

РЕБУС – ИНТЕЛЛЕКТУАЛЬНЫЙ РЕШАТЕЛЬ КОМБИНАТОРНЫХ ЗАДАЧ В БУЛЕВЫХ ОГРАНИЧЕНИЯХ*

В статье рассматриваются архитектура, функциональные возможности и технология применения инструментального комплекса РЕБУС, ориентированного на использование в фундаментальных и прикладных исследованиях при решении задач в разнообразных областях знаний, где естественным образом возникают дискретные модели в виде систем булевых уравнений (ограничений). Методы и средства автоматизации представления, накопления, модификации и использования знаний при решении задач удовлетворения булевых ограничений составляют основу организации решателя РЕБУС.

Ключевые слова: булевы уравнения и ограничения, комбинаторные задачи, булева выполнимость.

Введение

Одним из перспективных направлений исследований в современном программировании является направление, связанное с программированием в ограничениях. К важному классу ограничений относятся булевы ограничения, представляемые в виде систем булевых уравнений, и связанные с этим классом задача выполнимости булевых ограничений (Boolean Constraint Satisfaction Problem – BCSP) и задача программирования в булевых ограничениях (Constraint Boolean Programming – CBP).

Задача булевой выполнимости является фундаментальной проблемой в математической логике и теории вычислений. Многие практические задачи могут быть сформулированы в терминах решения системы булевых уравнений. Существует достаточно большое количество программ решения задачи булевой выполнимости¹. В качестве входного языка этих программ используется фрагмент международного формата DIMACS² представления булевых ограничений в виде конъюнктивной нормальной формы пропозициональной логики (так называемый CNF-формат; SAT-формат и его расширения для представления булевых функций общего вида практически не используются ввиду отсутствия решателей, ориентированных на этот формат). Следует отметить, что средства автоматизации построения булевых моделей (в CNF-формате) по содержательной (ориентированной на конечного пользователя) постановке задачи в этих программах отсутствуют, взаимодействие с программой возможно только в режиме командной строки. Таким образом, актуальной является проблема разработки принципиально новой интегрированной инструментальной среды, объединяющей решатели комбинаторных задач большой размерности в единую систему, позволяющую автоматизировать процесс построения булевой модели решаемой дискретной задачи, выбирать из базы знаний (исходя из свойств полученной модели и накопленной статистики решения задач) подходящий алгоритм решения, проводить вычислительный эксперимент, сохранять результаты расчетов в пользовательской базе данных, включать в базу знаний инструментальной среды новые модели и алгоритмы.

* Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (проект № 06-01-00340).

¹ <http://www.lri.fr/~simon/satex/satex.php3>.

² <http://www.satlib.org>.

В статье рассматриваются концептуальная модель, принципы организации и архитектура интегрированной инструментальной визуальной среды ИК РЕБУС, предназначенной для автоматизации программирования в булевых ограничениях с использованием полного формата DIMACS (CNF-формат & SAT-формат).

Постановка задачи

Формально задача решения булевых уравнений формулируется следующим образом [Левченков, 1999]: пусть $X = (x_1, \dots, x_n)$ – упорядоченное множество булевых переменных ($x_i \in B_2$ для всех $i = \overline{1, n}$; $B_2 = \{0, 1\}$). Задано m булевых соотношений (система уравнений) вида

$$f_i(x_{i_1}, \dots, x_{i_{k(i)}}) = 0, \quad (1)$$

где $f_i (i = \overline{1, m})$ – произвольные булевы функции своих аргументов. Требуется найти одно, несколько или все решения системы (1) или установить ее несовместность.

Следует заметить, что первая NP-полная задача – задача «выполнимости КНФ» (задача булевой выполнимости или SAT-задача) есть, по существу, задача распознавания совместности систем уравнений [Горшков, 1995]:

$$x_{i_1}^{\sigma_{i_1}} \vee \dots \vee x_{i_{k(i)}}^{\sigma_{i_{k(i)}}} = 1, \quad (i = \overline{1, \dots, m}), \quad (2)$$

где

$$x_{i_p}^{\sigma_{i_p}} = \begin{cases} \overline{x_{i_p}} & \text{при } \sigma_{i_p} = 0, \quad p = \overline{1, k(i)} \\ x_{i_p} & \text{при } \sigma_{i_p} = 1. \end{cases}$$

Двойственная к (2) система уравнений вида (1) записывается следующим образом:

$$\overline{x_{i_1}^{\sigma_{i_1}}} \wedge \dots \wedge \overline{x_{i_{k(i)}}^{\sigma_{i_{k(i)}}}} = 0, \quad (i = \overline{1, \dots, m}). \quad (3)$$

Известно, что любая система вида (1) может быть представлена одним эквивалентным уравнением $f(x_1, \dots, x_n) = \bigcup_{i=1}^m f_i(x_{i_1}, \dots, x_{i_{k(i)}}) = 0$. Поэтому далее будем рассматривать в основном одно стандартное уравнение

$$f(x_1, \dots, x_n) = 0. \quad (4)$$

В инструментальном вычислительном комплексе РЕБУС алгоритмы решения задач BCSP, составляющие вычислительное ядро, ориентированы на часто встречающиеся в приложениях так называемые большие разреженные системы булевых уравнений, когда в (1) выполняется условие $m \gg n \gg k(i)$ (символом « \gg » обозначено отношение «много больше»). В частности, для многих практических задач $k(i)$ принимает только два фиксированных значения: $k(i) = 2$ и $k(i) = p$ ($p > 2$), причем бинарных ограничений в системе (1) много больше, чем p -арных. Примерами таких систем являются полученные с помощью базовой технологии моделирования ИК РЕБУС булевы модели для решения ряда классических дискретных задач (задача об n -ферзях, раскраска графа, покрытие конечного множества системой его подмножеств, ход шахматного коня, задача о голубях и клетках), а также булевы модели для решения некоторых практических задач (например, равномерная загрузка локомотивного депо железной дороги, составление оперативного плана отправления локомотивных бригад [Богданова, 2005]).

Основы автоматизация программирования в булевых ограничениях в ИК РЕБУС

Подход к спецификации задачи как задачи удовлетворения ограничений требует создания специальных средств представления и обработки знаний. Это обусловлено следующими обстоятельствами. Во-первых, такой подход позволяет избежать процедурных деталей спецификации путем задания модели задачи. Оперировать декларативными спецификациями на-

много проще, чем алгоритмическими [Ушаков, Телерман, 2000]. Во-вторых, выделение задач удовлетворения булевых ограничений в отдельный класс привело к появлению универсальных алгоритмов их решения, а также к разработке стандартного формата для представления системы ограничений³, позволяющих применять к задачам, представленным в этом формате, известные решатели. Новые специальные средства представления и обработки знаний должны обеспечивать совместимость с этими ранее разработанными средствами.

Постановка задачи в виде системы ограничений приносит свою специфику в аппарат знаний. Эта специфика определяется следующими свойствами:

- расслоением знания, возникающим при формализации сложной проблемной области (ПО). При программировании в булевых ограничениях естественным образом выделяется ряд тесно связанных слоев, представляющих: 1) булевы модели дискретных задач; 2) методы решения систем булевых уравнений, алгоритмы, разработанные на основе этих методов, и программы, реализующие алгоритмы на одном из языков программирования; 3) статистические данные, накапливаемые в результате проведения вычислительных экспериментов, и результаты их обработки;

- присутствием в ПО как формализованных знаний (имеются в виду булевы модели, методы и алгоритмы их решения), так и экспертных знаний (речь идет о выборе алгоритма и управляющих параметрах вычислительного эксперимента);

- разнообразием форм описания булевой модели (формулы алгебры логики, формат LaTeX, DIMACS формат и др.);

- высокой размерностью вектора булевых переменных;

- сильной зависимостью вычислительного алгоритма от представления булевой модели (структуры данных, которая ее характеризует).

Тесно связаны со спецификой ПО требования к организации проведения вычислительного эксперимента, главными из которых являются:

- возможность построения, редактирования булевой модели и преобразования этой модели из одной формы представления в другую;

- генерация программы-вычислителя булевой функции;

- наличие механизмов специфицирования, составляющих программу-решатель модулей;

- наличие гибких управляющих программных структур реконфигурации программ-решателей;

- возможность включения новых алгоритмов, решателей и моделей;

- необходимость проведения быстрых расчетов в диалоговом режиме и ресурсоемких в автономном режиме;

- необходимость проведения многовариантных расчетов и средств их обеспечения (в виде пользовательских баз данных для хранения вариантов булевых моделей и результатов решения);

- ограниченность вычислительных ресурсов ЭВМ требует предусмотреть возможность модификации и расширения инструментальных средств для проведения распределенных и параллельных вычислений.

Методологической основой автоматизации проведения вычислительных экспериментов в области решения дискретных задач является использование подхода, базирующегося на знаниях. В связи с этим вопросы представления, накопления, модификации и использования знаний занимают центральное место в технологии, используемой ИК РЕБУС.

Языком описания булевых ограничений в ИК РЕБУС является визуально-ориентированный язык ЯСФОР (Язык Содержательных Формулировок), в котором для ввода ограничений используются интуитивно понятные средства визуального интерфейса. Нужно отметить, что для построения системы ограничений не обязательно знать какой-либо традиционный язык программирования, хотя ИК РЕБУС имеет также встроенные средства и для задания системы ограничений на языках Фортран и С++. Такой подход позволяет работать в этой системе широкому кругу пользователей. При построении ограничений средствами визуального интерфейса запускается встроенный редактор, преобразующий ограничения в текстовый формат языка ЯСФОР. Предусмотрена также обратная возможность трансляции ограничений из описания булевой модели на этом языке в формат DIMACS.

³ <http://www.satlib.org>.

Авторские алгоритмы решения систем булевых уравнений ИК РЕБУС [Опарин, Богданова, 2001] разработаны на основе дискретного подхода и относятся к классу полных методов поиска с возвратом. Преимуществом первых двух алгоритмов является то, что они предназначены для решения уравнения (4), левая часть которого $f(x_1, x_2, \dots, x_n)$ представлена булевой формулой общего вида. Третий алгоритм разработан для решения частного случая уравнения (4), когда булева формула представлена в дизъюнктивной нормальной форме (система вида (3)). Алгоритмы используют метод интеллектуального возврата, статические и динамические стратегии для определения локальной несогласованности, схемы обратного просмотра, трехзначную и многозначную логику.

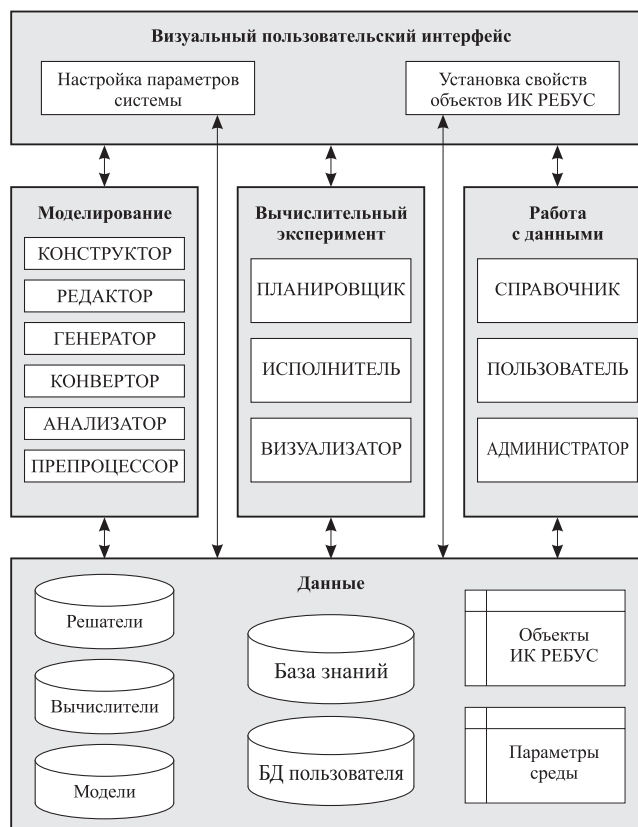
Технология работы в ИК РЕБУС включает три этапа обработки информации: моделирование, проведение вычислений и работу с базой знаний.

Конечной целью первого этапа является получение булевой модели, которая может быть представлена как в формате DIMACS, так и на языке ЯСФОР или в виде подпрограммы вычисления булевой функции. На этом этапе работы поддерживается принцип ориентации на пользователей различной квалификации. Учет квалификации пользователя осуществляется путем реализации различных режимов моделирования и разнообразия форматов представления булевой модели. На втором этапе проводится вычислительный эксперимент для выбранной модели с помощью подходящего для этой модели решателя согласно постановке задачи, указанной пользователем. Многообразие решателей с различными вычислительными характеристиками предусмотрено для поддержки принципа полноты функционального наполнения. Формат представления булевой модели является одним из условий выбора решателя. В качестве остальных условий могут быть заданы метод, алгоритм и вид постановки задачи. По этим условиям формируется запрос к базе знаний. Результатом запроса является список подходящих решателей, один из которых выбирается для выполнения. Результаты работы решателя подлежат постпроцессорной обработке, предусматривающей их анализ, визуализацию и сохранение в базе данных пользователя. Третий этап работы предназначен для ввода, редактирования и обработки информации из базы знаний ИК РЕБУС. Возможность введения в базу знаний данных о новых содержательных формулировках, видах ограничений, решателях, вычислителях обеспечивает свойство расширяемости программного комплекса.

На рисунке представлена архитектура ИК РЕБУС в виде структурной модели, отображающей на самом общем уровне связь между следующими тремя основными подсистемами: моделирования; вычислительного эксперимента; работы с базой знаний.

Обмен информацией между подсистемами организован следующим образом. Вся совместно используемая информация о предметной области хранится в общей базе знаний ИК РЕБУС, доступной всем подсистемам.

Промежуточные результаты обработки информации на всех этапах работы ИК, являющиеся входными для других этапов работы, передаются в виде файлов, сохраняющихся в каталогах ИК РЕБУС. Конечные результаты работы сохраняются в файлах и БД пользователя. Данные, представляющие собой значение свойств объектов ИК, служащих для настройки интерфейса, и параметры среды хранятся в оперативной памяти.



Технология булева моделирования

Хорошо известно, что поскольку булевы системы имеют с точки зрения человеческого восприятия весьма сложную и специфическую природу, важной и нетривиальной задачей является построение средств, позволяющих человеку моделировать задачи на языке булевых ограничений.

Базовая технология автоматического синтеза булевых ограничений по содержательному описанию дискретной задачи, применяемая в ИК РЕБУС, была разработана в процессе исследования построения булевых моделей для ряда классических и практических дискретных задач. Анализ этого процесса позволил выделить характерные особенности, классифицирующие дискретные задачи по ряду признаков.

При классификации дискретных задач использовалось два признака. Первый признак – это зависимость булева вектора X от дискретного времени $t = 0, 1, 2, \dots, k$. С этой точки зрения мы различаем *статические* (или одношаговые) задачи, вектор состояния которых не зависит от времени t , и *динамические* (или многошаговые) задачи, вектор состояния которых зависит от времени t .

В первом случае булевы ограничения (в векторной форме) имеют вид $F(X) = 0$, во втором случае рассматривается уравнение вида $F(X_0, X_1, \dots, X_k) = 0$, где $X_i = X(t)$ при $t = i$.

Второй признак – это структуризация множества X булевых переменных задачи. Здесь рассматриваются варианты *линейного упорядочения множества X* (X рассматривается как вектор независимых переменных), и вариант *упорядочения множества X в виде многомерного массива* булевых переменных.

Приведенная классификация является в определенной степени условной, и существуют дискретные задачи, которые занимают промежуточное положение в рамках этой классификации. В общем случае размерность динамических дискретных задач (при прочих равных условиях) существенно выше размерности статических задач, так как для динамической (многошаговой) задачи необходимо определить значение вектора состояния для каждого момента времени $t = 0, 1, 2, \dots, k$.

Можно отметить, что в рамках этой классификации к классу статических задач с булевой матрицей состояния относятся задачи об n -ферзях, голубях и клетках, раскраске графа. К классу динамических задач относятся, например, задача о покрытии конечного множества системой его подмножеств и задача Эйлера о ходе шахматного коня (в первой задаче множество переменных состояния представлено в виде вектора, а во второй – в виде трехмерного массива).

В ИК РЕБУС в качестве программного средства построения компьютерной модели выступает подсистема моделирования. Построение компьютерной модели включает следующие этапы:

- задание множества булевых переменных $X = \{x_1, x_2, \dots, x_n\}$;
- задание системы булевых ограничений (множество $C = \{C_1, C_2, \dots, C_m\}$);
- представление системы ограничений в одном из форматов входных данных программ-решателей ИК РЕБУС.

Выбор множества булевых переменных X осуществляется пользователем на этапе формализации булевой модели. Подсистема моделирования структурирует это множество в виде матрицы булевых переменных. Для динамических задач строки матрицы соответствуют булевым переменным на очередном шаге, для статических задач такое соответствие определяется спецификой решаемой задачи.

Для автоматизации второго и третьего этапов была разработана специальная методика моделирования. Эта методика предоставляет два различных способа построения компьютерной модели: автоматический синтез булевой модели в виде текста программ на языке программирования (Фортран, С++) либо синтез системы булевых ограничений в международном формате DIMACS.

Оба способа построения булевой модели предполагают многократное (по числу ограничений) повторение следующих двух шагов: 1) выбрать подмножество множества X булевых переменных; 2) задать для этого подмножества булевы ограничения (уравнения) с помощью одной из содержательных формулировок, предусмотренных в ИК РЕБУС.

Если обозначить произвольное подмножество булевых k переменных из X как $Y = \{y_1, y_2, \dots, y_k\}$, а означенный набор этих переменных как $\{a_1, a_2, \dots, a_k\}$, тогда содержательные формулировки для задания ограничений над булевыми переменными и соответствующие им булевы уравнения имеют вид:

— отсутствие единиц в наборе $\{a_1, a_2, \dots, a_k\}$, или $\bigcup_{i=1}^k y_i = 0$;

— одна и только одна единица в наборе $\{a_1, a_2, \dots, a_k\}$, или $\bigcup_{i=1}^{k-1} \bigcup_{j=i+1}^k y_i \wedge y_j \vee \bigcap_{i=1}^k \overline{y_i} = 0$;

— не больше одной единицы в наборе $\{a_1, a_2, \dots, a_k\}$, или $\bigcup_{i=1}^{k-1} \bigcup_{j=i+1}^k y_i \wedge y_j = 0$;

— хотя бы одна единица в наборе $\{a_1, a_2, \dots, a_k\}$, или $\bigcap_{i=1}^k \overline{y_i} = 0$.

Аналогичным образом задаются формулировки и ограничения, если в утверждении о числе единиц в наборе фигурирует произвольное число из диапазона $2, \dots, k$.

В процессе построения булевых моделей для ряда классических и прикладных дискретных задач были выявлены некоторые закономерности в формировании множества Y , определяемые спецификой задачи. Эти результаты привели к включению в методику моделирования ИК РЕБУС двух способов, автоматизирующих перечисление элементов множества $Y \subset X$.

Регулярный способ: множество Y , над элементами которого задаются булевы ограничения, может быть получено с помощью применения индексного триплета [Бартеньев, 1998] к массиву булевых переменных (частным случаем применения индексного триплета являются строки и столбцы двумерного массива). Другими словами, индексы элементов, включаемых в это множество, подчиняются определенной рекуррентной зависимости.

Нерегулярный способ: множество Y содержит произвольные элементы массива булевых переменных (в том смысле, что их индексы не связаны рекуррентной зависимостью), обладающие некоторой общей характеристикой. Для включения в множество Y таких элементов было введено понятие селекторного индекса, который задает последовательность индексов тех элементов, характеристики которых отвечают некоторому условию. Для задания подмножества булевых переменных в нерегулярных задачах требуется дополнительная информация, которая задается во вспомогательном вещественном или целочисленном массиве, характеризующем содержательную постановку задачи. Этот массив называется далее селекторным и содержит данные, позволяющие получить некоторое подмножество массива булевых переменных с помощью применения селекторного индекса. Например, в динамических задачах селекторный массив определяет ограничения для следующего шага.

Программная реализация этих двух способов построения булевой модели предполагает представление множества булевых переменных в виде матрицы булевых переменных V размерности $t \times n$, где t и n зависят от конкретной задачи. Для задач, требующих в дополнение к первому способу применения второго способа построения (нерегулярный), вводится дополнительная матрица W размерности $t \times k$, называемая далее *селекторной*, где k зависит от конкретной задачи. Эта матрица содержит данные (целочисленные значения), позволяющие получить некоторое подмножество (сечение) матрицы булевых переменных с помощью применения селекторного индекса. Обычный индексный триплет задает последовательность индексов, в которой первый элемент равен нижней границе, а каждый последующий больше (меньше) предыдущего на величину шага. Селекторный индекс задает последовательность индексов тех элементов, для которых выполняется некоторое условие над значениями, заданными в селекторной матрице W . К нерегулярным задачам относятся, например, динамические задачи, селектор W которых содержит ограничения для последующего шага.

Обозначим через $I_k^c = \{i \mid w_{i,k} = c\}$, $k = \overline{1, n}$ упорядоченное множество индексов строк матрицы W , соответствующих индексам столбцов матрицы V , где c – целая положительная ненулевая константа (в большинстве случаев $c = 1$, в этом случае верхний индекс можно не указывать); через v_{rk} – текущий элемент матрицы V . Тогда можно перечислить следующие варианты формирования множества Y :

– в множество Y входят элементы столбца с индексом k и элементы столбцов V_j' матрицы V , $j \in I_k^c$;

– в множество Y входят текущий элемент v_{rk} матрицы V и элементы следующей строки $v_{r+1,p}$, $p \in I_k^c$;

– в множество Y входят элементы i -й строки матрицы V v_{ij} , $j \in I_k^c$;

– в множество Y входят элементы, пара индексов которых включены в селекторный индекс: $Y = \{v_{ij} \mid (i, j) \in I^c\}$, $I^c = \{(i, j) \mid w_{ij} = c\}$. В этом случае указывается только верхний индекс множества I .

Для генерации ограничений в подсистеме моделирования в режиме конструктора достаточно выбрать содержательную формулировку и применить ее к заданному описанными выше способами множеству Y .

Пример применения технологии моделирования ИК РЕБУС для построения булевой модели прикладной дискретной задачи

Последнее десятилетие ознаменовалось широким внедрением вычислительной техники на железнодорожном транспорте и в локомотивном хозяйстве в частности. Наиболее важным для управления инфраструктурой локомотивного хозяйства является локомотивное депо, главная задача которого – своевременная выдача локомотивов и локомотивных бригад под поезд [Лакин и др., 2002]. От устойчивой работы локомотивного депо в значительной степени зависит устойчивая работа дороги в целом. Оперативное управление в эксплуатационном процессе локомотивного хозяйства на уровне депо начинается с составления графика оборота. Для этого используется информация о готовности мотор-вагонного подвижного состава, расписания движения поездов и информация об окне в поездном графике. Затем назначаются бригады на работу. Одной из задач, возникающих в оперативном управлении, является задача составления оперативного плана отправления локомотивных бригад в условиях отсутствия жесткого расписания движения грузовых поездов. Постановка задачи для m бригад и n поездов железнодорожного узла требуется составить график привязки бригады к поезду так, чтобы обеспечить k пар «бригада – поезд» при условии, что прогнозируемое время отправки поезда попадает в допустимый, согласно режиму труда и отдыха, интервал рабочего времени бригады.

Согласно изложенной выше классификации данная задача является динамической. В качестве массива булевых переменных выбрана матрица X размерности $N \times k$, где N – количество ребер двудольного графа, одна доля которого представляет бригады, другая – поезда, а ребро показывает возможность составления пары; $x_{t,j} = 1$ обозначает истинность факта, что выбрано j -е ребро в качестве t -го элемента списка пар длины k .

Ограничения:

– должно быть выбрано только одно ребро в качестве элемента списка, т. е. в каждой строке матрицы X должна быть только одна единица. Это ограничение выражается уравнением

$$\bigcup_{t=1}^k \left[\bigcup_{j=1}^{N-1} \bigcup_{p=j+1}^N x_{t,j} \wedge x_{t,p} \vee \bigcap_{j=1}^N \bar{x}_{t,j} \right] = 0; \quad (5)$$

– каждое ребро может быть выбрано не больше одного раза, т. е. в каждом столбце матрицы X должно быть не больше одной единицы:

$$\bigcup_{j=1}^N \bigcup_{t=1}^{k-1} \bigcup_{i=t+1}^k x_{t,j} \wedge x_{i,j} = 0; \quad (6)$$

— условие упорядоченности списка пар (ребра графа выбираются в качестве элементов списка пар в порядке возрастания индекса ребра, т. е. j -е ребро, выбранное на t -м шаге, будет связано отношением «не больше одной единицы» со всеми ребрами, индексы которых меньше либо равны j для $t + 1$ -го шага):

$$\bigcup_{t=1}^{k-1} \bigcup_{j=t+1}^N \bigcup_{p=1}^{j-1} x_{t,j} \wedge x_{t+1,p} = 0; \quad (7)$$

— выбранные ребра не имеют общих вершин (т. е. выбранное ребро будет связано отношением «не больше одной единицы» со всеми ребрами, имеющими с ним общую вершину и задаваемыми с помощью селекторной матрицы):

$$\bigcup_{j=1}^N \left[\bigcup_{p=1}^{|I_j|} \bigcup_{t=1}^N \bigcup_{l \in I_p} x_{t,j} \wedge x_{t,l} \right] = 0. \quad (8)$$

В формулах (5)–(7) ограничения задаются регулярным способом. В формуле (8) ребра, имеющие общую вершину с j -м ребром, задаются с помощью селекторного индекса I_j , полученного посредством использования селекторного массива, являющегося матрицей смежности для данной задачи.

Очевидно, что приведенные формулы позволяют получить булеву модель задачи в виде ДНФ или в двойственной ей КНФ, что позволяет применить для решения этой задачи не только решатели РЕБУС, но и любые известные SAT-решатели, ориентированные на формат DIMACS.

Заключение

Проведение вычислительных экспериментов показало, что применение представленных в статье методов и инструментальных средств автоматизации программирования в булевых ограничениях позволяет сократить трудозатраты на разработку, отладку и исследование булевых моделей, повышает качество и сокращает сроки выполнения необходимых многовариантных расчетов. При проведении вычислительных экспериментов в качестве тестов были использованы классические и практические дискретные задачи, а также тесты из библиотеки SATLIB⁴.

Понятно, что высокая вычислительная сложность решения задач в булевых ограничениях требует применения соответствующих методов и средств параллельных вычислительных технологий. С авторским подходом параллельного решения булевых уравнений в распределенной вычислительной среде можно ознакомиться в работах [Опарин и др., 2004; Опарин и др., 2006].

Список литературы

- Бартеньев О. В. Современный Фортран. М.: Диалог МИФИ, 1998. 397 с.
- Богданова В. Г. Методы и инструментальные средства программирования в булевых ограничениях: Дис. . . . канд. техн. наук. Иркутск, 2005. 137 с.
- Горшков С. П. Применение теории NP-полных задач для оценки сложности решения систем булевых уравнений // Обзорные прикладной и промышленной математики. Серия: Дискретная математика. 1995. Т. 2, вып. 3. С. 325–398.
- Лакин И. К., Смирнов Ю. В., Тимченко А. Ю. и др. Автоматизированная система управления локомотивным хозяйством. АСУТ Под ред. И. К. Лакина. М.: ОЦВ, 2002. 516 с.
- Левченков В. С. Булевы уравнения. М.: Диалог МГУ, 1999. 72 с.

⁴ <http://www.satlib.org>.

Опарин Г. А., Богданова В. Г. Алгоритмы решения больших разреженных систем булевых уравнений // Методы оптимизации и их приложения: Тр. XII Байкальской Междунар. конф. Секция 5: Дискретная математика. Иркутск: ИГУ, 2001. С. 114–118.

Опарин Г. А., Феоктистов А. Г., Богданова В. Г. и др. Инструментальная среда булева моделирования и решения булевых уравнений в сети Интернет // Распределенные вычисления и Грид-технологии в науке и образовании: Тр. Междунар. конф. Дубна: ОИЯИ, 2006. С. 369–378.

Опарин Г. А., Феоктистов А. Г., Богданова В. Г. и др. Решение булевых уравнений большой размерности в распределенной вычислительной среде // Распределенные вычисления и Грид-технологии в науке и образовании: Тр. Междунар. конф. Дубна: ОИЯИ, 2004. С. 164–169.

Ушаков Д. М., Телерман В. В. Системы программирования в ограничениях. // Системная информатика: Сб. науч. тр. Новосибирск: Наука, 2000. Вып. 7: Проблемы теории и методологии создания параллельных и распределенных систем. 312 с.

Материал поступил в редколлегию 22.03.2008

Г. А. Опарин, В. Г. Богданова

Rebus – Intellectual Solver for Combinatorial Problems in Boolean Constraints

The architecture, functionalities and applications of REBUS tools are described in our paper. This toolkit can be used for problem solving in various fields of knowledge where there are discrete models as systems of Boolean equations (constraints). Methods and facilities for automatization of representation, accumulation, modification and using of knowledge at the solving of Boolean constraint satisfaction problems underlie in base of REBUS-solver.

Keywords: Boolean equations and constraints, combinatorial problems, Boolean satisfiability.