

ПОДХОД К РЕШЕНИЮ МЕТРИЧЕСКИХ ЗАДАЧ СОГЛАСОВАНИЯ ВРЕМЕННЫХ ОГРАНИЧЕНИЙ В ПОДКЛАССЕ STR¹

Куриленко И.Е.

Московский Энергетический Институт (Технический Университет),
Россия

В практических приложениях часто необходимо не только обеспечивать работу с качественной информацией о времени (когда обрабатываются отношения типа раньше, позже, одновременно), но и метрическую, позволяющую, например, задать длину временных интервалов. При этом для интеллектуальных систем (ИС), ориентированных на применение в открытых и динамических предметных областях, часто требуется обеспечить быстродействующие алгоритмы вывода, способные выдавать решение за минимально возможное время в условиях постоянного изменения исходных данных [1]. В работах [2-5] рассматриваются методы, позволяющие организовать такие алгоритмы для решения задачи качественных временных рассуждений для ИС типа ИС поддержки принятия решений реального времени (ИСППР РВ). Данная работа посвящена адаптации этих методов для обработки метрической информации.

Определим метрическую точечную задачу согласования временных ограничений (МЗСВО) как $Z=(V,D,C_1,C_2)$, где $V = \{V_1, V_2, \dots, V_m\}$ - конечное множество временных переменных, соответствующих моментам времени; D - область значений временных переменных (является либо множеством вещественных чисел, либо множеством целых чисел); C_1 - конечное число

¹ Работа выполнена при финансовой поддержке РФФИ

бинарных временных ограничений вида $C_{ij} = \{[a_1, b_1], \dots, [a_k, b_k]\}$, где интервалы попарно не пересекаются; C_2 - конечное число унарных временных ограничений вида $C_i = \{[a_1, b_1], \dots, [a_l, b_l]\}$, где интервалы попарно не пересекаются. Бинарные ограничения позволяют определить разрешенную дистанцию между моментами времени и интерпретируются как $(a_1 \leq V_j - V_i \leq b_1) \cup \dots \cup (a_k \leq V_j - V_i \leq b_k)$. Унарные – интерпретируются как $(a_1 \leq V_i \leq b_1) \cup \dots \cup (a_l \leq V_i \leq b_l)$ и могут быть представлены с помощью бинарных (через введение нулевого момента времени V_0 и преобразовании унарных ограничений C_i в C_{0i}). Требуется скорректировать все ограничения так, чтобы они не противоречили друг другу. Если подобная корректировка невозможна ЗСМВО считается *несогласованной*.

Рассмотрим основные операции над метрическими точечными ограничениями. Отрицание ограничения $C_{ij} = \{[a_1, b_1], \dots, [a_k, b_k]\}$ определяется как $\sim C_{ij} = \{[-b_1, -a_1], \dots, [-b_k, -a_k]\}$. Пересечение $T \cap S$ ограничений T и S содержит только такие значения, которые одновременно присутствуют в обоих ограничениях. Композиция $T \cdot S$ ограничений T и S содержит только такие значения r , для которых существуют значения $t \in T$ и $s \in S$, такие, что $r = t + s$. Для решения МЗСВО может быть использован алгоритм согласования по путям (алг. 1).

Алгоритм 1. Согласование по путям для МЗСВО.

Входные данные: (V, C) - МЗСВО.

Выходные данные: (V, C) – согласованная по путям МЗСВО.

01: $Q \leftarrow \{(i, k, j) \mid (i < j) \text{ и } k \neq i, j\}$

02: **while** $(Q \neq \emptyset)$ // Выбираем и удаляем путь (i, k, j) из множества Q

03: **if** $(C_{ij} \neq C_{ik} \bullet C_{kj})$ {

04: $C_{ij} \leftarrow C_{ij} \cap (C_{ik} \bullet C_{kj})$

05: **if** $(C_{ij} = \emptyset)$ **return** *inconsistent*

06: $Q \leftarrow Q \setminus \{(i, j, k), (k, i, j) \mid 1 \leq k \leq n, i \neq k \neq j\}$

07: }

08: **return** (V, C)

Ограничение C_{ij} согласовано по путям тогда и только тогда, когда $C_{ij} \subseteq \bigcap_{k \neq i, j} (C_{ik} \bullet C_{kj})$. Множество ограничений согласовано по путям тогда и только

тогда, когда все входящие в него ограничения согласованы по путям. Доказано [6], что алгоритм согласования по путям находит решение целочисленной МЗСВО за время $O(n^3 \delta^3)$, где n – число переменных, а δ – диапазон ограничений (разница между наибольшим и наименьшим значениями, использованными в качестве ограничений). Но при больших значениях δ обеспечение согласованности становится проблематичным. Аналогичная ситуация возникает при попытке адаптации этого алгоритма для МЗСВО, в которой в качестве значений переменных используются вещественные числа. Второй проблемой, возникающей при решении МЗСВО, является фрагментация. Верхняя оценка числа интервалов в композиции временных ограничений $C_{ik} \bullet C_{kj}$ равняется $|C_{ik}| \cdot |C_{kj}|$, где $|C_{ik}|$ и $|C_{kj}|$ – число интервалов в C_{ik} и C_{kj} соответственно. В результате общее число интервалов в согласованной по путям ЗСВО становится настолько велико, что обеспечение согласованности по путям выполняется за неприемлемое время. Однако существует подкласс МЗСВО, известный как подкласс простых задач или STP (Simple Temporal Problems), для которого не возникает этой проблемы. В нем допустимы ограничения, содержащие только один интервал. МЗСВО в подклассе STP определяется как $MZ=(V,C)$, где V – множество переменных, интерпретируемых как моменты времени; C – множество бинарных ограничений. В качестве области значений переменных используется либо множество вещественных, либо множество целых чисел. Ограничения задаются на основе базового временного ограничения $V_j - V_i \in [a, b]$ и представляют собой интервал $C_{ij} \equiv \{[a, b]\}$. Ограничение C_{ij} интерпретируется как $(a \leq V_i - V_j \leq b)$ и задает разрешенную дистанцию между моментами времени.

Расширим подход, предложенный в работе [5] для решения МЗСВО в подклассе STP. Определим граф, взвешенный метрической временной информацией (TL^M-граф), как $G=(W,E,L)$, где W – множество вершин, $W \neq \emptyset$; $E = \{(w_i, [l_k, u_k], w_j) \mid w_i, w_j \in W; l_k, u_k \in \mathbb{R}\}$ – множество связей. Для каждой

временной переменной МЗСВО создается вершина в TL^M -графе, а для каждого ограничения $C_{ik}=[a,b]$ – направленная связь $(W_{i,[a,b]},W_k)$. Пометками ребер в TL^M -графе являются верхняя и нижняя границы разрешенной дистанции между моментами времени, соответствующих вершинам, соединенными этими ребрами. Для решения может быть использован алгоритм Флойда [6]. При пошаговом уточнении решения МЗСВО (когда требуется осуществлять проверку согласованности и обеспечивать ответы на запросы после каждого изменения) с помощью алгоритма Флойда при k изменениях сложность составляет $O(kn^3)$, где n – число переменных, однако можно построить алгоритм меньшей сложности.

В качестве верхней границы веса пути в TL^M -графе примем $\bar{w}(\pi_{xy}) = \sum_{i=0}^{k-1} \bar{w}_{x_i x_{i+1}}$, где $\bar{w}_{x_i x_{i+1}}$ – максимальная разрешенная дистанция между x_i и x_{i+1} (вес b ребра $(x_i, [a,b], x_{i+1})$), k – число связей, задействованных в пути. В качестве нижней границы веса пути в TL^M -графе примем $\underline{w}(\pi_{xy}) = \sum_{i=0}^{k-1} \underline{w}_{x_i x_{i+1}}$, где $\underline{w}_{x_i x_{i+1}}$ – минимальная разрешенная дистанция между x_i и x_{i+1} (вес a ребра $(x_i, [a,b], x_{i+1})$). Будем считать путь π_{xy} существенным, если $\bar{w}(\pi_{xy}) \neq \infty$ и $\underline{w}(\pi_{xy}) \neq -\infty$. Введем множество P всех направленных путей между всеми вершинами в графе G . Введем функцию $\Omega(\pi_{xy}) = (\bar{w}(\pi_{xy}) \neq \infty) \wedge (\underline{w}(\pi_{xy}) \neq -\infty)$. Определим: P' – множество существенных путей в графе G , $P' \subseteq P$; множество входящих в вершину x путей $L_p(x) = \{\pi_{lx} | l \neq x, \pi_{lx} \in P'\}$ и множество выходящих из вершины x путей $R_p(x) = \{\pi_{xm} | m \neq x, \pi_{xm} \in P'\}$. Для вычисления выполнимого ограничения между временными переменными предлагается использовать алг. 2, сложность которого составляет $O(\max(|R_p(x)|, |R_p(y)|, |L_p(x)|, |L_p(y)|))$. Среди существенных путей осуществляется поиск максимальной оценки нижней границы и минимальной оценки верхней границы веса пути. В результате вычисляется выполнимое метрическое ограничение между переменными.

Алгоритмы внесения ограничений (алг. 3) и их удаления (алг. 4) построены таким образом, что они оперируют с множеством существенных путей и что ограничения, которые могут привести к несогласованности, отсекаются на этапе внесения и не участвуют в порождении путей. Сложность алг. 3 оценивается величиной $O(|R_p(y)| \cdot |L_p(x)|)$, требуемый объем памяти – $O(e^2)$. Рассмотрим алгоритм удаления ограничений (алг. 4). Путь π_{xy} вложен в путь π_{ab} ($\pi_{xy} \subseteq \pi_{ab}$), тогда и только тогда, когда π_{ab} представим в виде $\pi_{ax}\pi_{xy}\pi_{yb}$. Вместе с удаляемым ограничением для переменных x и y удаляются все существенные пути, в которые элементарный путь π_{xy} входит в качестве подпути.

Алгоритм 2. Алгоритм вычисления выполнимого ограничения

Входные данные: x, y – переменные; P' – множество всех существенных путей.

Выходные данные: r – выполнимое ограничение для x и y .

01: $ForwardPatches \leftarrow R_p(x) \cap L_p(y)$

02: $RearwardPatches \leftarrow R_p(y) \cap L_p(x)$

03: $u \leftarrow \min_{\pi_{ij} \in ForwardPatches} \bar{w}(\pi_{ij})$

04: $l \leftarrow \max_{\pi_{ij} \in RearwardPatches} \underline{w}(\pi_{ij})$

05: **return** $[l, u]$

Алгоритм 3. Алгоритм создания метрических ограничений

Входные данные: x, y – переменные; $[a, b]$ – метрическое ограничение для создания, $a \leq b$, $a, b \in \mathbb{R}$; P' – множество всех существенных путей в графе G .

01: $[a', b'] \leftarrow$ **Алгоритм 2**(x, y, P')

02: **if** ($[a', b'] \cap [a, b] = \emptyset$) **return inconsistent**

03: **if** ($[a', b']$ сильнее $[a, b]$) **return** // Существующее ограничение сильнее вносимого

04: **foreach** ($\pi_{lx} \in L_p(x)$) {

05: **if** ($\Omega(\pi_{ly}) \cap P = P \cup \pi_{ly}$ // π_{ly} - существенный

06: **foreach** ($\pi_{ym} \in R_p(y)$)

07: **if** ($\Omega(\pi_{lm}) \cap P = P \cup \pi_{lm}$ // π_{lm} существенный

08: }

09: **foreach** ($\pi_{ym} \in R_p(y)$)

10: **if** ($\Omega(\pi_{xm}) \cap P = P \cup \pi_{xm}$ // π_{xm} - существенный

Нетрудно показать, что при применении пошаговых алгоритмов решения МЗСВО в подклассе STP будет получено то же решение, что и при последовательном применении алгоритма Флойда для графа

расстояний.

Алгоритм 4. Алгоритм удаления метрических ограничений

Входные данные: x, y – переменные; P' – множество всех существенных путей в графе G .

Выходные данные: P' – множество всех существенных путей в графе G .

01: **foreach** ($\pi_{ab} : \pi_{xy} \subseteq \pi_{ab}, \pi_{ab} \in P'$)

02: $P' \leftarrow P' \setminus \pi_{ab}$

Рассмотренные алгоритмы реализованы в рамках системы временного вывода Singularity (рис. 1) [7]. Она выполнена с применением языка C# и Microsoft .NET Framework и является кроссплатформенной.

На рис. 1 показаны основные компоненты решателя МЗСВО: таблица путей, словарь временных переменных, блок транзакций и индексы (рис. 1). Словарь временных переменных хранит соответствие строковых имен переменных числовым кодам. Существенные пути хранятся в таблице путей. Каждый путь представляется в виде записи в таблице и представляет собой совокупность следующих характеристик:

- ID – уникальный идентификатор пути (целое число);
- S – вершина, в которой начинается путь (целое число);
- T – вершины, в которую ведет путь (целое число);
- W – вес пути (в виде интервала $[a, b]$);
- L – индекс присоединенного пути слева (-1, если такого пути нет);
- R – индекс присоединенного пути справа (-1, если такого пути нет);
- P – порождающий путь.

Индексы служат для ускорения поиска по таблице путей (в частности, для ускорения выполнения запроса на поиск всех существенных путей, выходящих из вершины v). Для внесения нового ограничения используется алгоритм, порождающий существенные пути, который заполняет таблицу путей в порядке возрастания идентификатора пути.

Стек транзакций служит для поддержки транзакционного режима. По сути, в нем хранится целочисленный идентификатор максимального пути, существующего в таблице путей на момент старта транзакции. Еще одним

преимуществом рассматриваемого подхода является то, что могут быть реализованы алгоритмы возврата и фиксации транзакции обладающие вычислительной сложностью $O(l)$, где l - некоторая константа. Это вызвано тем, что для возврата необходимо просто удалить из таблицы путей все пути с большим уникальным идентификатором, чем сохраненный в стеке.



Рис. 1. Компоненты решателя МЗСВО

На рис. 2 приведены результаты сравнения предлагаемого подхода и решателя МЗСВО, построенного на базе алгоритма Флойда, на множестве тестовых МЗСВО в подклассе STP. Видно, что предлагаемые алгоритмы способны решать МЗСВО в подклассе STP в режиме готовности к ответам на запросы после каждого изменения за существенно меньшее время, чем это можно сделать с помощью алгоритма Флойда.

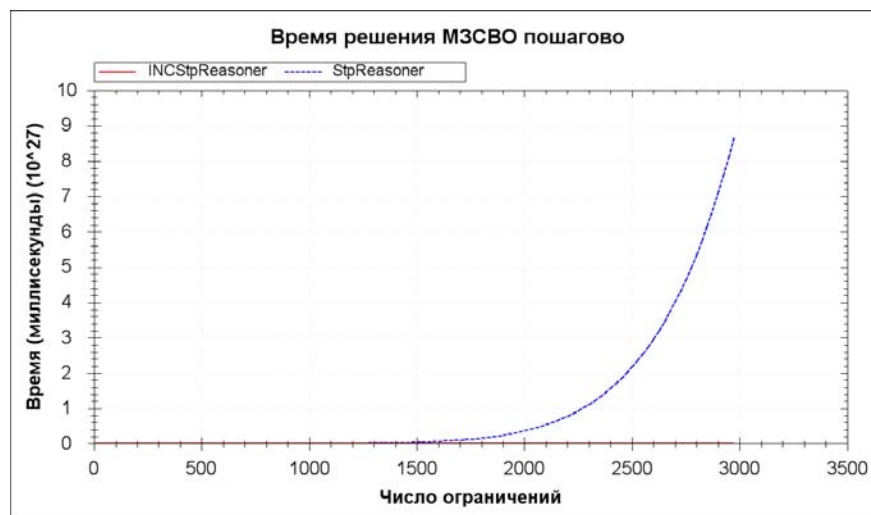


Рис. 2. Сравнение производительности пошаговых решателей МЗСВО

Основным преимуществом рассмотренного подхода является то, что после каждого изменения исходных данных не требуется вызов дополнительных алгоритмов проверки согласованности и поиска решения

МЗСВО, что позволяет экономить время, затрачиваемое на обеспечение временного вывода. За счет этого расширяются возможности по применению механизмов временного вывода в системах типа ИСППР РВ.

ЛИТЕРАТУРА

1. Еремеев А.П., Троицкий В.В. Модели представления временных зависимостей в интеллектуальных системах поддержки принятия решений // Изв. РАН. Теория и системы управления. 2003, № 5, С. 75-88.
2. Еремеев А.П., Куриленко И.Е. Реализация механизма временных рассуждений в современных интеллектуальных системах // Изв. РАН. Теория и системы управления, 2007, № 2, С. 120–136.
3. Еремеев А.П., Куриленко И.Е. «Моделирование временных рассуждений в интеллектуальных системах реального времени» // Вестник МЭИ, 2008. №1. С. 114-123.
4. Куриленко И.Е. О методах улучшения алгоритмов вывода для системы временных рассуждений // Сб. тр. IV–й междунар. научно–практической конф. Интегрированные модели и мягкие вычисления в искусственном интеллекте в 2 т. – Т.1. – М.:ФизМатЛит, 2007 – С.149–156.
5. Куриленко И.Е. «Пошаговые алгоритмы временных рассуждений для точечной модели времени» // Сб. тр. научной сессии МИФИ–2008 в 15 т. – Т.10. – М.:МИФИ, 2008. – С. 132–133.
6. Schwalb E. M. Temporal reasoning with constraints // Ph. D. dissertation, University of California, Irvine, 1998.
7. Еремеев А.П., Куриленко И.Е. Компонента временных рассуждений для интеллектуальных систем поддержки принятия решений реального времени // Искусственный интеллект и принятие решений, 2009, № 1, С. 31–45.