t, сохраняя лишь один из полученных одинаковых наборов. Диапазон получающегося отношения – $S \cup T$. Соединение двух отношений с одинаковыми диапазонами эквивалентно пересечению этих отношений.

Пример 1. Отношения R_1, R_2, R_3:

R_1:

x_1	x_2	x_3
a	b	c
b	b	c
c	b	c
c	b	s

R_2:

x_1	x_2	x_3
b	b	c
c	b	c
c	n	n

R_3:

x_2	x_3	x_4
a	a	1
b	c	2
b	c	3

Результаты действий над отношениями из примера 1

$R_1 \cap R_2$:

x_1	x_2	x_3
b	b	c
c	b	c

$R_1 \cup R_2$:

x_1	x_2	x_3
a	b	c
b	b	c
c	b	c
c	b	s
c	b	n

$R_1 - R_2$:

x_1	x_2	x_3
a	b	c
c	b	s

$\Pi_{\{x_2,x_3\}} R_2$:

x_2	x_3
b	c
n	n

$R_2 \bowtie R_3$:

x_1	x_2	x_3	x_4
a	b	c	2
b	b	c	3
c	b	c	2
c	b	c	3

1.2. **Примеры задач удовлетворения ограничений.** В числе важнейших примеров задач УО в обзоре [17] указаны задача о раскраске графа, задача составления расписаний и задача SAT.

Задача о раскраске вершин графа и задача составления расписаний

Задача о раскраске карты или эквивалентная ей задача о раскраске вершин графа (рис. 1) так, чтобы никакие две смежные вершины не были окрашены в одинаковый цвет, является одной из хорошо известных задач, к которой сводится задача составления расписания.

Определение 3. [16]. *Устойчивым множеством* вершин графа $G = (V, E)$ называется подмножество вершин X, никакие две из которых не являются смежными.

Определение 4. [16]. *Правильной раскраской* графа $G = (V, E)$ называется разбиение множества вершин V: $V = X_1 + \ldots + X_c$, такое, что каждое X_i является устойчивым множеством.

В этом случае говорят, что элементы подмножества X_i "окрашены" цветом i, причем смежные вершины окрашиваются разными цветами.

Пример 2. *Задача составления расписания*

Дано множество $V = \{v_i\}$ учебных курсов в университете. Известны интервалы времени T_i, в течение которых читаются соответствующие курсы $\{v_i\}$. Необходимо приписать учебные курсы аудиториям так, чтобы никакие два курса не пересекались по времени для одной и той же аудитории. Эта задача может быть сведена к поиску правильной раскраски графа $G = (V, E)$, где $(v_i, v_j) \in E \Leftrightarrow T_i \bigcap T_j \neq \emptyset$. Здесь каждой аудитории соответствует свой цвет.

Задача SAT (satisfiability) (задача проверки выполнимости формулы логики высказываний) имеет важное прикладное значение, причем приложения находятся в области тестирования электронных схем, проектирования компьютеров, анализа изображений [17]. Именно с задачи SAT Cook в 1971 г. начал исследование задач

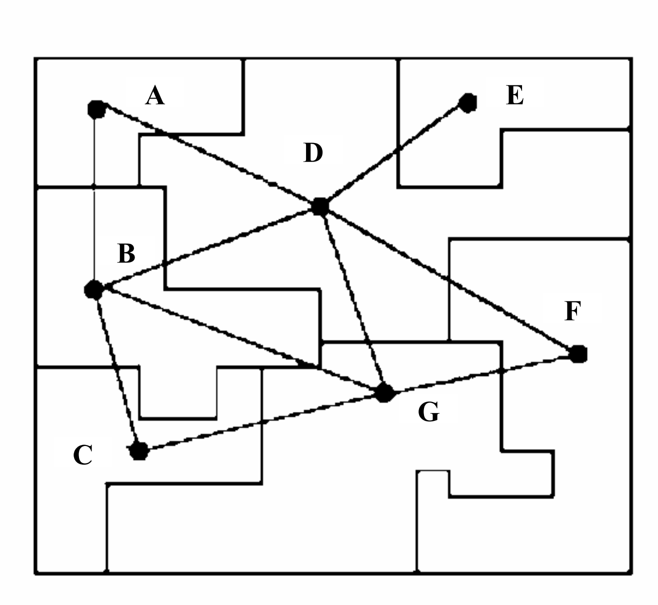

Рис. 1. Раскраска карты и соответствующий граф [14].

на NP-полноту [12]. Задача SAT состоит в определении, истинна ли данная формула логики высказываний при каком-нибудь значении литералов. Под решением задачи SAT понимается *интерпретация*, т.е. такое присваивание истинностных значений (1 или 0) литералам в формуле, при которых эта формула станет истинной.

2. Локальные элиминационные алгоритмы вычисления информации

При изучении сложных объектов не всегда возможно получить (или вычислить) полную информацию об объекте в целом, поэтому представляет интерес получение информации об объекте, рассматривая его по частям, т.е. локально. В [1] описаны предложенные Ю.И. Журавлевым локальные алгоритмы вычисления информации. Локальный алгоритм (ЛА) изучает элементы в порядке, задаваемом алгоритмом упорядочения A_π, используя при этом локальную информацию об элементах из окрестности [2] данного элемента. Алгоритм, расставляющий отметки, производит вычисление функции ϕ, значение которой на каждом шаге алгоритма будет определять вид отметки, выставляемой на этом шаге. Функция ϕ, порождающая алгоритм, – это функция от двух аргументов, первый из которых пробегает множество элементов, а второй – множество окрестностей. ЛА декомпозиции [6] задач ДО имеют свою специфику, состоящую в том, что они не вычисляют предикаты, а, используя принцип оптимальности Беллмана, вычисляют оптимальные частичные решения подзадач, соответствующих блокам задачи ДО.

Важной особенностью локальных алгоритмов является вычисление и использование именно *локальной информации* (т.е. информации об элементах окрестности элемента) при решении задач, поэтому локальные элиминационные алгоритмы (ЛЭА) вычисления информации позволяют осуществлять вычисление глобальной информации с помощью локальных вычислений.

Структура ЗУО задается ограничениями, и может быть задана как системой окрестностей переменных задачи (графом взаимосвязей переменных) и порядком просмотра этих переменных с помощью ЛЭА [5], так и различными производными структурами – блочными [4, 11], блочно-древовидными [7], задаваемыми структурными конденсированными графами. В конденсированном графе вершинами являются подмножества переменных задачи.

ЛЭА вычисляет информацию о локальных элементах структуры ЗУО, задаваемой структурным графом, записывая локальную информацию об этих элементах в виде новых зависимостей, добавляемых к задаче, затем элиминируя просмотренные элементы и использованные ограничения. Алгоритмическая схема ЛЭА представляет собой бесконтурный орграф (directed acyclic graph (DAG)), вершинами которого являются локальные подзадачи, соответствующие окрестностям элементов, а дуги – выражают информационную зависимость подзадач друг от друга.

Процедура ЛЭА состоит из двух частей:

- *прямая часть* — выделение локальных элементов, их окрестностей в текущем структурном графе и соответствующих им подзадач, вычисление и запоминание локальной информации в виде новых ограничений, добавляемых к задаче, элиминация просмотренных локальных элементов и использованных ограничений, получение значения критерия (совместна ли ЗУО);

- *обратная часть* — нахождение глобального решения исходной ЗУО по имеющимся таблицам с локальными решениями, обеспечивающего достижение критерия (совместности ЗУО).

Прямая часть ЛЭА анализирует окрестность $Nb(x)$ текущего элемента x в структурном графе задачи, применяет оператор элиминации, состоящий в решении подзадачи ЗУО, соответствующей окрестности $Nb(x)$ этого элемента в текущем структурном графе, к этому элементу, вычисляя локальную информацию об элементе x в виде нового ограничения $(R,\ x\bigcup Nb(x))$, содержащего локальные решения в виде допустимых наборов переменных вида $R(x,\ Nb(x))$. Потом строится проекция $R^{'} = \Pi_{Nb(x)}R(x,\ Nb(x))$; ограничение $(R^{'},\ Nb(x))$ добавляется к системе ограничений. Далее элемент x элиминируется вместе с использованными ограничениями, а из элементов его окрестности $Nb(x)$ создается клика в структурном графе (эта клика соответствует ограничению $(R^{'},\ Nb(x))$). Создание клик изменяет структурный граф и окрестности элементов. Обратная часть ЛАЭ восстанавливает решение всей задачи ЗУО на основе сохраненных таблиц с локальными решениями $R(x,\ Nb(x))$.

Для решения ЗУО, описываемой переменными $x_1, \ldots, x_n$, системой ограничений $(R_1, S_1), \ldots, (R_m, S_m)$, где R_i– отношение, $S_i = scope(R_i),\ i = 1, \ldots, m$, и при заданном упорядочении A_π переменных ЛЭА выглядит следующим образом:

1. Выбрать очередной элемент x (переменную или группу переменных) согласно упорядочению A_π. Сформулировать подзадачу задачи УО, соответствующую окрестности $Nb(x)$ элемента x в текущем структурном графе, сформировав новое ограничение R, $x \bigcup Nb(x)$ с диапазоном $(x,\ Nb(x))$,решить ее, запоминая в таблице все решения этого ограничения.

2. Спроектировать полученное ограничение на множество элементов подзадачи, соответствующих окрестности $Nb(x)$ элемента x. В результате получится новое ограничение, добавляемое к ограничениям ЗУО. Если ограничение с тем же набором переменных уже имеется, найти их пересечение. Если пересечение пусто, то ЗУО несовместна и допустимых решений не имеет.

3. Элиминировать элемент x вместе с соответствующими ограничениями.

4. Продолжать до тех пор, пока не останется нерешенных ограничений.

Рассмотрим подробнее детали реализации ЛЭА при решении задач УО в случае, когда структурный граф является *графом взаимосвязей* переменных [11], который в литературе называется также *графом ограничений* [14]. В графе взаимосвязей ЗУО вершины соответствуют переменным ЗУО, причем две вершины соединяются ребром, если соответствующие переменные имеются в одном и том же ограничении (т.е., в одном и том же диапазоне какого-то отношения). Рассмотрим переменную x_i и ее окрестности $U(x_i)$, $Nb(x_i)$ в текущем графе взаимосвязей G: $U(x_i) = \{R_p \mid x_i \in \mathit{scope}(R_p)\}$, $Nb(x_i) = \{x_j \mid \exists p:\ x_i,\ x_j \in \mathit{scope}(R_p)\}$. Пусть $R_{i_1}, \dots, R_{i_{m_i}}$ – отношения с диапазонами $S_{i1}, \dots, S_{im_i}$, индексы которых входят в $U(x_i)$, причем их диапазоны содержат x_i.

Решение ЗУО, заданной ограничениями $R_{i1}, \dots, R_{im_i}$ и соответствующими переменными, и последующая элиминация x_i может быть описана следующим образом. Определяем диапазон нового ограничения как $S^{(i)} = \bigcup_{r=1}^{m_i} S_{i_r} - \{x_i\}$ и новое отношение $R^{(i)} = \Pi_{S^{(i)}}(\bowtie_{r=1}^{m_i} R_{i_r})$. Затем находится пересечение с существовавшим ранее отношением с тем же диапазоном $S^{(i)}$: $R^{(i)} = R_{S^{(i)}} \bigcap R^{(i)}$. Граф взаимосвязей $G = (V, E)$ при элиминации переменной x_i изменяется согласно алгоритму элиминационной игры [21]:

$$V \leftarrow V - \{x_i\};$$
$$E \leftarrow E \bigcup \{(x_k,\ x_r) |\ x_k,\ x_r \in Nb(x_i)\}.$$

Пример 3. Рассмотрим решение с помощью ЛЭА задачи УО, содержащей ограничения $(R_{345},\ S_{345})$, $(R_{25},\ S_{25})$, $(R_{42},\ S_{42})$, $(R_{31},\ S_{31})$, $(R_{21},\ S_{21})$, где диапазоны отношений: $S_{345} = \{x_3,\ x_4,\ x_5\}$, $S_{25} = \{x_2,\ x_5\}$, $S_{42} = \{x_4,\ x_2\}$, $S_{31} = \{x_3,\ x_1\}$, $S_{21} = \{x_2,\ x_1\}$, а отношения заданы в табличной форме:

R_{345} :

x_3	x_4	x_5
1	1	1
0	0	1
0	1	1

R_{25} :

x_2	x_5
0	1
1	0

R_{42} :

x_4	x_2
1	0
0	1

R_{31} :

x_3	x_1
0	1
1	0

R_{21} :

x_2	x_1
1	1
0	0

В качестве структурного графа ЗУО используем граф взаимосвязей переменных. Для данного примера граф взаимосвязей переменных показан на рис. 2 а). Для определения порядка A_π элиминации переменных используем эвристику "Minimum Degree" [8], которая выбирает на каждом шаге вершину текущего графа с минимальной степенью. Первой выберем переменную x_1. Ее соседи – x_2, x_3, $Nb(x_1) = \{x_2, x_3\}$. Рассмотрим ЗУО, задаваемую ограничениями (R_{31}, S_{31}), (R_{21}, S_{21}), где $S_{31} = \{x_3, x_1\}$, $S_{21} = \{x_2, x_1\}$. Построим новое ограничение $R'_{123} = R_{31} \bowtie R_{21}$, используя операцию $\bowtie$ соединения отношений (табл. 2).

Таблица 2. Отношение R'_{123}

	x_1	x_2	x_3
R'_{123}:	1	1	0
	0	0	1

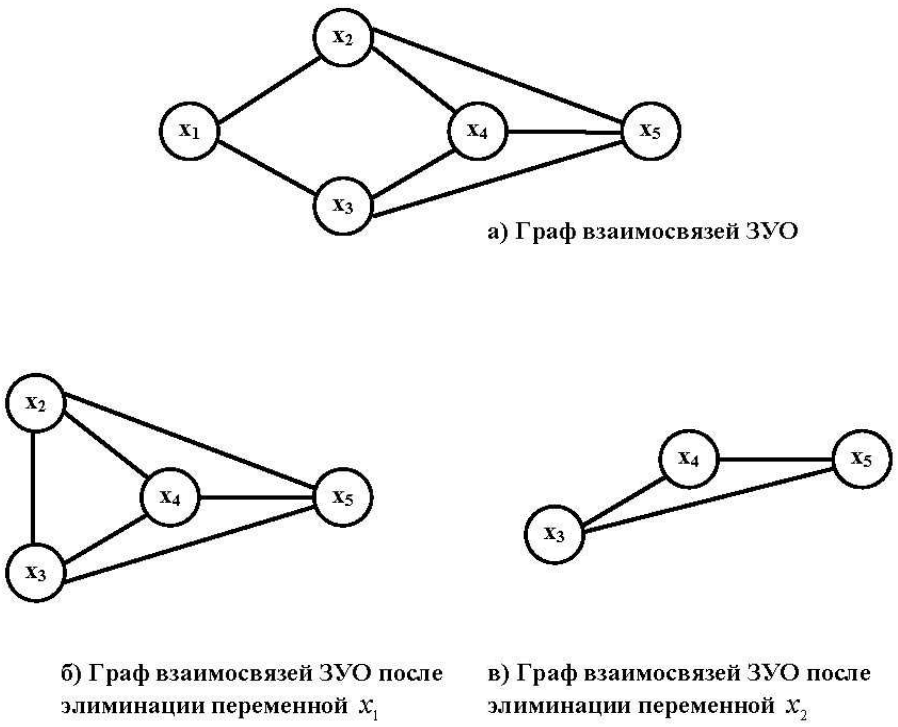

Рис. 2. Граф взаимосвязей для ЗУО.

Найдем проекцию $R'_{23} = \Pi_{\{x_2,x_3\}} R'_{123}$

	x_2	x_3
R'_{23}:	1	0
	0	1

В результате вершина x_1 элиминируется вместе с ребрами, исходящими из нее (эти ребра соответствуют уже использованным отношениям $R_{31},\ R_{21}$), вершины $x_2,\ x_3 \in Nb(x_1)$ соединяются ребром, соответствующим новому отношению R'_{23} (рис. 2 б)).

Рассмотрим переменную x_2, окрестность которой $Nb(x_2) = \{x_3,\ x_4,\ x_5\}$. Рассмотрим ЗУО, задаваемую ограничениями $(R_{25},\ S_{25})$, $(R_{42},\ S_{42})$, $(R'_{23},\ S_{23})$, где $S_{25} = \{x_2,\ x_5\}$, $S_{42} = \{x_4,\ x_2\}$, $S_{23} = \{x_2,\ x_3\}$. Построим новое ограничение $R'_{2345} = R_{25} \bowtie R_{42} \bowtie R_{23}$ (табл. 3). Найдем проекцию $R'_{345} = \Pi_{\{x_3,x_4,x_5\}} R'_{2345}$ от-

Таблица 3. Отношение R'_{2345}

x_2	x_3	x_4	x_5
0	1	1	1
1	0	0	0

ношения R'_{2345} на $\{x_3, x_4, x_5\}$ (табл. 4). Вершина x_2 элиминируется вместе с ребра-

Таблица 4. Отношение R'_{345}

x_3	x_4	x_5
1	1	1
0	0	0

ми, исходящими из нее (и соответствующими отношениям $R_{25},\ R_{42},\ R'_{23}$), вершины $x_3,\ x_4,\ x_5 \in Nb(x_2)$ соединены между собой ребрами, соответствующими отношению R_{345}, поэтому соединять их не нужно (рис. 2 в)).

Рассмотрим отношения R_{345} и R'_{345} и найдем их пересечение: $R^{+}_{345} = R_{345} \bigcap R'_{345}$ (табл. 5). Прямая часть ЛАЭ закончена, так как оставшиеся переменные ЗУО най-

Таблица 5. Отношение R^{+}_{345}

x_3	x_4	x_5
1	1	1

дены: $x_3 = 1,\ x_4 = 1,\ x_5 = 1$. Для нахождения остальных переменных применим обратную часть процедуры ЛАЭ, просматривая таблицы с частичными решениями. Из таблицы 3, зная $x_3 = 1,\ x_4 = 1,\ x_5 = 1$, найдем $x_2 = 0$. Из таблицы 2, зная $x_2 = 0,\ x_3 = 1$, найдем $x_1 = 0$. Таким образом, решение исходной ЗУО:

$$x_1 = 0,\ x_2 = 0,\ x_3 = 1,\ x_4 = 1,\ x_5 = 1.$$

2.1. **Задача SAT и ее графовое представление.** Рассмотрим пример решения задачи SAT [19], заданной в конъюнктивной нормальной форме, состоящей из 7 элементарных дизъюнкций $C_1,\ C_2,\ C_3,\ C_4,\ C_5,\ C_6,\ C_7$, называемых дизъюнктами:

$$(x_2 \vee x_6) \wedge (x_5 \vee \neg x_6) \wedge (\neg x_1 \vee x_5) \wedge (\neg x_2 \vee \neg x_5) \wedge (x_1 \vee x_3 \vee x_4) \wedge (x_2 \vee \neg x_4) \wedge (x_1 \vee x_2)$$

Структура формулы может быть задана графом взаимосвязей – неориентированным графом, вершины которого соответствуют переменным-литералам, причем ребро соединяет две вершины, если соответствующие переменные входят в один и тот же дизъюнкт формулы. Оператор элиминации в данном случае — *резолюция*, которая по двум дизъюнктам $(\alpha \vee Q)$ и $(\beta \vee \neg Q)$ выводит дизъюнкт $(\alpha \vee \beta)$, называемый *резольвентой*, в которой литерал Q элиминирован. Оператор элиминации (в данном случае – резолюция) порождает новые дизъюнкты, которым соответствуют новые ребра в графе взаимосвязей. Определим порядок рассмотрения окрестностей [1] на основе применения эвристики "Minimal Degree" [8]: $A_\pi = \{6, 5, 2, 4, 3, 1\}$. Рассмотрим вершину 6 и соответствующую переменную x_6. Ее окрестности $U(x_6) = \{C_1, C_2\}$, $Nb(x_6) = \{x_2, x_5\}$. Выпишем дизъюнкты из окрестности $U(x_6) : (x_2 \vee x_6) \wedge (x_5 \vee \neg x_6)$. Применяя оператор элиминации – резольвенту, получим новый дизъюнкт $C_{12} : (x_2 \vee x_5)$, т.е. новое отношение с переменными из $Nb(x_6)$. Для обеспечения возможности последующего выполнения обратной части ЛЭА необходимо запомнить вычисленную информацию о локальном решении $h_6(x_2, x_5) = x_2 \vee x_5$ и $x_6^*(x_2, x_5)$ – значение переменной x_6 для заданных значений x_2, x_5, при котором оба дизъюнкта будут истинны.

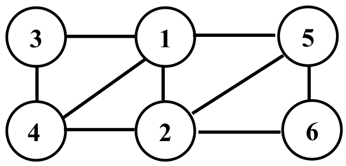

Рис. 3. Граф взаимосвязей для задачи SAT.

Таблица 6

(x_2, x_5)	$h_6(x_2, x_5)$	$x_6^*(x_2, x_5)$
(0, 0)	0	–
(0, 1)	1	1
(1, 0)	1	0
(1, 1)	1	1, 0

Элимируем дизъюнкты $C_1,\ C_2$ и переменную x_6. Соседи x_6 – переменные x_2, x_5 – связаны теперь в новом дизъюнкте $C_{12} : (x_2 \vee x_5)$, который добавляется в систему дизъюнктов. Граф взаимосвязей изменяется следующим образом: вершина 6 удаляется вместе с ребрами, исходящими из нее; соседи вершины 6 – вершины 2 и 5, соответствующие переменным x_2, x_5 должны быть соединены ребром в графе взаимосвязей (что соответствует появлению нового дизъюнкта C_{12}). Поскольку такое ребро было, то его добавлять не надо (рис. 4). Согласно порядку рассмот-

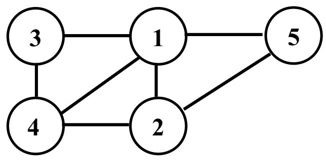

Рис. 4. Граф взаимосвязей для задачи SAT после элиминации вершины 6.

рения окрестностей A_π, перейдем к переменной x_5 (вершине 5) и ее окрестностям $U(x_6) = \{C_{12}, C_3, C_4\}$, $Nb(x_5) = \{x_1, x_2\}$.

$$(x_2 \vee x_5) \wedge (\neg x_1 \vee x_5) \wedge (\neg x_2 \vee \neg x_5).$$

Применяя оператор элиминации – резольвенту к последним двум дизъюнктам, получим новый дизъюнкт $C_{34} : \neg x_1 \vee \neg x_2$. Резольвента первого и третьего дизъюнктов дает $\neg x_2 \vee \neg x_2$, что можно отбросить. Элимируем дизъюнкты C_{12}, C_3, C_4 и переменную x_5. Соседи x_5 – переменные x_1, x_2 – связаны теперь в новом дизъюнкте $C_{34} : \neg x_1 \vee \neg x_2$, который добавляется в систему дизъюнктов. Граф взаимосвязей изменяется следующим образом: вершина 5 удаляется вместе с ребрами, исходящими из нее; поскольку вершины 1 и 2 соединены ребром, то новое ребро между ними не добавляется.

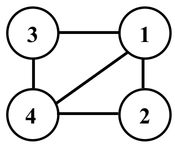

Рис. 5. Граф взаимосвязей для задачи SAT после элиминации вершины 5.

Вычислим и запомним информацию о локальном решении $h_5(x_1, x_2) = \neg x_1 \vee \neg x_2$ и $x_5^*(x_1, x_2)$ – значение переменной x_5 для заданных значений x_1, x_2, при котором оба дизъюнкта будут истинны.

Рассмотрим переменную x_2 и ее окрестности $U(x_2) = \{C_{34}, C_6, C_7\}$, $Nb(x_2) = \{x_1, x_4\}$.

$$(\neg x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_4) \wedge (x_1 \vee x_2).$$

Применяя оператор элиминации – резольвенту к первым двум дизъюнктам, вычислим новый дизъюнкт $\neg x_1 \vee x_4$. Резольвента первого и третьего дизъюнктов дает $\neg x_1 \vee \neg x_1$, что можно отбросить.

Таблица 7

(x_1, x_2)	$h_5(x_1, x_2)$	$x_5^*(x_1, x_2)$
(0, 0)	1	1
(0, 1)	1	0
(1, 0)	1	1
(1, 1)	0	–

Элиминируем дизъюнкты C_{34}, C_6, C_7 и переменную x_2. Соседи x_2 – переменные $x_1,\ x_4$ связаны теперь в новом дизъюнкте $C_{14} = \neg x_1 \vee \neg x_4$, который добавляется в систему дизъюнктов. Граф взаимосвязей изменяется следующим образом: вершина 2 удаляется вместе с ребрами, исходящими из нее; поскольку 1 и 4 соединены ребром, то новое ребро не добавляется. Вычислим и запомним информацию о локальном

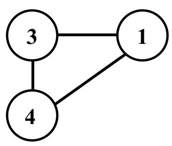

Рис. 6. Граф взаимосвязей для задачи SAT после элиминации вершины 2.

решении $h_2(x_1, x_4) = \neg x_1 \vee \neg x_4$ и $x_2^*(x_1,\ x_4)$ – значение переменной x_2 для заданных значений $x_1,\ x_4$, при котором все дизъюнкты будут истинны.

Таблица 8

(x_1, x_4)	$h_2(x_1, x_4)$	$x_2^*(x_1, x_4)$
(0, 0)	1	1
(0, 1)	1	1
(1, 0)	1	0
(1, 1)	0	–

Рассмотрим переменную x_4 и ее окрестности $U(x_4) = \{C_{14}, C_5\},\ Nb(x_4) = \{x_1, x_3\}$.

$$(\neg x_1 \vee \neg x_4) \wedge (x_1 \vee x_3 \vee x_4)$$

Применяя оператор элиминации – резольвенту к этим дизъюнктам, получим новый дизъюнкт $h_4(x_1, x_3) \equiv 1$ при любых x_1, x_3. Элимируем дизъюнкты $C_{14},\ C_5$ и переменную x_4. Соседи x_4 – переменные x_1, x_3 принимают произвольные бинарные значения, причем данная формула логики высказываний истинна. Вычислим и запомним информацию о локальном решении: $x_4^*(x_1,\ x_3)$ - значение переменной x_4 для заданных значений $(x_1,\ x_3)$, при котором все дизъюнкты будут истинны. Найдем теперь

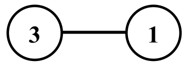

Рис. 7. Граф взаимосвязей для задачи SAT после элиминации вершины 4.

Таблица 9

(x_1, x_3)	$h_4(x_1, x_3)$	$x_4^*(x_1, x_3)$
(0, 0)	1	1
(0, 1)	1	1,0
(1, 0)	1	0
(1, 1)	1	0

решения, т.е. те значения переменных x_1, x_3, при которых данная формула алгебры высказываний истинна. Поскольку значения произвольны, построим следующую таблицу возможных решений, используя таблицы 6–9 в обратном порядке.

Таблица 10. Решения задачи SAT.

x_1	x_2	x_3	x_4	x_5	x_6
0	1	0	1	0	0
0	1	1	1	0	0
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	0	1	1

2.2. **Задача о раскраске графа.** В задаче о раскраске вершин графа граф взаимосвязей ЗУО идентичен графу, вершины которого нужно раскрасить. При этом переменные ЗУО соответствуют вершинам графа, области значений переменных – множеству возможных цветов, а отношения являются неравенствами вида: $A \neq B$ (цвет вершины A отличен от цвета вершины B).

Пример 4. Рассмотрим пример задачи о раскраске графа, показанного на рис. 8 а), вершины которого нужно раскрасить в два цвета: красный (1) и зеленый (2) [14]. При формулировании ЗУО, в качестве отношения R_{ED} принимается следующее: $R_{ED} : E \neq D$, т.е. вершины E и D окрашены в разные цвета, или $R_{ED} = \{(1,2),(2,1)\}$; в табличной форме

$$R_{ED} : \begin{array}{c|c} E & D \\ \hline 1 & 2 \\ 2 & 1 \end{array}$$

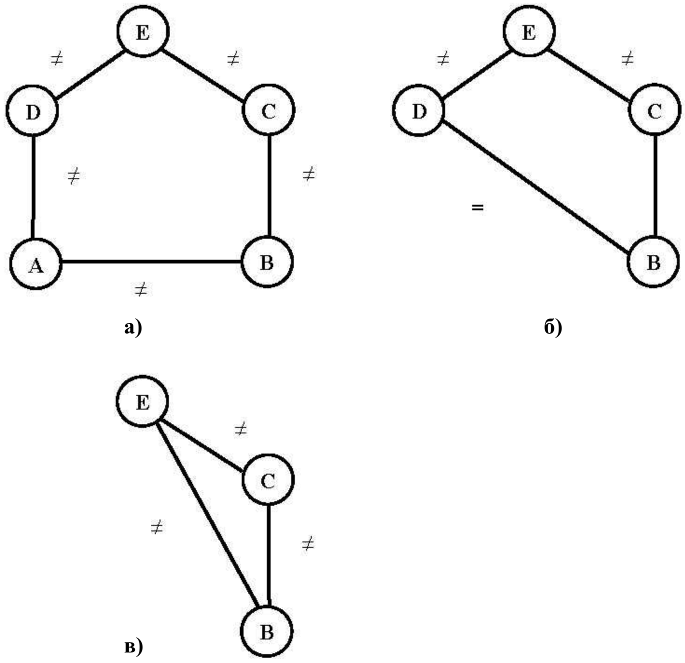

Рис. 8. Граф взаимосвязей для задачи о раскраске: а) исходный; б) после элиминации вершины A; в) после элиминации вершины D.

Используем порядок элиминации $A_\pi = \{A,\ D,\ C,\ B,\ E\}$. Рассмотрим вершину A графа на рис. 8 а), окрестность которой $Nb(A) = \{B,\ D\}$. Ребрам $(A,\ B)$ и $(A,\ D)$ соответствуют отношения R_{AB} и R_{AD}: $R_{AB} = \{(1,2),(2,1)\}$, $R_{AD} = \{(1,2),(2,1)\}$.

Находя $R_{ABD} = R_{AB} \bowtie R_{AD}$, получим $R_{ABD} = \{(1,2,2),(2,1,1)\}$. Проектируя отношение R_{ABD} на $Nb(A) = \{B,\ D\}$, получим $R_{BD} = \{(2,2),(1,1)\}$, т.е. $B = D$ (цвета вершин B и D одинаковы) или в табличной форме

$$R_{BD} : \begin{array}{c|c} B & D \\ \hline 2 & 2 \\ 1 & 1 \end{array}$$

Элиминируем вершину A вместе с инцидентными ей ребрами $(A,\ B)$ и $(A,\ D)$ (соответствующие этим ребрам отношения R_{AB} и R_{AD} также далее не рассматриваются). Добавим новое отношение $R_{BD} : B = D$. Измененный граф взаимосвязей показан на рис. 8 б). В нем вершины B и D соединены ребром, соответствующим отношению $R_{BD} : B = D$.

Рассмотрим следующую, согласно порядку A_π, вершину D. $Nb(D) = \{B,\ E\}$, поэтому рассмотрим ребра $(D,\ B)$ и $(D,\ E)$ графа на рис. 8 б) и соответствующие им отношения: $R_{BD} : B = D$ и $R_{DE} : D \neq E$. Найдем $R_{DBE} = R_{BD} \bowtie R_{DE}$ в виде $R_{DBE} = \{(1,1,2),(2,2,1)\}$. Найдем проекцию $R_{BE} = \Pi_{\{B,E\}} R_{DBE}$:

$R_{BE} = \{(1,2),(2,1)\}$ или $B \neq E$. Элиминируем вершину D вместе с инцидентными ей ребрами (D, B) и (D, E) (соответствующие отношения R_{BD} и R_{DE} далее не рассматриваются). Добавим новое отношение $R_{BE} : B \neq E$. Измененный граф взаимосвязей показан на рис. 8 в). В нем вершины B и E соединены ребром, соответствующим добавленному отношению $R_{BE} : B \neq E$.

Рассмотрим следующую, согласно порядку A_π, вершину C. $Nb(C) = \{B,\ E\}$, так что рассмотрим ребра (C, B) и (C, E) графа на рис. 8 в) и соответствующие им отношения: $R_{CB} : C \neq B$ и $R_{CE} : C \neq E$. Найдем $R_{CBE} = R_{CB} \bowtie R_{CE} : R_{CBE} = \{(1,2,2),(2,1,1)\}$. Найдем проекцию $R'_{BE} = \Pi_{\{B,E\}} R_{CBE}$: $R'_{BE} = \{(2,2),(1,1)\}$ или $B = E$. Поскольку уже имеется отношение $R_{BE} : B \neq E$ с тем же диапазоном $\{B, E\}$, необходимо найти пересечение $R_{BE} \bigcap R'_{BE}$, которое пусто, что означает несовместность ЗУО, т.е. данная задача о раскраске вершин графа с помощью двух красок не имеет решения.

Заключение

Локальный элиминационный алгоритм вычисления информации — перспективный подход, делающий возможным решение прикладных разреженных задач удовлетворения ограничений, в том числе задач SAT и задач о раскраске графа. *Перспективными направлениями* дальнейших исследований являются разработка эффективных схем локального элиминационног алгоритма при решении конкретных задач удовлетворения ограничений, обладающих специальной структурой с использованием различных видов структурных графов.

Список литературы

1. *Журавлев Ю.И.* Избранные научные труды. – М.: Магистр, 1998. – 420 с.
2. *Журавлев Ю.И., Лосев Г.Ф.* Окрестности в задачах дискретной математики // Кибернетика и системный анализ. — 1995. – №2. — С. 32-41.
3. *Сараев А.Д., Щербина О.А.* Системный анализ и современные информационные технологии // Труды Крымской академии наук. – Симферополь: СОНАТ. – 2006. – С. 47-59.
4. *Щербина О.А.* Элиминационные алгоритмы декомпозиции задач дискретной оптимизации // Таврический вестник информатики и математики. – 2006. – №2. – С. 28-41.
5. *Щербина О.А.* О несериальной модификации локального алгоритма декомпозиции задач дискретной оптимизации // Динамические системы. — 2005. — Вып.19. — С.179—190.
6. *Щербина О.А.* О локальных алгоритмах решения задач дискретной оптимизации // Проблемы кибернетики. – 1983. – № 40. – P.171-200.
7. *Щербина О.А.* Локальные алгоритмы для блочно-древовидных задач дискретного программирования // Журн. вычисл. математики и матем. физики. – 1985. – Т. 25. – N 8. – С. 1143-1154.
8. *Amestoy P.R., Davis T.A., Duff I.S.* An approximate minimum degree ordering algorithm // SIAM Journal on Matrix Analysis and Applications. — 1996. — V. 17, N.4. — P.886—905.
9. *Arnborg S., Corneil D.G., Proskurowski A.* Complexity of finding embeddings in a k-tree // SIAM J. Alg. Disc. Meth. — 1987. – 8. – P.277–284.
10. *Barnier N., Brisset P.* Graph coloring for air traffic flow management // Annals of Operations Research. – 2004. – V.130. – P.163-178.
11. *Bertele U., Brioschi F.* Nonserial Dynamic Programming. – New York: Academic Press, 1972. – 235 p.

12. *Cook S.A.* The complexity of theorem-proving procedures // In: Proc. 3rd Ann. ACM Symp. on Theory of Computing Machinery. – New York. – 1971. – P.151-158.
13. *Courcelle B.* Graph rewriting: An algebraic and logic approach // In J. Van Leeuwen (ed.). Handbook of Theoretical Computer Science, Volume B. — Elsevier. — 1990. – P.193-242.
14. *Dechter R.* Constraint processing. – Morgan Kaufmann, 2003. – 481 p.
15. *Dechter R.* Bucket elimination: A unifying framework for reasoning // Articial Intelligence. — 1999. — V. 113. — P.41-85.
16. *Golumbic M.C.* Algorithmic Graph Theory and Perfect Graphs. – New York: Academic Press, 1980. – 284 p.
17. *Gu J., Purdom P.W., Franco J., Wah B.W.* Algorithms for the satisfiability (SAT) problem: A survey // In: Satisfiability Problem Theory and Applications. – 1997. – P. 19-153.
18. *Hicks I.V., Koster A.M.C.A., Kolotoglu E.* Branch and tree decomposition techniques for discrete optimization // In: Tutorials in Operations Research. – New Orleans: INFORMS – 2005. – P.1—29. (http://ie.tamu.edu/People/faculty/Hicks/bwtw .pdf)
19. *Hooker J.* Logic-based Methods for Optimization: Combining Optimization and Constraint Satisfaction. – John Wiley, 2000. – 495 p.
20. *Neumaier A., Shcherbina O.* Nonserial dynamic programming and local decomposition algorithms in discrete programming (submitted). Available online: http://www.optimization-online.org/DB_HTML/2006/03/1351.html
21. *Parter S.* The use of linear graphs in Gauss elimination // SIAM Review. – 1961. – V. 3. – P.119–130.
22. *Shenoy P.P.* Axioms for dynamic programming // In: A. Gammerman (ed.). Computational Learning and Probabilistic Reasoning. John Wiley & Sons, Ltd. – 1996. — P. 259–275.
23. *Tsang E.* Foundations of Constraint Satisfaction. – New York: Academic Press, 1993. – 421 p.

Статья поступила в редакцию 17.05.2007