Несогласованные компактные структуры данных в задачах комбинаторной оптимизации

В. Ф. Романов

Аннотация: В сфере исследований эффективности алгоритмов дискретной оптимизации значительное место занимает метод полиномиального сведения одних задач к другим с использованием компонент специального назначения. В статье излагается нетрадиционный метод компактного представления множеств двоичных последовательностей в виде «структур компактных троек» и «структур компактных пар», допускающих как логическую, так и арифметическую интерпретацию данных. Ключевым пунктом исследования является адаптация метода распространения ограничений, используемого в оптимизационных задачах, к эффективному разрешению булевой формулы, закодированной средствами несогласованных компактных структур.

Ключевые слова: *структура компактных троек, структура компактных пар, несогласованные структуры, унификация структур, совместный выполняющий набор.*

Введение

В статье излагается неортодоксальный метод компактного представления множеств двоичных последовательностей в виде *структур компактных троек* (СКТ) и *структур компактных пар* (СКП), допускающих как логическую, так и арифметическую интерпретацию данных. Подходящей иллюстрацией применения СКТ и СКП является уникальная комбинаторная модель для классической трудноразрешимой задачи 3-ВЫПОЛНИМОСТЬ (3-SAT) [1–3]. Синтезирована формула, по своему строению являющаяся вариацией 2-КНФ, в соответствии с которой построена процедура вычислений, представляющая собой адаптацию полиномиального алгоритма распространения ограничений (в частности, алгоритма *ограниченного возврата*), к эффективному разрешению булевой формулы, закодированной средствами несогласованных СКП.

1. Постановка задачи. Табличные формулы

Структуры компактных троек, рассматриваемые здесь в качестве основных объектов исследования, порождены одной из возможных интерпретаций задачи 3-SAT. Исходная формулировка задачи: для данных m элементарных дизъюнкций $C_1, \ldots, C_m$, каждая из которых содержит в точности три литерала, относящихся к булевым переменным $x_1, \ldots, x_n$, определить, выполнима ли формула $F = C_1 \wedge C_2 \wedge \ldots \wedge C_m$.

Для формулы, представленной в КНФ, введем условную запись в виде таблицы (табличной формулы), содержащей n столбцов, отмеченных именами переменных, и m строк, каждая из которых представляет терм C_i последовательностью из нулей и единиц: 0 в j-м столбце и i-й строке обозначает появление переменной x_j в терме C_i без инверсии, 1 — со знаком инверсии. Так, формуле

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

соответствует табличная формула:

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

Очевидно, что если 0 обозначает *ложь*, а 1 — *истину* (независимо от обозначений, принятых для табличной формулы), то $F = 0$ на тех и только тех наборах значений переменных, где в качестве подмножества присутствует хотя бы одна строка из табличной формулы.

2. Структуры компактных троек

Рассмотрим сначала формулу, термы которой представляют собой *компактные тройки*, т. е. тройки литералов l_j, l_{j+1}, l_{j+2} с номерами, расположенными подряд; $l_j \in \{x_j, \neg x_j\}$, $1 \le j \le (n-2)$. Такую формулу назовем *формулой компактных троек* (ФКТ). Идея разрешения ФКТ, заданной таблицей для F, заключается в ее преобразовании (полиномиальном по сложности) в *структуру компактных троек* (СКТ). Элементами СКТ являются строки — компактные тройки значений переменных, расположенные на $n-2$ ярусах. Ярусы включают соответственно переменные с номерами 1, 2, 3; 2, 3, 4; …; $n-2$, $n-1$, n. При построении СКТ каждый ярус заполняется только теми двоичными наборами трех переменных, которые отсутствуют в ФКТ. Далее выполняется *очищение* структуры — удаление из СКТ строк, не образующих конкатенации (сочленения по совпадающим значениям двух переменных) хотя бы с одной строкой каждого соседнего яруса.

Если хотя бы один ярус СКТ оказывается пустым, вся структура объявляется пустым множеством, а формула F — невыполнимой. СКТ, содержащая $n-2$ яруса, может быть сформирована тогда и только тогда, когда формула F выполнима. Действительно, из 2^n теоретически возможных двоичных наборов, образованных конкатенациями строк при полностью заполненных ярусах (по восемь строк на каждом ярусе), при формировании СКТ удаляются все те и только те, которые содержат в качестве подмножества значений переменных хотя бы одну строку таблицы, представляющей формулу F. Таким образом, в СКТ содержатся в виде конкатенаций троек все наборы, выполняющие ФКТ. *Итак, сам факт существования СКТ, содержащей $n-2$ непустых яруса, означает выполнимость формулы компактных троек.*

Например, для ФКТ F_1 преобразование приводит к СКТ Z; в промежуточной структуре Q зачеркнуты строки, удаленные согласно процедуре очищения. В рассмотренном примере выявляются два выполняющих набора: 01101 и 10011.

F_1

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

Q

x_1	x_2	x_3	x_4	x_5
~~0~~	~~1~~	~~0~~		
0	1	1		
1	0	0		
~~1~~	~~1~~	~~0~~		
	0	0	1	
	~~0~~	~~1~~	~~0~~	
	~~0~~	~~1~~	~~1~~	
	1	1	0	
		~~0~~	~~0~~	~~0~~
		~~0~~	~~0~~	~~1~~
		0	1	1
		1	0	1
		~~1~~	~~1~~	~~0~~

Z

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

Подструктура S' СКТ S представляет собой СКТ, составленную из подмножества согласованных по ярусам строк S (обозначение операции: $S' \subseteq S$). Подчеркнем, что в подструктуре, как и в любой непустой СКТ, должны присутствовать $n-2$ яруса. *Элементарной* назовем СКТ, содержащей по одной строке на каждом из ярусов. Элементарной СКТ соответствует единственная конкатенация строк и, соответственно, один набор значений переменных.

Из сказанного следует, что по заданному множеству наборов значений переменных может быть синтезирована СКТ, содержащая эти наборы: для этого достаточно объединить по одноименным ярусам соответствующие элементарные СКТ.

Введем на множестве СКТ $S_1, S_2, \ldots, S_q$, построенных на основе *различных* перестановок переменных, многоместную операцию *унификации*, которая включает несколько правил совместного преобразования структур:

1. Если некоторая переменная имеет *константное значение*, т. е. $x_j \equiv 0$ или $x_j \equiv 1$ хотя бы в одной СКТ S_p ($1 \le j \le n$, $1 \le p \le q$), то все строки, содержащие инверсное значение указанной переменной, удаляются из всех унифицируемых СКТ.
2. Если две переменные x_j и x_r присутствуют вместе (в любом порядке) в компактных тройках двух или более унифицируемых СКТ, то сочетания значений этих переменных должны быть общими во всех таких СКТ. Все строки, не отвечающие этому требованию, удаляются из СКТ.
3. Удаление строк из СКТ согласно правилам 1–2 всегда сопровождается очищением соответствующих структур.

При унификации возникают варианты получения пустого множества структур в следующих случаях:

– некоторый ярус хотя бы в одной СКТ оказывается пустым;
– существует «конфликт» константных значений для переменной x_j хотя бы в двух различных СКТ, т. е. $x_j \equiv 0$ в одной из них и $x_j \equiv 1$ — в другой;
– хотя бы одна СКТ пуста изначально (тривиальный случай).

Итак, унифицированные СКТ могут быть пустыми либо непустыми только одновременно.

3. Декомпозиция формулы

Для решения задачи 3-SAT в общем случае используем декомпозицию формулы:

$$F = F_1 \wedge F_2 \wedge \ldots \wedge F_k,$$

при которой каждая формула F_r ($r = 1, \ldots, k;\ k \le m$) сопряжена с собственной перестановкой переменных $P_r = \langle x_{r_1}, x_{r_2}, \ldots, x_{r_n} \rangle$ и представлена в виде ФКТ. Декомпозиция может быть реализована различными алгоритмами полиномиальной сложности, сводящимися к перебору строк табличной формулы F и формированию подходящих перестановок таким образом, что группы строк, относящиеся к каждой перестановке, образуют компактные тройки. Нетрудно заключить, что значение k (число ФКТ) находится в пределах $\lceil w/(n-2) \rceil \le k \le m$, где w — число групп термов в F, содержащих одинаковые переменные. При «идеальных» исходных данных задачи $k = 1$; предельный случай $k = m$ связан с формированием отдельной перестановки переменных для каждого терма исходной формулы. Заметим, что минимизация числа ФКТ при декомпозиции не является принципиальной.

Сложность процедуры декомпозиции формулы, заданной таблицей описанного типа, можно оценить исходя из простого (но не тривиального) способа формирования множества перестановок переменных, каждая из которых сопряжена с отдельной ФКТ. Строки таблицы просматриваются по порядку, и, в зависимости от расположения тройки в очередной строке, выполняется одно из следующих действий:

- инициализация формирования новой перестановки с включением тройки в соответствующую ФКТ;
- продолжение формирования одной из незавершенных перестановок с включением тройки в соответствующую ФКТ;
- включение тройки в одну из ФКТ, для которых формирование перестановок завершено.

Таким образом, перестановки не перебираются, а конструируются, что исключает экспоненциальную сложность вычислений. Асимптотическая сложность алгоритма, основанного на приведенных пунктах и сводящегося к k-кратному просмотру строк матрицы размера m n, выражается оценкой $O(m\ n\ k)$.

ФКТ преобразуются в k структур компактных троек: $S_1, S_2, \ldots, S_k$, после чего решение задачи для формулы F сводится к вопросу: существует ли непустое пересечение множеств наборов, закодированных в СКТ, которые построены на основе несовпадающих перестановок переменных? Цель состоит в ответе на этот вопрос, т. е. установлении факта существования или отсутствия совместных выполняющих наборов (СВН) для k формул компактных троек, без генерирования самих наборов, имеющего следствием в общем случае экспоненциальную сложность вычислений.

В качестве примера для иллюстрации теоретических построений (не ограничивающего общность рассуждений) рассмотрим формулу F, представленную таблицей 1.

Оптимальная декомпозиция формулы, приведенной в таблице 1, базируется на трех перестановках переменных и является основой для формирования трех формул компактных троек F_1, F_2, F_3 (таблица 2) и структур компактных троек S_1, S_2, S_3 (таблица 3).

Отметим, что в ФКТ, в отличие от СКТ, сочленение ярусов по совпадающим значениям двух переменных не является обязательным.

Таблица 1. Исходная формула *F*

a	b	c	d	e	f	g	h
0	0	0					
0	1	0					
0	1	1					
1	0	0					
1	1	1					
0	1			0			
1				1	0		
1				0	1		
0		0			0		
0		0			1		
		1			0		
1		0			1		
1			0		0		
0			0		1		
1			0		1		

a	b	c	d	e	f	g	h
0		1					0
1		0					1
1		1					1
	1					0	0
	0					1	0
	0					0	1
	1					0	1
	1					1	1
	0			0		0	
	1			1		1	
	0			0		1	
	1			1			1
	0			0			0
	1			0		0	
		0	0	0			

a	b	c	d	e	f	g	h
		1	1	0			
		1	0		0		
		0	0		0		
		1		0			1
		1		0			0
			0	1	0		
			1	0	1		
				0	0	0	
				1	0	0	
				1	1	1	
					0	1	0
					1	0	1
		1	0	0			
0			1		1		

Таблица 2. Формулы компактных троек

ФКТ1 (F_1)

a	b	c	d	e	f	g	h
0	0	0					
0	1	0					
0	1	1					
1	0	0					
1	1	1					
		0	0	0			
		1	0	0			
		1	1	0			
			0	1	0		
			1	0	1		
				0	0	0	
				1	0	0	
				1	1	1	
					0	1	0
					1	0	1

ФКТ2 (F_2)

h	g	b	e	a	f	c	d
0	0	1					
0	1	0					
1	0	0					
1	0	1					
1	1	1					
	0	0	0				
	1	1	1				
	1	0	0				
		1	0	0			
			1	1	0		
			0	1	1		
				0	0	0	
				0	1	0	
				1	0	1	
				1	1	0	
					0	1	0
					0	0	0

ФКТ3 (F_3)

d	f	a	c	h	e	b	g
0	0	1					
0	1	0					
0	1	1					
1	1	0					
	0	0	0				
	0	1	1				
		0	1	0			
		1	0	1			
		1	1	1			
			1	1	0		
			1	0	0		
				0	0	0	
				1	1	1	
					0	1	0
					1	1	1

Таблица 3. Структуры компактных троек

СКТ1 (S_1)

a	b	c	d	e	f	g	h
0	0	1					
1	0	1					
1	1	0					
	0	1	0				
	0	1	1				
	1	0	0				
	1	0	1				
		0	0	1			
		0	1	0			
		0	1	1			
		1	0	1			
		1	1	1			
			0	1	1		
			1	0	0		
			1	1	0		
			1	1	1		
				0	0	1	
				1	0	1	
				1	1	0	
					0	1	1
					1	0	0

СКТ2 (S_2)

h	g	b	e	a	f	c	d
0	0	0					
0	1	1					
1	1	0					
	0	0	1				
	1	0	1				
	1	1	0				
		0	1	0			
		0	1	1			
		1	0	1			
			1	0	0		
			1	0	1		
			0	1	0		
			1	1	1		
				0	0	1	
				0	1	1	
				1	0	0	
				1	1	1	
					0	1	1
					0	0	1
					1	1	0
					1	1	1

СКТ3 (S_3)

d	f	a	c	h	e	b	g
0	0	0					
1	0	0					
1	0	1					
1	1	1					
	0	0	1				
	0	1	0				
	1	1	0				
	1	1	1				
		0	1	1			
		1	0	0			
		1	1	0			
			1	1	1		
			0	0	0		
			0	0	1		
			1	0	1		
				1	1	0	
				0	0	1	
				0	1	0	
				0	1	1	
					1	0	1
					0	1	1
					1	0	0
					1	1	0

4. Решение задачи о существовании совместных выполняющих наборов на основе генерирования СКТ с нулевыми СВН

Ключевым является метод решения задачи о существовании СВН для двух СКТ (назовем их S_1 и S_2), основанных на несовпадающих перестановках переменных. Предварительный этап решения заключается в унификации S_1 и S_2, так как унификация по принципу действия сохраняет в структурах-операндах все существующие СВН и одновременно минимизирует упомянутые структуры по числу элементов (строк).

Для унифицированных СКТ сохраним первоначальные обозначения, подчеркнув тем самым, что унификация заключается в замещении исходных СКТ эквивалентными по критерию включения СВН структурами. В нашем примере унифицированные структуры для СКТ S_1 и S_2, включенных в таблицу 3, представлены таблицей 4.

Примем S_1 за базисную структуру и установим для нее исходную нумерацию переменных: $x_1, x_2, \ldots, x_n$ (в рассматриваемом примере исходная нумерация переменных: a, b, …, h). В нашем иллюстративном примере отыскание СВН для СКТ S_1 и S_2 будет означать решение задачи 3-SAT для подформулы F' формулы F. В таблице 1 F' представлена подмножеством строк, на основе которых сформированы структуры S_1 и S_2 до их унификации.

Рассмотрим тривиальный частный случай: тройки 000 присутствуют на всех ярусах сравниваемых СКТ, что легко обнаруживается алгоритмически. Очевидно, что при этом исключаются трудности, связанные с различными перестановками, и какой-либо перебор: СВН представляет собой нулевой набор[1] длины n.

На этом факте базируется предлагаемый подход к решению задачи. Введем в рассмотрение операцию инвертирования элементов столбца — перемену местами нулей и единиц (кратко: *инвертирование столбца*). Дополним строку заголовка каждой СКТ двоичным вектором управления инвертированием столбцов (*вектором УИС*); поместим этот вектор над упомянутой строкой, соблюдая соответствие элементов в перестановках. Элемент 0 в векторе УИС означает сохранение столбца, элемент 1 — инвертирование.

Описанные средства позволяют модифицировать СКТ. Алгоритм полного перебора всех 2^n векторов УИС гарантирует обнаружение всех *нулевых* СВН для модифицированных СКТ или установление факта отсутствия СВН, при этом формируется информация об истинных СВН в первоначальных структурах.

Например, векторы УИС: 10101100 для S_1 и, соответственно, 00011110 для S_2, порождают модифицированные СКТ S_1^* и S_2^* (таблица 5).

ЗАМЕЧАНИЕ 1. В верхней строке приведенных табличных структур записан один и тот же вектор УИС, сопряженный с двумя перестановками переменных.

ЗАМЕЧАНИЕ 2. Каждый новый вектор УИС, сгенерированный при переборе, применяется для инвертирования столбцов первоначальных СКТ. Обнаружение в каждой структуре нулевой элементарной СКТ на некотором шаге перебора фиксирует нахождение совместного выполняющего набора для первоначальных СКТ, при этом сам СВН по построению совпадает с вектором УИС.

Таблица 4. Унифицированные СКТ S_1 и S_2

S_1

a	b	c	d	e	f	g	h
0	0	1					
1	0	1					
	0	1	0				
	0	1	1				
		1	0	1			
		1	1	1			
			0	1	1		
			1	1	0		
			1	1	1		
				1	0	1	
				1	1	0	
					0	1	1
					1	0	0

S_2

h	g	b	e	a	f	c	d
0	0	0					
1	1	0					
	0	0	1				
	1	0	1				
		0	1	0			
		0	1	1			
			1	0	0		
			1	0	1		
			1	1	1		
				0	0	1	
				0	1	1	
				1	1	1	
					0	1	1
					1	1	0
					1	1	1

[1] В контексте статьи нулевым набором будем называть набор с нулевыми компонентами.

Таблица 5. Модифицированные СКТ S_1^* и S_2^*

S_1^*

1	0	1	0	1	1	0	0
a	*b*	*c*	*d*	*e*	*f*	*g*	*h*
1	0	0					
0	0	0					
	0	0	0				
	0	0	1				
		0	0	0			
		0	1	0			
			0	0	0		
			1	0	1		
			1	0	0		
				0	1	1	
				0	0	0	
					1	1	1
					0	0	0

S_2^*

0	0	0	1	1	1	1	0
h	*g*	*b*	*e*	*a*	*f*	*c*	*d*
0	0	0					
1	1	0					
	0	0	0				
	1	0	0				
		0	0	1			
		0	0	0			
			0	1	1		
			0	1	0		
			0	0	0		
				1	1	0	
				1	0	0	
				0	0	0	
					1	0	1
					0	0	0
					0	0	1

На последнем утверждении остановимся подробнее, используя для иллюстрации таблицы 4 и 5. СКТ S_1 и S_2 содержат СВН 10101100 в терминах исходной нумерации переменных, но он явно «не виден» из-за несовпадающих перестановок переменных в двух структурах.

Подходящий вектор УИС, сгенерированный при переборе, аннулирует единицы в указанном наборе, трансформируя набор в нулевой, который легко обнаруживается в различных структурах. Из самого определения вектора УИС следует, что это возможно лишь в том случае, когда единицы в нем расположены на тех же местах, что и в приведенном СВН.

Здесь и далее будет явно проявляться дуализм в интерпретации одинаково обозначенных данных, отмеченный в самом начале статьи. В зависимости от контекста и цели применения нули и единицы рассматриваются как логические значения *ложь* и *истина* или как бинарные арифметические величины, в частности, компоненты двоичных последовательностей и векторов. Такая интерпретация составляет концептуальную основу предлагаемого исследования.

Для СКТ S_1 и S_2 перебор векторов УИС приводит к обнаружению пяти СВН:

a	b	c	d	e	f	g	h		h	g	b	e	a	f	c	d
0	0	1	0	1	1	0	0		0	0	0	1	0	1	1	0
0	0	1	1	1	1	0	0		0	0	0	1	0	1	1	1
1	0	1	0	1	1	0	0		0	0	0	1	1	1	1	0
1	0	1	1	1	1	0	0		0	0	0	1	1	1	1	1
0	0	1	1	1	0	1	1		1	1	0	1	0	0	1	1

и выводу о выполнимости формулы F'.

Таблица 6. Унифицированные СКТ S_1, S_2 и S_3

S_1

a	b	c	d	e	f	g	h
0	0	1					
1	0	1					
	0	1	1				
		1	1	1			
			1	1	0		
			1	1	1		
				1	0	1	
				1	1	0	
					0	1	1
					1	0	0

S_2

h	g	b	e	a	f	c	d
0	0	0					
1	1	0					
	0	0	1				
	1	0	1				
		0	1	0			
		0	1	1			
			1	0	0		
			1	1	1		
				0	0	1	
				1	1	1	
					0	1	1
					1	1	1

S_3

d	f	a	c	h	e	b	g
1	0	0					
1	1	1					
	0	0	1				
	1	1	1				
		0	1	1			
		1	1	0			
			1	1	1		
			1	0	1		
				1	1	0	
				0	1	0	
					1	0	1
					1	0	0

Процедура отыскания СВН для исходной формулы F рассматриваемого примера с использованием векторов УИС базируется на унифицированных СКТ S_1, S_2 и S_3, представленных в таблице 6 (основой для формирования структур таблицы 6 служит таблица 3). Для формулы F обнаруживаются два выполняющих набора: 00111011 и 10111100 при исходной нумерации переменных (a, b, …, h).

Отметим, что приведенные теоретические построения не претерпевают принципиальных изменений в случае больших значений n и k.

Дальнейшая часть работы посвящена решению задачи о нахождении СВН для множества компактных структур без использования перебора экспоненциальной сложности.

5. Структуры компактных пар

По аналогии с СКТ, введем в рассмотрение для любой 3-КНФ формулы системы *структур компактных пар* (СКП), принцип построения которых следует вышеизложенной концепции для компактных троек с некоторыми естественными отличиями. В частности, при n переменных число ярусов СКП равно $n - 1$, на каждом ярусе максимальное число пар равно 4 (00, 01, 10, 11), сочленение пар соседних ярусов допустимо при совпадении значений только одной переменной. При полностью заполненных ярусах СКП (как и СКТ) содержит 2^n двоичных наборов, образованных конкатенациями строк.

Любой СКТ можно сопоставить СКП путем разбиения каждой компактной тройки $x_j\ x_{j+1}x_{j+2}$ на две пары с одним общим элементом: $x_j\ x_{j+1}$ и $x_{j+1}\ x_{j+2}$ ($j = 1, \ldots, n - 2$), и размещения этих пар, представленных своими значениями, на соседних j-м и ($j + 1$)-м ярусах. Повторяющиеся пары на любом ярусе записываются однократно.

Для рассмотренного выше примера СКП G_1 и G_2, поставленные в соответствие унифицированным СКТ S_1 и S_2 (таблица 4), приведены в таблице 7.

Заметим, что каждая СКП, полученная преобразованием СКТ, по построению содержит все двоичные наборы, которые закодированы в соответствующей СКТ, и, в общем случае, еще некоторое множество новых наборов (назовем их *побочными наборами*). Например, G_1 включает конкатенацию 00101011, отсутствующую в S_1.

Причиной появления побочных наборов является образование в СКП при трансформации структур новых (по сравнению с СКТ) конкатенаций строк соседних ярусов, которые соответствуют несуществующим в СКТ компактным тройкам. В таблице 7 пары строк, одновременное включение которых в СВН недопустимо, отмечены в первых столбцах структур одноименными метками, так как тройки 010 ($d\ e\ f$) в S_1 и 010 ($f\ c\ d$) в S_2 отсутствуют. Несложный алгоритмический анализ допустимости конкатенаций пар на соседних ярусах СКП путем их сравнения с породившими их тройками СКТ должен завершаться расстановкой упомянутых выше меток (обязательно до модификации структур).

Само по себе существование недопустимых пар строк не является препятствием для поиска СВН, так как любая строка с меткой по построению образует конкатенации и со строками без меток, расположенными на соседних ярусах, — эти конкатенации и есть те компактные тройки, которые породили компактные пары. Поэтому метки играют для алгоритма поиска СВН роль «указателей» некорректного выбора пар, которого следует избежать; теоретически, неизбежность такого выбора равносильна разновидности конфликта.

Таблица 7. Структуры компактных пар G_1 и G_2

СКП1 (G_1)

a	b	c	d	e	f	g	h
0	0						
1	0						
	0	1					
		1	0				
		1	1				
1			0	1			
			1	1			
1				1	0		
				1	1		
					0	1	
					1	0	
						1	1
						0	0

СКП2 (G_2)

h	g	b	e	a	f	c	d
0	0						
1	1						
	0	0					
	1	0					
		0	1				
			1	0			
			1	1			
				0	0		
				0	1		
				1	1		
2					0	1	
					1	1	
						1	1
2						1	0

Сказанное распространяется и на общий случай применения описываемого метода решения задачи для больших значений *n* и *k*.

6. Полиномиальная процедура для решения задачи о существовании СВН

Завершающий этап конструктивного подхода к решению задачи базируется на модификации системы СКП с целью предъявления нулевого СВН, с одновременным формированием вектора *U* (УИС) для исходной системы, или установления факта отсутствия СВН, означающего невыполнимость исходной КНФ.

Схема распространения нулевых значений

Логическая формула

$$\bigwedge_{t=1}^{m_1} \neg(s_{i_t} \vee v_{j_t}), \tag{1}$$

где литералы s_{i_t} и v_{j_t} обозначают несовпадающие переменные задачи или их инверсии,

m_1 — число дизъюнктов, сформированных на основе системы СКП, $m_1 \le (n-1)k$, служит основой для описания действий, направленных на построение нулевого двоичного набора.

Предварительно стартовый вектор *U* приобретает значения 1 для компонент, помечающих столбцы СКП с единичными константными значениями, что означает инвертирование таких столбцов (необходимая часть схемы распространения нулевых значений). Столбцы, заполненные нулевыми компонентами, на последующих шагах процедуры не претерпевают изменений.

Далее для произвольно выбранной *стартовой* переменной x_{i_1} (любого литерала в (1)) фиксируется значение; естественно сначала выбрать $x_{i_1} = 0$. В системе СКП отыскиваются все пары, содержащие $x_{i_1} = 0$, безразлично, на первом или втором месте по порядку. Пусть на некотором ярусе (ярусах) в паре с x_{i_1} оказалась переменная x_j ($j \ne i_1$). Как показано на рис. 1, на соответствующих ярусах $x_{i_1} = 0$ может присутствовать в двух парах или в одной:

а)

x_{i_1}	x_j
0	0
0	1

б)

x_{i_1}	x_j
0	0

в)

x_{i_1}	x_j
0	1

Рис. 1. Варианты присутствия пар

В случае (а) переменная x_j не фиксируется: каждое из двух значений уточняется в дальнейшем или переменная остается «свободной» (с любым значением), что связано с вариантностью допустимого решения. В случае (б) фиксируется $x_j = 0$, в случае (в) — $\neg\, x_j = 0$, при этом вектор *U* инвертирует столбец x_j во всех структурах, обращая значение 1 в 0. Сказанное относится и к перестановке $x_j\, x_{i_1}$ в паре.

Если для переменной x_{i_1} зафиксировано значение 1, то соответствующие столбцы инвертируются ($\neg\, x_{i_1} = 0$), а значение 1 перемещается в компоненту вектора *U*; далее отыскиваются новые переменные, образующие пары с $\neg\, x_{i_1}$, и повторяются действия, описанные выше для случая $x_{i_1} = 0$.

Логическим следствием первого шага (использования нулевого значения стартовой переменной) является фиксация нулевых значений ряда переменных, что эквивалентно перечислению пар скобок с нулевыми литералами s_{i_t} и v_{i_t} в (1). Эти переменные, в свою очередь, продолжают (подобно x_{i_1}) по приведенной схеме порождение новых переменных с нулевыми значениями в модифицированных СКП до тех пор, пока в системе СКП возможны импликации, отображенные на рис. 1, б или рис. 1, в.

Заметим, что в этой части процедуры нет никаких возвратов. Фиксация нулевых значений абсолютно вынуждена. Целью процедуры является получение истинного (в арифметической интерпретации *единичного*) значения формулы (1). Дизъюнкты в (1) не заданы первоначально, а формируются в процессе анализа таблиц СКП.

Варианты продолжения и завершения процедуры

1. Конфликт.

Обнаружено противоречие при фиксации константных значений для некоторой переменной хотя бы в двух различных СКП.

Конфликт не приводит к завершению процедуры, если она не испытана с альтернативным значением стартовой переменной: $x_{i_1} = 1$ ($\neg\, x_{i_1} = 0$) для исходных СКП. И лишь при повторной неудаче (противоречии) устанавливается отсутствие СВН.

2. Одно из альтернативных значений стартовой переменной инициирует фиксацию значений всех переменных (вариант наиболее эффективной реализации процедуры, обусловленный исходными данными).

3. Значение стартовой переменной порождает бесконфликтную фиксацию значений для подмножества $X_1 \subset X$, где *X* — множество всех переменных. Для множества $X_2 = X \setminus X_1$ задача еще не решена. Далее процедура формирует новую стартовую переменную x_{i_2} для множества $X_2 \subset X$ и продолжает применение схемы распространения нулевых значений для переменных $x_{i_t} \in X_2$ с использованием сущест-

вующих на этом шаге модифицированных СКП в качестве стартовых для подзадачи. Переменные из множества X_1 и соответствующие столбцы уже не изменяют своих значений; подзадача для X_2 является автономной и от нее зависит общий результат.

Заметим, что для первой стартовой переменной альтернативное значение уже не потребуется, в отличие от второй, для которой оно возможно (до формирования новой подзадачи) — согласно концепции, на которой основана процедура.

Далее процесс формирования новых подмножеств X_3, …, X_w и, соответственно, новых стартовых переменных может продолжаться. Подзадача с заключительным подмножеством X_w является определяющей для общего результата.

В X_w могут быть включены и те переменные, выбор значений которых произволен (см. рис. 1, а), — при условии обнаружения выполнимости исходной формулы.

Детализация описания процедуры определяет некоторые шаги алгоритма как *точки старта подзадач* и, соответственно, выбора стартовых переменных и их значений для подзадач. Последующие шаги алгоритма могут быть связаны с возвратом некоторой подзадачи к *ее* точке старта с целью использования для вычислений альтернативного значения последней стартовой переменной. Возврат сопряжен с восстановлением структур, существовавших при первом старте подзадачи. Значения переменных, зафиксированные при решении всех предшествующих подзадач, и соответствующие столбцы СКП остаются неизменными.

Пример 1. Решение задачи для двух СКП, приведенных в таблице 7.

Зафиксируем для стартовой переменной *a* значение 0. В таблице 8 отражено действие стартового вектора *U:* замена единичных столбцов *c* и *e* нулевыми столбцами.

Анализ в модифицированных структурах G_1^* и G_2^* столбцов, содержащих строки с $a = 0$, приводит к выделению пар *a b* = 00, *e a* = 00, не влекущих никаких изменений таблиц (*b* и *e* — константы), а также *a f* = 00 и *a f* = 01. Две последние пары не влекут фиксации значения *f* (см. рис. 1, а), и эта переменная на данном шаге остается неопределенной.

Итак, при $a = 0$ действие стартовой переменной *a* заканчивается сохранением всех столбцов в модифицированных структурах. Отсутствие конфликта значений переменных означает корректность выбора: привлечение альтернативного значения *a* в дальнейшем не потребуется.

Процедура продолжается для подзадачи, содержащей переменные *d, f, g, h*, — соответствующие столбцы в таблицах для G_1^* и G_2^* выделены затенением.

Старт для подзадачи начнем с $f = 0$ (выбор переменной и ее значения произволен).

Используем далее обозначение импликации «→».

$(f = 0) \rightarrow (\neg\, g = 0) \rightarrow (\neg\, h = 0)$ (см. столбцы *f, g, h* в СКП G_1^*);

$(f = 0) \rightarrow (\neg\, d = 0)$ (см. столбцы *f, c, d* в СКП G_2^*: при $f = 0$ метки допускают для пары *c d* выбор только предпоследней строки; константа $c = 0$ не входит явно в подмножество подзадачи и играет пассивную транзитную роль для передачи значений).

Модификация структур на основе приведенных импликаций отражена в таблице 9.

Процедура фиксации значений всех переменных завершена. Заключительный вектор U = 00111011 представляет один из выполняющих наборов для формулы F'.

Для полноты картины рассмотрим действие процедуры для единичного значения стартовой пере-

Таблица 8. Разбиение множества переменных в модифицированных СКП

СКП1 (G_1^*)

	a	b	c	d	e	f	g	h
a=0	0	0						
	1	0						
		0	0					
			0	0				
			0	1				
	^			0	0			
				1	0			
	^				0	0		
					0	1		
						0	1	
						1	0	
							1	1
							0	0

Вектор U

	a	b	c	d	e	f	g	h
a=0	0	0	1	0	1	0	0	0

СКП2 (G_2^*)

h	g	b	e	a	f	c	d
0	0						
1	1						
	0	0					
	1	0					
		0	0				
			0	0			
			0	1			
				0	0		
				0	1		
				1	1		
^^					0	0	
					1	0	
						0	1
^^						0	0

Таблица 9. Модификация структур в подзадаче

СКП1 (G_1^{**})

	a	b	c	d	e	f	g	h
a=0	0	0						
	1	0						
		0	0					
			0	1				
			0	0				
	^			1	0			
				0	0			
	^				0	0		
					0	1		
f=0						0	0	
						1	1	
							0	0
							1	1

Вектор U

	a	b	c	d	e	f	g	h
a=0	0	0	1	0	1	0	0	0
f=0	0	0	1	1	1	0	1	1

СКП2 (G_2^{**})

h	g	b	e	a	f	c	d
1	1						
0	0						
	1	0					
	0	0					
		0	0				
			0	0			
			0	1			
				0	0		
				0	1		
				1	1		
^^					0	0	
					1	0	
						0	0
^^						0	1

менной: $a = 1$ ($\neg\, a = 0$). Несложный анализ столбцов таблицы 8 с инверсией столбцов, озаглавленных переменной *a,* порождает импликации:

$$(\neg\, a = 0) \rightarrow (\neg\, f = 0) \rightarrow (g = 0) \rightarrow (h = 0);$$

подзадача для d : d = * (любое из двух значений допустимо).

Итак, вектор $U = 101 * 1100$ представляет два СВН.

Еще два СВН выражаются записью $U = 001{*}1100$ (стартовые значения для задачи и подзадачи представлены первой и шестой компонентами: $a = 0$, $f = 1$).

Пример 2. Решение задачи 3-SAT для исходной формулы F.

Таблица 10 построена на основе таблицы 6 для унифицированных СКТ. Запрещающие метки для пар строк присутствуют только в СКП H_3.

С целью повышения наглядности вывода в начальной таблице значения констант $c = 1$ и $e = 1$ заменены на инверсные и, соответственно, единичные столбцы трансформированы в нулевые

$$(U = 00101000).$$

Пары меток запрещают компактные тройки

$$a\ c\ h = 010$$

$$\text{и } a\ c\ h = 111$$

(а не 000 и 101), так как метки расставлены до инвертирования столбца c. Затененные пары в таблице 10 поясняют возникновение следующих логических следствий, составляющих эффективный вывод со стартового значения $a = 0$:

$$(a = 0) \rightarrow (f = 0) \rightarrow (\neg\, d = 0) \rightarrow$$
$$\rightarrow (\neg\, g = 0) \rightarrow (\neg\, h = 0).$$

Таблица 11 отражает результаты распространения нулей в системе СКП. Пары 00, выделенные затенением, присутствуют на каждом ярусе всех СКП и в совокупности представляют нулевой совместный выполняющий набор в преобразованных структурах. Соответствующий СВН для исходных структур совпадает с вектором $U = 00111011$.

Для установления факта выполнимости булевой формулы, естественно, достаточным является обнаружение одного СВН, что в ряде случаев влечет раннее завершение рассмотренной процедуры.

Пример 3. Решение задачи для системы СКП при отсутствии СВН.

Вывод результата требует применения альтернативных значений стартовой переменной:

$$(a = 0) \rightarrow (f = 1) \rightarrow (g = 0) \rightarrow (h = 0)$$
$$(g = 0) \rightarrow (h = 1) \text{ — конфликт;}$$
$$(a = 1) \rightarrow (f = 0) \rightarrow (g = 1) \rightarrow (h = 1)$$
$$(g = 1) \rightarrow (h = 0) \text{ — конфликт;}$$
вывод: СВН отсутствует.

В силу простоты вывода полная схема табличного вывода с инвертированием столбцов и распространением нулей здесь не развернута. Логические следствия для альтернативных значений переменной a поясняются в таблице 12 затенением различной интенсивности пар клеток, подходящих для этой цели.

7. Классификация предложенной модели и выводы

Обратимся к формуле, описывающей известную задачу 2-КНФ:

$$\bigwedge_{t=1}^{m_2} (s_{i_t} \vee v_{j_t}), \tag{2}$$

Таблица 10. Унифицированные СКП H_1, H_2, H_3

H_1

a	b	c	d	e	f	g	h
0	0						
1	0						
	0	0					
		0	1				
			1	0			
				0	0		
				0	1		
					0	1	
					1	0	
						1	1
						0	0

H_2

h	g	b	e	a	f	c	d
0	0						
1	1						
	0	0					
	1	0					
		0	0				
			0	0			
			0	1			
				0	0		
				1	1		
					0	0	
					1	0	
						0	1

H_3

d	f	a	c	h	e	b	g
1	0						
1	1						
	0	0					
	1	1					
^		0	0				
~		1	0				
~			0	1			
^			0	0			
				1	0		
				0	0		
					0	0	
						0	1
						0	0

Таблица 11. Преобразованные СКП H_1, H_2, H_3

H_1

a	b	c	d	e	f	g	h
0	0						
1	0						
	0	0					
		0	0				
			0	0			
				0	0		
				0	1		
					0	0	
					1	1	
						0	0
						1	1

H_2

h	g	b	e	a	f	c	d
1	1						
0	0						
	0	0					
	1	0					
		0	0				
			0	0			
			0	1			
				0	0		
				1	1		
					0	0	
					1	0	
						0	0

H_3

d	f	a	c	h	e	b	g
0	0						
0	1						
	0	0					
	1	1					
^		0	0				
~		1	0				
~			0	0			
^			0	1			
				0	0		
				1	0		
					0	0	
						0	0
						0	1

Таблица 12. Система СКП L_1, L_2, L_3

L_1

a	b	c	d	e	f	g	h
0	0						
1	0						
	0	1					
		1	1				
			1	1			
				1	0		
				1	1		
					0	1	
					1	0	
						1	1
						0	0

L_2

h	g	b	e	f	c	d	a
1	1						
0	0						
	0	0					
	1	0					
		0	1				
			1	0			
			1	1			
				0	1		
				1	1		
					1	1	
						1	0
						1	1

L_3

b	c	d	e	a	f	h	g
0	1						
	1	1					
		1	1				
			1	0			
			1	1			
				0	1		
				1	0		
					0	1	
					1	0	
						1	0
						0	1

где литералы s_{i_t} и v_{j_t} обозначают несовпадающие переменные задачи или их инверсии;
m_2 — число дизъюнктов в КНФ.

Трудно не заметить сходства в строении формул (1) и (2): переход от (2) к (1) состоит лишь в добавлении к каждому дизъюнкту в (2) знака отрицания.

Формула (2) разрешается известным полиномиальным алгоритмом, реализующим процесс распространения ограничений [4]. Алгоритм стремится придать значение *истина* («1») каждому дизъюнкту, последовательно назначая подходящие значения переменным. Более подробно: первый дизъюнкт получает значение 1 простой фиксацией значения (x_{i_1} или $\neg\, x_{i_1}$) для любого литерала первой скобки. Тем самым выбирается стартовая переменная x_{i_1} и ее значение, индуцирующие, в общем случае, единичные значения ряда дизъюнктов; части остальных дизъюнктов можно придать истинные значения выбором подходящих значений новых переменных. Далее варианты продолжения и завершения алгоритма повторяют соответствующие варианты процедуры формирования нулевых наборов СКП с естественной поправкой на целевое назначение отдельных шагов.

В обеих процедурах вывода используются следующие аналитические действия: обнаружение конфликтов (противоречивых значений переменных); выявление подзадач для подформул и назначение соответствующих стартовых переменных, для которых предусмотрены испытания альтернативных значений; распознавание выполнимости формул и формирование выполняющих наборов. При решении подзадач обе процедуры не допускают никаких возвратов к переменным, значения которых зафиксированы в предшествующих подзадачах.

Различия в целевом назначении шагов: для формулы (2) — построить на очередном шаге дизъюнкт со значением *истина* с учетом уже завершенных построений; для формулы (1) — построить на очередном шаге компактную пару 00 на основе системы таблиц СКП, сформировав тем самым очередную пару скобок с нулевыми значениями литералов в (1).

Различия в сложности вычислений, вытекающие из организации данных, минимальны: формула (2) — линейный массив пар скобок, содержащих по два литерала; закодированная формула (1) — двумерный массив двоичных компактных пар, которые могут быть сопоставлены дизъюнктам в (2).

Очевидно, что асимптотическая вычислительная сложность обеих процедур обоснованно классифицируется как полиномиальная. Тот же вывод относительно процедуры формирования нулевых наборов СКП непосредственно следует из разделов 5 и 6.

Заключение

В целом приведенная алгоритмическая модель представления задачи 3-SAT и ее разрешения относится к классу полиномиальных моделей. Полное отсутствие эвристик и громоздких систем *гиперструктур* коренным образом отличают данную модель от более ранних разработок автора [2, 3], вызвавших резонанс и достаточно широкое открытое обсуждение в среде программистов.

Работа первична по используемым средствам. Структуры компактных троек и компактных пар представляют собой компоненты специального назначения. Несогласованность структур, построенных на основе различных перестановок переменных, преодолевается сведением поиска СВН к формированию и распознаванию общих для всех структур нулевых последовательностей.

Ключевым пунктом работы можно также назвать адаптацию метода распространения ограничений, используемого в оптимизационных задачах, к разрешению булевой формулы, закодированной средствами СКТ и СКП.

Результаты исследования модели рассчитаны на обобщения в силу полиномиальной сводимости трудноразрешимых задач друг к другу.

Литература

1. *Романов В. Ф.* Неортодоксальные модели для задач дискретного анализа и оптимизации. LAP LAMBERT Academic Publishing: Saarbrucken, Germany, 2012. 130 с.
2. *Romanov V. F.* Non-orthodox combinatorial models based on discordant structures. Electronic journal «Investigated in Russia» — http://zhurnal.ape.relarn.ru/articles/2007/143e.pdf
3. *Romanov V. F.* Non-orthodox combinatorial models based on discordant structures. ArXiv.org., 2011 — http://arxiv.org/abs/1011.3944
4. *Лорьер Ж.-Л.* Системы искусственного интеллекта. М.: Мир, 1991. 568 с.

Романов Владимир Федорович. С. н. с. Владимирского ГУ. К. т. н., профессор. Окончил в 1969 г. Горьковский ГУ им. Н. И. Лобачевского. Количество печатных работ: 98, в том числе 13 авторских свидетельств и патентов. Область научных интересов: информационные технологии в задачах дискретной оптимизации. E-mail: romvf@mail.ru