

Применение технологии программирования в ограничениях для планирования действий в чрезвычайных ситуациях*

А. А. Зуенко, С. Ю. Яковлев, А. С. Шемякин, Ю. А. Олейник

Институт информатики и математического моделирования – обособленное подразделение Федерального государственного бюджетного учреждения науки Федерального исследовательского центра «Кольский научный центр Российской академии наук», г. Апатиты, Россия

Аннотация. Разработана технология интеллектуального планирования, ориентированная на исследование слабо формализованных предметных областей, знания о которых носят количественный и качественный характер. Технология обеспечивает поддержку открытой для оперативных модификаций модели предметной области, допускающей включение/исключение ограничений, критериев качества, а также задание начального и целевого состояний с помощью недоопределенных параметров. Задачу интеллектуального планирования предлагается ставить и решать в рамках объектно-ориентированного расширения технологии программирования в ограничениях, что предъявляет повышенные требования к эффективности обработки качественных ограничений. Качественные ограничения предложено представлять в виде специализированных матрицеподобных структур, а их обработку осуществлять с помощью авторских методов удовлетворения нечисловых ограничений. Предложенный подход позволяет структурировать семантически тесно связанные количественные и качественные ограничения, упрощая их сопровождение, а также ускорить их автоматическую генерацию и обработку. В качестве примера прикладной задачи в работе рассматривается упрощенный вариант задачи планирования действий при локализации территориального разлива нефтепродуктов.

Ключевые слова: технология программирования в ограничениях, чрезвычайные ситуации, планирование действий сил.

DOI 10.14357/20718632190103

Введение

Интеллектуальное планирование – это область искусственного интеллекта, которая в настоящее время вызывает значительный интерес [1]. Следует различать два направления исследований [2]: интеллектуальное планирование (AI-planning) и составление расписаний (scheduling). В настоящей работе будем придерживаться точки зрения, принятой в сообществе искусственного интел-

лекта, что задача составления расписаний является одной из разновидностей задач интеллектуального планирования.

Интеллектуальное планирование заключается в отыскании действий, необходимых для решения задачи, и выстраивании этих действий в последовательность, приводящую к решению.

Среди базовых методов интеллектуального планирования следует выделить два направления: методы поиска, основанные на использо-

*Работа выполнена при частичной финансовой поддержке РФФИ (проекты №№ 16-07-00377а, 18-07-00615а, 16-07-00273а, 18-07-00167а).

вании графа планирования [3]; методы планирования на основе логических рассуждений [4].

В последнее время развивается подход к решению задач планирования в рамках технологии программирования в ограничениях [5, 6], однако интенсивность подобных исследований довольно низкая.

Задача удовлетворения ограничений (CSP – Constraint Satisfaction Problem) – это тройка (V, D, C) [1], где:

- 1) $V = \{v_1, \dots, v_n\}$ – множество переменных;
- 2) $D = \{D_1, \dots, D_n\}$ – множество доменов; каждый домен D_i – конечное множество, содержащее возможные значения соответствующей переменной;
- 3) $C = \{C_1, \dots, C_m\}$ – множество ограничений.

Ограничение C_i – отношение, определённое на подмножестве значений всех переменных, т.е. $C_i \subseteq D_1 \times \dots \times D_n$.

Решением CSP-задачи является такая подстановка для всех переменных, при которой все ограничения удовлетворены. В некоторых случаях необходимо получить все решения. Иногда требуется найти такое решение, в котором значения переменных оптимизировали бы некоторый заданный функционал. CSP-задачи принадлежат классу NP-полных задач [1].

Программирование в ограничениях как технология решения задач комбинаторного поиска и комбинаторной оптимизации находит широкое применение в задачах составления расписаний [7-9], однако в более общих задачах интеллектуального планирования ее применение все еще весьма ограничено. По мнению авторов, данная технология в области интеллектуального планирования не получила должного внимания, а многие методы планирования используют методы SAT [4]. При разложении CSP и представлении ее в форме задачи SAT теряется часть информации о структуре задачи, которую можно использовать для ускорения процедуры вывода.

Актуальность исследований, представленных в статье, определяется важностью моделирования слабо формализованных предметных областей, знания о которых разнородны, носят как количественный, так и качественный характер. Качественные соотношения (многочленные отношения, логические формулы, экс-

пертные правила и т.п.), описывающие еще до конца не изученные аспекты функционирования системы, дополняют аналитические закономерности, уже известные для полностью изученных фрагментов системы. Совместная обработка подобных ограничений затруднена, поскольку приходится привлекать различные математические аппараты. К тому же размерность решаемых задач может быть достаточно большой, что обуславливает применение специализированных методов ускорения вычислительных процедур. Архитектура систем программирования в ограничениях позволяет унифицировать совместную обработку количественных и качественных ограничений. Однако в большинстве существующих систем программирования в ограничениях обработка качественных ограничений реализуется недостаточно эффективно.

Решение практических задач планирования осложнено тем, что модель задачи формируется поэтапно, постепенно уточняясь новыми деталями. В процессе создания модели в нее могут включаться типы зависимостей, критерии оптимизации, которые не могли быть априорно учтены. С другой стороны, если в некоторой постановке решения задачи CSP не существует, должна быть обеспечена возможность “ослабить” задачу, исключив из рассмотрения некоторые ограничения и/или критерии оптимизации. В этих условиях применение традиционных методов теории исследования операций представляется проблематичным.

К числу задач, обладающих перечисленными особенностями, относится и рассматриваемая в работе задача планирования действий в чрезвычайных ситуациях.

Существенным недостатком современных инструментов программирования в ограничениях является невозможность специфицировать и решать задачи относительно переменных, являющихся сложно структурированными, семантически согласованными данными, в частности, объектно-ориентированными. Оригинальность проведенных исследований состоит в том, что объектно-ориентированное представление используется для структуризации переменных соответствующей задачи CSP. Это позволяет упростить автоматическую генера-

цию и обработку больших объемов семантически тесно связанных ограничений. Объектно-ориентированное представление задачи CSP предъявляет высокие требования к эффективности обработки качественных ограничений. В работе предложено качественные ограничения представлять в виде специализированных матрицеподобных структур (*C*- и *D*-систем), а их обработку осуществлять с помощью авторских методов удовлетворения нечисловых ограничений.

Предлагаемая технология планирования позволяет обрабатывать случаи, когда все или некоторые параметры в описании начального или целевого состояний недоопределены, то есть значения данных параметров могут задаваться не в виде единственного значения, а в виде конечного множества альтернатив.

1. Задача планирования действий сил при локализации разливов нефтепродуктов

Совместное использование разнородных сил и средств, рациональная координация их действий во времени и пространстве – задача, возникающая во многих областях деятельности. Одно из актуальных направлений – планирование действий в чрезвычайных ситуациях (ЧС).

В качестве примера рассмотрим задачу разработки плана по предупреждению и ликвидации разливов нефти и нефтепродуктов (ПЛРН) [10]. Отметим, что информационно-аналитическое обеспечение разработки ПЛРН следует признать неудовлетворительным [11].

В настоящей работе будут рассматриваться разливы нефтепродуктов на суше. Решение данной задачи может заключаться в составлении плана действий, который минимизирует общее время, необходимое для выполнения всех работ (действий), при соблюдении ограничений на ресурсы. Возможны и другие постановки, например, требуется составить план, который должен обеспечить заданные стоимостные характеристики моделируемого процесса и при этом сам процесс должен укладываться в определенные временные рамки.

Разработку наиболее «научкоёмкой», аналитической части ПЛРН можно разбить на этапы.

1. Определение возможных сценариев аварий.

2. Расчёт параметров сценариев: вероятность (частота), объём нефтепродукта, параметры (форма и размеры) зоны поражения, ущерб (социальный, экономический, экологический).

3. Локализация аварии. Определяется порядок действий, производится расчёт достаточности сил и средств, обеспечивающих локализацию (фиксацию, нераспространение) аварии на местности не более чем за 6 часов.

4. Ликвидация аварии (сбор нефти и грунта, тушение пожара).

5. Ликвидация последствий, восстановление работоспособности, рекультивация земель.

Таким образом, разработка ПЛРН осуществляется при наличии многих ограничений. Учитываются особенности объекта и его окружения (организационная структура, технологические процессы, рельеф местности и инфраструктура), нормативные требования, типовые сценарии аварий, технологии локализации и ликвидации, оснащённость и дислокация сил. Возможны различные критерии оптимизации и формы представления результатов планирования.

Особенности применения технологии программирования в ограничениях к задаче распределения разнородных сил и средств поясним на упрощенном примере задачи локализации территориального разлива нефтепродуктов.

В качестве опасного объекта выбран резервуар для хранения мазута объёмом 3000 м^3 . В резервуаре производится поддержание необходимой температуры мазута (не ниже 75°C), его перемешивание. Резервуар входит в состав склада горюче-смазочных материалов (ГСМ). Склад находится на ровной местности, в промзоне, населённых пунктов и особых территорий поблизости нет. На Рис. 1 приведена условная схема расположения основных пунктов окружающей территории, значимых для локализации ЧС, соединительными стрелками обозначены дороги, цифрами – длина дорог.

Рассматривается следующий возможный сценарий аварии – квазимгновенное разрушение резервуара и разлив мазута. Площадь разлива может составить до 60000 м^2 . Частота такой аварии оценивается величиной $3,8 \cdot 10^{-5} \text{ 1/год}$. Взрыв и пожар не рассматриваются.

Для локализации аварии могут быть применены различные технологии:

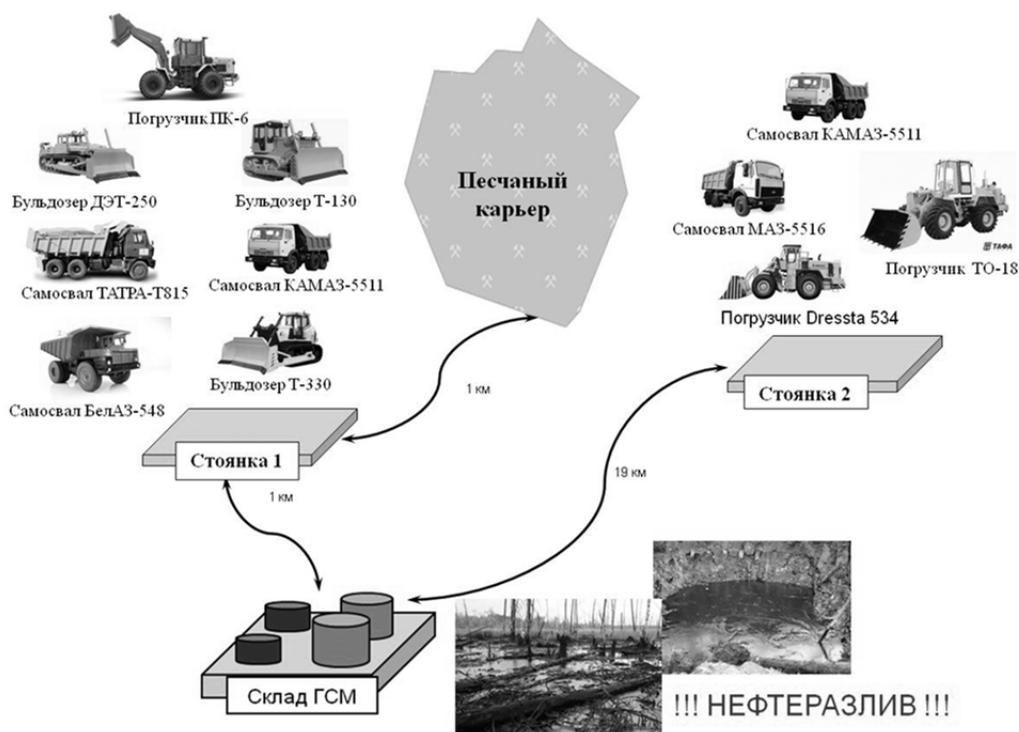


Рис. 1. Ситуационный план окружающей территории

- ограждение при помощи дамбы;
- применение подпорной стенки;
- применение сорбентов.

В данном примере рассматривается технология локализации с помощью ограждающей дамбы. Технология предъявляет требования к типам техники, которые задействованы в процессе локализации аварии, а также возможным типам действий, которые может выполнять данная техника. Технология накладывает ограничения на порядок выполнения тех или иных действий, время их выполнения. Дадим описание технологии локализации с помощью ограждающей дамбы. В качестве материала (чистого грунта) для дамбы используется песок из карьера.

Сначала определим типы участников процесса локализации. Под участниками понимаются технические средства, задействованные в процессе локализации. В рамках задачи рассматриваются три различных типа участников:

1. самосвалы, которые перевозят песок для строительства дамбы;
2. погрузчики, которые загружают песок в самосвалы;
3. бульдозеры, которые строят дамбу из привезенного песка.

Все участники способны совершать определенный набор действий. Среди них есть как общие действия для всех (перемещение между точками, ожидание), так и специфические, зависящие от типа участника (погрузка – для погрузчика, строительство дамбы – для бульдозера и т.п.).

Территория представляется в виде совокупности точек местности и дорог между ними. С помощью абстракции «Точка местности» представляются значимые для процесса локализации пункты дислокации и т.п.

Всего можно выделить три типа точек местности:

1. стоянки, откуда участники начинают процесс локализации, а также место, где они его заканчивают;
2. карьер, где берется песок для возведения дамбы;
3. место разлива, куда самосвалы привозят песок и где бульдозеры строят дамбу.

Для локализации разлива мазута с использованием ограждающей дамбы необходимо выполнить следующие действия:

1. доставка бульдозеров к месту разлива своим ходом;

2. доставка погрузчиков к источнику чистого грунта своим ходом;

3. доставка самосвалов к источнику чистого грунта своим ходом;

4. перевозка чистого грунта от карьера к месту разлива. Для выполнения этого действия требуются погрузчики и самосвалы. Перевозка грунта включает в себя несколько действий: погрузка самосвала в карьере, собственно транспортировка грунта, выгрузка грунта в месте разлива;

5. формирование дамбы вокруг разлива при помощи бульдозеров;

6. доставка техники в пункты постоянной дислокации – своим ходом.

Далее перейдем к детальному описанию способа формализации задачи планирования мероприятий по локализации разлива нефтепродуктов в виде задачи CSP.

2. Классификация ограничений, служащих для описания задачи

Ограничения рассматриваемой задачи CSP, среди которых встречаются как количественные, так и качественные зависимости, можно условно разбить на несколько классов.

1. *Ограничения, описывающие критерии оптимизации.* Например, затраты на локализацию не должны превышать 2 млн рублей или локализация должна быть завершена в течение 6 часов с момента поступления информации о разливе. В настоящей работе предлагается подобные ограничения представлять в виде глобальных ограничений [1], что существенно ускоряет скорость процесса решения задачи CSP. Глобальное ограничение представляет собой набор более простых ограничений, при рассмотрении которых в совокупности удается разработать более эффективную процедуру их удовлетворения по сравнению с алгоритмами общего назначения. У глобальных ограничений есть два преимущества. Во-первых, они облегчают задачу моделирования прикладной проблемы в виде задачи CSP. Вместо записи задачи с помощью многих исходных бинарных ограничений ее удастся описать с помощью небольшого числа глобальных ограничений для множеств переменных. Во-вторых, для глобальных ограничений разрабатываются алго-

ритмы распространения ограничений специального вида и поэтому более эффективные.

2. *Ограничения, которые продиктованы особенностями применяемой технологии локализации аварии.* Например, один погрузчик не может загружать два самосвала одновременно, формирование дамбы может вестись одновременно с доставкой грунта и т.п. Также к этой группе ограничений относятся ограничения перехода между этапами плана, которые составляют существенную часть ограничений рассматриваемой задачи CSP. Эти ограничения описывают переходы между i и $i+1$ этапами плана, задают допустимые последовательности действий в зависимости от типа участников и типа точек местности.

3. *Ограничения, характеризующие конкретные экземпляры или подтипы участников процесса локализации.* Например: самосвалы определенной марки могут загружаться только погрузчиками строго определенных марок; машина определенной марки может перемещаться между заданными пунктами со скоростью меньше расчетной и т.п.

3. Формальный аппарат для представления и обработки качественных зависимостей

Объектно-ориентированное представление задачи планирования предъявляет повышенные требования к обработке качественных ограничений. Существующие на данный момент в теории удовлетворения ограничений методы удовлетворения нечисловых ограничений с конечными доменами недостаточно эффективны, поэтому для представления качественных ограничений предметной области в настоящей работе предлагается использовать два типа матрицеподобных структур: C -системы и D -системы [12, 13].

Например для CSP, где $X = \{X_1, X_2\}$ и $D_1 = D_2 = \{a, b, c\}$, C -система

$$C_1[X_1X_2] = \begin{bmatrix} \{a\} & \{b,c\} \\ \{b\} & \{a,c\} \\ \{c\} & \{a,b\} \end{bmatrix}$$

является представлением ограничения, описывающего тот факт, что значения переменных X_1 и X_2 не совпадают.

D -системы позволяют легко вычислять дополнение C -систем, для чего берется дополнение для каждой компоненты-множества. Так, ограничение $X_1 = X_2$ можно выразить D -системой

$$D_1 = \bar{C}_1[X_1 X_2] = \begin{bmatrix} \{b, c\} & \{a\} \\ \{a, c\} & \{b\} \\ \{a, b\} & \{c\} \end{bmatrix}.$$

При задании C -систем может использоваться обозначение “*” (полная компонента), которое равносильно указанию всего домена соответствующей переменной, а также “∅” (пустая компонента), обозначающая, что компонента не содержит ни одного значения.

В работах [12, 13] предложены алгоритмы распространения нечисловых ограничений, опирающиеся на эквивалентные преобразования C - и D -систем. Преобразования позволяют исключать значения из доменов атрибутов или сами столбцы-атрибуты, а также исключать из рассмотрения «лишние» строки, в результате чего сокращается пространство поиска решения.

Для реализации авторских алгоритмов вывода на ограничениях, представленных в виде специализированных матрицеподобных структур, была выбрана библиотека Choco.

Проведенные исследования продемонстрировали, что применение предложенного подхода к обработке качественных ограничений дает существенный выигрыш времени по сравнению со встроенными в библиотеку Choco алгоритмами распространения нечисловых ограничений [14]. В качестве одной из тестовых была выбрана задача N-Queens, ввиду удобства ее масштабирования. Когда требовалось найти все решения задачи N-Queens с лимитом по времени в две минуты, предложенные методы справились с пространством поиска размерности 12^{12} . При этом, использование стандартных средств Choco не дало возможности исследовать пространство поиска более 9^9 . В случае, когда ставилась задача найти хотя бы одно решение за те же две минуты, стандартные средства Choco переставали укладываться в отведенное время уже при размерности пространства поиска, оцениваемого числом 19^{19} . Предлагаемые же авторами методы справились с пространством поиска размерности 76^{76} .

4. Представление задачи в рамках парадигмы программирования в ограничениях

В основе объектно-ориентированных моделей лежат качественные отношения, в частности, отношения иерархии “часть-целое”, “класс-подкласс” и т.д. В то же время простые атрибуты объектов (листовые элементы иерархии) могут быть как числовыми, так и нечисловыми. Следовательно, и соотношения на простых атрибутах могут носить характер как числовых, так и нечисловых зависимостей. Объектно-ориентированный подход является естественным инструментом структуризации знаний о слабо-формализованной предметной области и стратификации разнородных количественных и качественных ограничений. Однако несмотря на привлекательность идеи синтеза парадигмы программирования в ограничениях и объектно-ориентированного подхода, до сих пор не существует единого понимания, какие конструктивные очертания она может приобрести при дальнейшей проработке и развитии.

Новизна исследований состоит в том, что задачу интеллектуального планирования предлагается ставить и решать в рамках объектно-ориентированного расширения технологии программирования в ограничениях в виде задачи удовлетворения ограничений со сложно структурированными доменами переменных.

Далее приводятся описание программных классов, используемых при постановке задачи планирования, а также примеры реализации ограничений.

Все участники процесса локализации являются потомками абстрактного класса «Участник процесса локализации», включающего интерфейс для самосвала, погрузчика и бульдозера.

Для описания характеристик местности разработаны два программных класса: класс “Точка местности”, описывающий каждую значимую для построения плана точку местности, и класс “Дорога”, описывающий связь двух соседних точек местности.

Класс “Участник процесса локализации” содержит поле *status* – список переменных, обозначающий, какое действие выполняет участник на каждом этапе плана. Области

определения этих переменных состоят из экземпляров класса «Действие». Действия могут быть либо общими для всех типов участников, либо зависеть от типа участника и типа точки местности, где они выполняются. Таким образом, действия можно разделить на ряд подклассов (Рис. 2).

Рассмотрим простейшую ситуацию, когда на план территории нанесены только две точки местности: $point_1$ – точка, являющаяся одновременно карьером и стоянкой, и $point_2$ – место разлива. Также имеется дорога $path$ между ними. На территории находятся один самосвал tr и один погрузчик ld . В такой постановке задачи для построения возможных действий будут задействованы подклассы:

$start$ – подкласс начала работы, на основе которого строятся конкретные действия-экземпляры st_1^{ld} и st_1^{tr} – действия начала работы в точке $point_1$ для ld и tr , соответственно;

$stop$ – подкласс завершения работы, на основе которого строятся конкретные действия sp_1^{ld} и sp_1^{tr} – действия окончания работы в точке $point_1$;

$move$ – перемещение участника из одной точки в другую точку, на основе которого строятся m_{12}^{ld} , m_{21}^{ld} , m_{12}^{tr} , m_{21}^{tr} – действия перемещения участников ld и tr между точками $point_1$ и $point_2$ в ту и другую сторону;

$wait$ – подкласс ожидания участника в точке, на основе которого строятся w_1^{ld} , w_2^{ld} , w_1^{tr} , w_2^{tr} – действия ожидания в точках $point_1$ и $point_2$, соответственно.

При загрузке погрузчиком ld песка в самосвал tr в точке 1 строятся два действия: l_{tr} – действие погрузчика, f_{ld} – действие самосвала.

Далее более подробно остановимся на особенностях представления ограничений в рамках предложенного подхода, то есть с помощью матрицеподобных структур.

Введем переменные:

X_i^{ld} – действие, осуществляемое погрузчиком ld на i -ом этапе плана (соответствует программной структуре $ld.status[i]$);

X_{i+1}^{ld} – действие, осуществляемое погрузчиком ld на $(i+1)$ -ом этапе плана (соответствует программной структуре $ld.status[i+1]$);

Виды точек участников	Стоянка	Карьер	Место разлива (дамба)
Самосвал	•начать работу •закончить работу	•загрузиться	•разгрузиться
Погрузчик	•начать работу •закончить работу	•загрузить	нет особых действий
Бульдозер	•начать работу •закончить работу	нет особых действий	•строить дамбу

Общие действия для любой пары участник-точка:

- поехать в соседнюю точку
- приехать из соседней точки
- ожидать в текущей точке

Рис. 2. Классификация действий

T_i^{ld} – время начала действия X_i^{ld} (соответствует программной структуре $ld.status[i].time$);

T_{i+1}^{ld} – время начала действия X_{i+1}^{ld} (соответствует программной структуре $ld.status[i+1].time$);

S_i^{ld} – продолжительность действия X_i^{ld} (соответствует программной структуре $ld.status[i].duration$);

X_i^{tr} – действие, осуществляемое самосвалом tr на i -ом этапе плана (соответствует программной структуре $tr.status[i]$);

T_i^{tr} – время начала действия X_i^{tr} (соответствует программной структуре $tr.status[i].time$);

S_i^{tr} – продолжительность действия X_i^{tr} (соответствует программной структуре $tr.status[i].duration$).

Доменами переменных X_i^{ld} , X_{i+1}^{ld} является множество действий погрузчика ld : $\{st_1^{ld}, sp_1^{ld}, m_{12}^{ld}, m_{21}^{ld}, w_1^{ld}, w_2^{ld}, l_{tr}\}$. Доменом переменной X_i^{tr} является множество $\{st_1^{tr}, sp_1^{tr}, m_{12}^{tr}, m_{21}^{tr}, w_1^{tr}, w_2^{tr}, f_{ld}\}$. Домены остальных переменных состоят из целочисленных значений, обозначающих моменты времени – этапы процесса моделирования. Допустим, что этап плана длится $Steptime$ минут.

В качестве примера рассмотрим два ограничения, накладывающие условия на начало нового действия в первой точке:

$$I. ((X_i^{ld} = w_1^{ld}) \vee (X_i^{ld} = st_1^{ld}) \vee (X_i^{ld} = m_{21}^{ld}) \vee (X_i^{ld} = l_{tr})) \wedge (T_i^{ld} \leq Steptime * i - S_i^{ld}) \leftrightarrow ((X_{i+1}^{ld} = m_{12}^{ld}) \vee (X_{i+1}^{ld} = w_1^{ld}) \vee (X_{i+1}^{ld} = sp_1^{ld}) \vee (X_{i+1}^{ld} = l_{tr})) \wedge (T_{i+1}^{ld} = Steptime * i).$$

$$\text{II. } (X_{i+1}^{ld} = l_{tr}) \rightarrow ((X_i^{tr} = st_1^{tr}) \vee (X_i^{tr} = m_{21}^{tr}) \vee (X_i^{tr} = w_1^{tr})) \wedge (T_i^{tr} \leq Steptime * i - S_i^{tr}).$$

В ограничении I перечисляются все действия, которые могли завершиться на первой стоянке $\{st_1^{ld}, m_{21}^{ld}, w_1^{ld}, l_{tr}\}$ на i -ом шаге плана и указывается, что если они завершились, то на $(i+1)$ -ом этапе плана этап начнется выполнение какого-либо из действий $\{sp_1^{ld}, m_{12}^{ld}, w_1^{ld}, l_{tr}\}$. Ограничение II уточняет условия начала действия l_{tr} .

В таком виде ограничения обрабатываются недостаточно эффективно большинством современных систем программирования в ограничениях. Проиллюстрируем, как представляются данные ограничения с использованием предлагаемого аппарата нечисловых матриц. Для этого перепишем имеющиеся ограничения, вводя вспомогательные переменные булевского типа:

- 1) $B_i^{ld} \leftrightarrow (T_i^{ld} \leq Steptime * i - S_i^{ld});$
- 2) $B_{i+1}^{ld} \leftrightarrow (T_{i+1}^{ld} = Steptime * i);$
- 3) $((X_i^{ld} = w_1^{ld}) \vee (X_i^{ld} = st_1^{ld}) \vee (X_i^{ld} = m_{21}^{ld}) \vee (X_i^{ld} = l_{tr})) \wedge B_i^{ld} \leftrightarrow ((X_{i+1}^{ld} = m_{12}^{ld}) \vee (X_{i+1}^{ld} = w_1^{ld}) \vee (X_{i+1}^{ld} = sp_1^{ld}) \vee (X_{i+1}^{ld} = l_{tr})) \wedge B_{i+1}^{ld};$
- 4) $B_i^{tr} \leftrightarrow (T_i^{tr} \leq Steptime * i - S_i^{tr});$
- 5) $(X_{i+1}^{ld} = l_{tr}) \rightarrow ((X_i^{tr} = st_1^{tr}) \vee (X_i^{tr} = m_{21}^{tr}) \vee (X_i^{tr} = w_1^{tr})) \wedge B_i^{tr}.$

Первые 3 из вновь полученных ограничений соответствуют ограничению I. Ограничения 4 и 5 соответствуют ограничению II. Ограничение 3 может быть выражено следующей D -системой:

$$C_1[X_i^{ld} B_i^{ld} X_{i+1}^{ld} B_{i+1}^{ld}] = \begin{bmatrix} \{w_2^{ld}, m_{12}^{ld}, sp_1^{ld}\} & \{0\} & \{w_1^{ld}, sp_1^{ld}, m_{12}^{ld}, l_{tr}\} & \emptyset \\ \{w_2^{ld}, m_{12}^{ld}, sp_1^{ld}\} & \{0\} & \emptyset & \{1\} \\ \emptyset & \{1\} & \{w_2^{ld}, m_{21}^{ld}, st_1^{ld}\} & \{0\} \\ \{w_1^{ld}, st_1^{ld}, m_{21}^{ld}, l_{tr}\} & \emptyset & \{w_2^{ld}, m_{21}^{ld}, st_1^{ld}\} & \{0\} \end{bmatrix}.$$

Ограничение 5 может быть записано в виде такой D -системы:

$$C_2[X_{i+1}^{ld} X_i^{tr} B_i^{tr}] = \begin{bmatrix} \{st_1^{ld}, sp_1^{ld}, m_{12}^{ld}, m_{21}^{ld}, w_1^{ld}, w_2^{ld}\} & \{st_1^{tr}, m_{21}^{tr}, w_1^{tr}\} & \emptyset \\ \{st_1^{ld}, sp_1^{ld}, m_{12}^{ld}, m_{21}^{ld}, w_1^{ld}, w_2^{ld}\} & \emptyset & \{1\} \end{bmatrix}.$$

Поясним, как формируются описанные выше D -системы на примере ограничения 5. Ограничение 5 может быть записано в виде:

$$(\neg(X_{i+1}^{ld} = l_{tr}) \vee ((X_i^{tr} = st_1^{tr}) \vee (X_i^{tr} = m_{21}^{tr}) \vee (X_i^{tr} = w_1^{tr}))) \wedge (\neg(X_{i+1}^{ld} = l_{tr}) \vee B_i^{tr}).$$

Каждая из “перемножаемых” скобок соотносится с соответствующей строкой D -системы $C_2[X_{i+1}^{ld} X_i^{tr} B_i^{tr}]$.

Подобное представление изначально заданных ограничений I и II позволяет каждое из вновь полученных ограничений 1-5 обрабатывать специализированными высокоэффективными алгоритмами-пропагаторами. Ограничения 1, 2, 4 обрабатываются с использованием алгоритмов распространения числовых ограничений. Ограничения 3, 5 обрабатываются с помощью авторских методов удовлетворения нечисловых ограничений.

Теперь введем в рассмотрение еще одно ограничение. Заметим, что в рамках парадигмы программирования в ограничениях исходная задача решается как серия задач CSP. Сначала, исходя из предварительных оценок, предполагается, что задача CSP может быть решена за определенное количество этапов (n). Если за это количество этапов решение не получено, то строится предположение, что задачу можно решить за $n+1$ этап и в задачу CSP добавляются ограничения, соответствующие еще одному этапу планирования. Естественным ограничением в этом случае является запрет на каждом из этапов планирования на выполнение действий, которые заведомо не могут быть завершены к концу моделируемого процесса.

Рассмотрим, как выглядит это ограничение для действий X_i^{ld} , осуществляемых погрузчиком ld :

$$\forall i X_i^{ld} .time + X_i^{ld} .duration \leq Steptime * n.$$

Запись $X_i^{ld} .time$ – обозначает время начала любого из действий, осуществляемых погрузчиком ld на i -ом этапе планирования, $X_i^{ld} .duration$ – продолжительность действия X_i^{ld} .

Теперь выберем некоторый конкретный этап моделирования i . Если рассматривать вновь сформированное количественное ограничение в совокупности с качественными ограничениями,

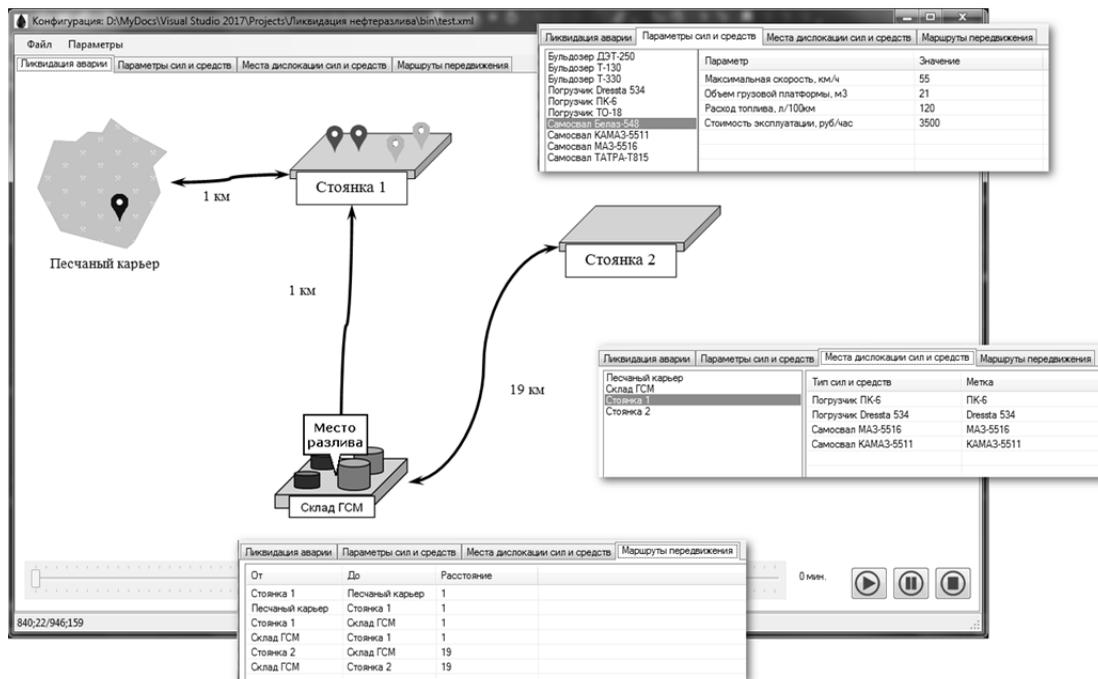


Рис. 3. Окна графического редактора

записанными в виде матриц $C_1[X_i^{ld} B_i^{ld} X_{i+1}^{ld} B_{i+1}^{ld}]$ и $C_2[X_{i+1}^{ld} X_i^{lr} B_i^{lr}]$, то можно заметить, что сужение доменов числовых переменных $X_i^{ld.time}$ и $X_i^{ld.duration}$ приводит к сужению домена нечисловой переменной X_i^{ld} , то есть исключению из рассмотрения некоторых экземпляров класса “Действие”, и, наоборот, – сокращение домена переменной X_i^{ld} ведет к редуцированию доменов упомянутых числовых переменных. Эти тесно семантически связанные переменные предлагается записывать в объектно-ориентированном стиле. Подобная запись позволяет скрыть манипуляции, направленные на сопровождение отношений “класс – свойства класса”: при создании экземпляра класса конструктором класса автоматически создается кортеж C -системы, ставящий в соответствие экземпляру класса значения его свойств; вся C -система соответствует ограничению “класс – свойства класса”.

5. Пользовательский интерфейс (графический редактор)

Внешний вид главного окна графического редактора приведен на Рис. 3.

Пользователь может указать файл карты-подложки, которая загружается в редактор. В результате действий пользователя формируется XML-файл конфигурации, который впоследствии используется в качестве исходных данных для библиотеки Choco. На основе переданной конфигурации при помощи Choco формируется план действий, который может быть представлен в табличной форме (Рис. 4).

На основе полученного плана действий воспроизводится анимация процесса локализации аварии в главном окне (Рис. 3).

Заключение

В отличие от традиционных методов интеллектуального планирования и методов теории исследования операций, предполагающих наличие полной и корректной информации, разработанная авторами программная технология ориентирована на поддержку открытой для оперативных модификаций модели предметной области, допускающей включение / исключение ограничений, критериев качества, а также задание начального и целевого состояний с помощью недоопределенных параметров.

Новизна исследований состоит в том, что задачу интеллектуального планирования для слабо

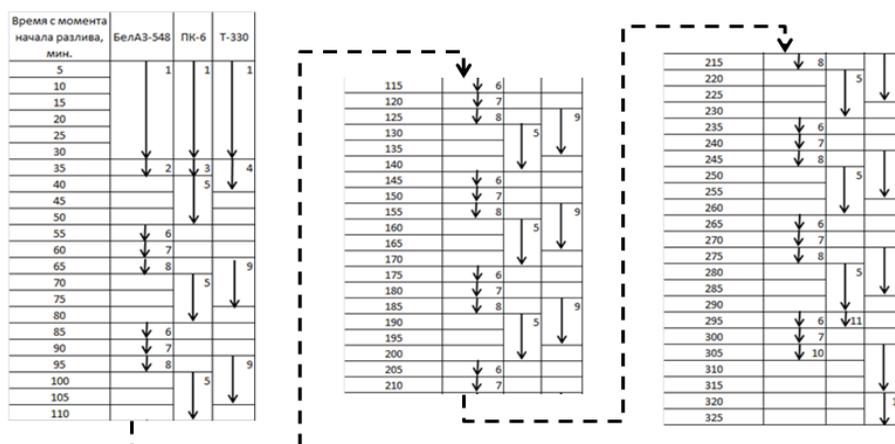


Рис. 4. Вариант решения

формализованных предметных областей предлагается ставить и решать в рамках объектно-ориентированного расширения технологии программирования в ограничениях в виде задачи удовлетворения ограничений со сложно структурированными доменами переменных. Использование объектно-ориентированного представления задачи CSP предъявляет повышенные требования к эффективности обработки качественных ограничений. Предложено качественные ограничения представлять в виде специализированных матрицеподобных структур (C- и D-систем), а их обработку осуществлять с помощью авторских методов удовлетворения нечисловых ограничений, что дает существенный выигрыш в производительности вычислений по сравнению с традиционными алгоритмами распространения табличных ограничений.

В результате, объектно-ориентированное представление задачи интеллектуального планирования позволяет структурировать семантически тесно связанные количественные и качественные ограничения, упрощая их сопровождение, а также ускорить их автоматическую генерацию и обработку.

Литература

1. Russel S., Norvig P. Artificial Intelligence: A Modern Approach. 3rd edition. Prentice Hall. 1132 p.
2. Smith D., Frank J., Jónsson A. Bridging the Gap Between Planning and Scheduling. // Knowledge Engineering Review. No. 15(1). 2000. P. 47-83.
3. Blum A., Furst M. Fast planning through planning graph analysis. // Artificial Intelligence. 1997. No. 90. P. 281-300.
4. Rintanen J. Planning and SAT. // Handbook of Satisfiability. IOS Press. 2009. P. 483-504.

5. Steger-Jensen K., Hvolby H.-H., Nielsen P., Nielsen I. Advanced planning and scheduling technology // Production Planning and Control. Vol. 22. No. 8. 2011. P. 800–808.
6. Bit-Monnot A. A Constraint-Based Encoding for Domain-Independent Temporal Planning // Proceedings of 24th International Conference «Principles and Practice of Constraint Programming» (CP 2018, Lille, France, August 27-31). P 30-46. doi: 10.1007/978-3-319-98334-9.
7. Margaux N., Artigues C., Lopez P. Cumulative scheduling with variable task profiles and concave piecewise linear processing rate functions // Constraints. Vol. 22. Issue 4. 2017. P. 530-547.
8. Kreter S., Rieck J., Zimmermann J. Models and solution procedures for the resource-constrained project scheduling problem with general temporal constraints and calendars. // European Journal of Operational Research. No. 251(2). 2016. P. 387-403.
9. Letort A., Carlsson M., Beldiceanu N. Synchronized sweep algorithms for scalable scheduling constraints // Constraints. Vol. 20. Issue 2. 2015. P 183-234. doi: 10.1007/s10601-014-9172-8.
10. Приказ МЧС России "Об утверждении Правил разработки и согласования планов по предупреждению и ликвидации разливов нефти и нефтепродуктов на территории РФ" от 28.12.04 № 621.
11. Яковлев С.Ю., Рыженко А.А., Исакевич Н.В. Разработка планов по предупреждению и ликвидации разливов нефтепродуктов для территориальных объектов // Моделирование и анализ безопасности и риска в сложных системах: Труды Международной научной школы МАБР-2010 (Санкт-Петербург, 6-10 июля, 2010 г.). СПб: ГУАП. 2010. С. 447-452.
12. Zuenko A. Matrix-like Structures for Representation and Processing of Constraints Over Finite Domains // Advances in Intelligent Systems and Computing (AISC). Vol. 875. Springer Nature Switzerland AG. 2019. P 428-438. doi: 10.1007/978-3-030-01821-4_45.
13. Зуенко А.А., Ломов П.А., Олейник А.Г. Применение методов распространения ограничений для ускорения обработки запросов к онтологиям // Труды СПИИРАН. № 1(50). 2017. С. 112-136. doi: 10.15622/sp.50.5

14. Zuenko A., Oleynik Y. Programming of Algorithms of Matrix-Represented Constraints Satisfaction by Means of Choco Library // Advances in Intelligent Systems and Computing (AISC). Vol. 875. Springer Nature Switzerland AG. 2019. P. 1-10. doi: 10.1007/978-3-030-01821-4_46.

Зуенко Александр Анатольевич. Институт информатики и математического моделирования – обособленное подразделение Федерального государственного бюджетного учреждения науки Федерального исследовательского центра «Кольский научный центр Российской академии наук» (ИИММ КНЦ РАН), г. Апатиты, Россия. Старший научный сотрудник, кандидат технических наук. Количество печатных работ: 150 (в том числе 2 монографии). Область научных интересов: программирование в ограничениях; моделирование слабо формализованных предметных областей. E-mail: zuenko@iimm.ru

Яковлев Сергей Юрьевич. Институт информатики и математического моделирования – обособленное подразделение Федерального государственного бюджетного учреждения науки Федерального исследовательского центра «Кольский научный центр Российской академии наук» (ИИММ КНЦ РАН), г. Апатиты, Россия. Старший научный сотрудник, кандидат технических наук, доцент. Количество печатных работ: 180. Область научных интересов: информационные технологии управления промышленно-экологической безопасностью. E-mail: Yakovlev@iimm.ru

Шемякин Алексей Сергеевич. Институт информатики и математического моделирования – обособленное подразделение Федерального государственного бюджетного учреждения науки Федерального исследовательского центра «Кольский научный центр Российской академии наук» (ИИММ КНЦ РАН), г. Апатиты, Россия. Младший научный сотрудник. Количество печатных работ: 40. Область научных интересов: информационные технологии управления промышленно-экологической безопасностью. E-mail: shemyakin@iimm.ru

Олейник Юрий Андреевич. Институт информатики и математического моделирования – обособленное подразделение Федерального государственного бюджетного учреждения науки Федерального исследовательского центра «Кольский научный центр Российской академии наук» (ИИММ КНЦ РАН), г. Апатиты, Россия. Младший научный сотрудник. Количество печатных работ: 10. Область научных интересов: программирование в ограничениях. E-mail: yoleynik@iimm.ru

Application of constraint programming technology for planning action in emergency situations

A. A. Zuenko, S. Yu. Yakovlev, A. S. Shemyakin, Yu. A. Oleynik

Institute for Informatics and Mathematical Modeling – Subdivision of the Federal Research Centre “Kola Science Centre of the Russian Academy of Sciences”, Apatity, Russia

Abstract. The authors developed a technology of AI planning, focused on the study of poorly formalized subject domains, knowledge about which is quantitative and qualitative. The technology provides support for the domain model open for operational modifications, allowing the inclusion / exclusion of restrictions, quality criteria, as well as setting the initial and target states using undetermined parameters. The problem of AI planning is proposed to be set and solved in the framework of an object-oriented extension of the programming technology, which places increased demands on the efficiency of processing qualitative constraints. It is proposed to present the qualitative constraints in the form of specialized matrix-like structures, and their processing should be carried out using the author's methods of non-numerical constraint satisfaction. The proposed approach allows structuring semantically closely coupled quantitative and qualitative constraints, simplifying their maintenance, as well as speeding up their automatic generation and processing. As an example of an applied task, the paper considers a simplified version of the task of planning actions for the localization of a territorial spill of petroleum products.

Keywords: AI planning, poorly formalized subject domain, constraint satisfaction problem, constraint programming, object-oriented representation, emergency situation.

DOI 10.14357/20718632190103

References

1. Russel, S., and P. Norvig. 2010. *Artificial Intelligence: A Modern Approach*. 3rd edition. Prentice Hall. 1132 p.
2. Smith, D. E., J. Frank, and Ari K. Jónsson. 2000. Bridging the Gap Between Planning and Scheduling. *Knowledge Engineering Review*. 15(1):47-83.
3. Blum, A. and M. Furst. 1997. Fast planning through planning graph analysis. *Artificial Intelligence* 90:281-300.
4. Rintanen, J. 2009. Planning and SAT. In A. Biere, H. van Maaren, M. Heule and T. Walsh, Eds. *Handbook of Satisfiability*. IOS Press. 483-504.
5. Steger-Jensen, K., H.-H. Hvolby, P. Nielsen, and I. Nielsen, 2011. Advanced planning and scheduling technology. *Production Planning and Control*. Vol. 22. 8:800–808.
6. Bit-Monnot, A. 2018. A Constraint-Based Encoding for Domain-Independent Temporal Planning. *Proceedings of 24th International Conference “Principles and Practice of Constraint Programming”*. Lille. 30-46. doi: 10.1007/978-3-319-98334-9.
7. Margaux, N., C. Artigues, and P. Lopez. 2017. Cumulative scheduling with variable task profiles and concave piecewise linear processing rate functions. *Constraints*. Vol. 22. 4:530-547.
8. Kreter, S., J. Rieck, and J. Zimmermann. 2016. Models and solution procedures for the resource-constrained project scheduling problem with general temporal constraints and calendars. *European Journal of Operational Research*. 251(2):387-403.
9. Letort, A., M. Carlsson, and N. Beldiceanu. 2015. Synchronized sweep algorithms for scalable scheduling constraints. *Constraints*. Vol. 20, 2:183-234. doi: 10.1007/s10601-014-9172-8.
10. Prikaz MChS Rossii No 621. 2004. Ob utverzhdenii Pravil razrabotki i soglasovaniya planov po preduprezhdeniyu i likvidacii razlivov nefi i nefteproduktov na territorii RF. [Order no. 621 of EMERCOM of Russia. About approval of the Rules for the development and coordination of plans for the prevention and elimination of spills of oil and oil products in the territory of the Russian Federation].
11. Yakovlev, S. Yu., A.A. Ry'zhenko, and N.V. Isakevich. 2010. Razrabotka planov po preduprezhdeniyu i likvidacii razlivov nefteproduktov dlya territorial'ny'x ob`ektov [Development of plans for the prevention and elimination of oil spills for territorial objects]. *Modelirovanie i analiz bezopasnosti i riska v slozhny'x sistemax: Trudy` Mezhdunarodnoj nauchnoj shkoly` MABR-2010*[Modeling and analysis of safety and risk in complex systems: Proceedings of the International Scientific School MASR-2010]. Saint-Peterburg. 447-452.
12. Zuenko A. 2019. Matrix-like Structures for Representation and Processing of Constraints Over Finite Domains. *Advances in Intelligent Systems and Computing*. Springer Nature Switzerland AG. Vol. 875:428-438. doi:10.1007/978-3-030-01821-4_45.
13. Zuenko A.A., Lomov P.A., and Oleynik A.G. 2017. Primenenie metodov rasprostraneniya ogranicheniy dlya uskoreniya obrabotki zaprosov k ontologiyam [Applying constraint propagation methods to speed up the processing of ontology requests]. *Trudy SPIIRAN [SPIIRAS Proceedings]*. 1(50):112-136. doi: 10.15622/sp.50.5
14. Zuenko, Alexander, and Y. Oleynik. 2019. Programming of Algorithms of Matrix-Represented Constraints Satisfaction by Means of Choco Library. *Advances in Intelligent Systems and Computing*. Springer Nature Switzerland AG. Vol. 875:1-10. doi: 10.1007/978-3-030-01821-4_46.

Zuenko A. A., PhD (Tech), senior researcher, Institute for Informatics and Mathematical Modeling – Subdivision of the Federal Research Centre “Kola Science Centre of the Russian Academy of Sciences”, 24A Fersman str., 184209, Apatity, Russia; number of publications: 150; research interests: constraint programming poorly formalized subject domains modeling; e-mail: zuenko@iimm.ru (Responsible for the correspondence).

Yakovlev S. Yu. PhD (Tech), associate professor, senior researcher, Institute for Informatics and Mathematical Modeling – Subdivision of the Federal Research Centre “Kola Science Centre of the Russian Academy of Sciences”, 24A Fersman str., 184209, Apatity, Russia; number of publications: 180; research interests: information technology management of industrial and environmental safety; e-mail: Yakovlev@iimm.ru

Shemyakin A. S. Junior researcher, Institute for Informatics and Mathematical Modeling – Subdivision of the Federal Research Centre “Kola Science Centre of the Russian Academy of Sciences”, 24A Fersman str., 184209, Apatity, Russia; number of publications: 40; research interests: information technology management of industrial and environmental safety, e-mail: shemyakin@iimm.ru

Oleynik Yu. A. Junior researcher, Institute for Informatics and Mathematical Modeling – Subdivision of the Federal Research Centre “Kola Science Centre of the Russian Academy of Sciences”, 24A Fersman str., 184209, Apatity, Russia; number of publications: 10; research interests: constraint programming, e-mail: yoleynik@iimm.ru