

Галочкин В. И. ————— ■

ПЕРЕЧИСЛЕНИЕ РЕШАЮЩИХ ДЕРЕВЬЕВ ОГРАНИЧЕННОЙ СТОИМОСТИ НА И-ИЛИ ДЕРЕВЕ

Аннотация: Рассматриваются И-ИЛИ деревья с заданной стоимостью дуг либо вершин, широко применяемые в системах искусственного интеллекта. Описывается алгоритм типа ветвей и грани, позволяющий перечислить все решающие деревья, стоимость которых не превышает наперед заданной константы. Трудоемкость получения очередного решающего дерева составляет $O(N)$, где N – количество вершин И-ИЛИ дерева. Указывается способ стековой организации информации, позволяющий свести затраты памяти к величине $O(N)$ без изменения прежней оценки трудоемкости. Выполнена программная реализация описанного алгоритма, подтвердившая при тестировании полученные теоретические оценки трудоемкости и объема необходимой памяти. Эффективность поиска повышена благодаря введенному понятию минимальной оболочки И-ИЛИ дерева по ограничению стоимости, что позволяет гарантировать наличие допустимых решающих деревьев при спуске по дереву решений. Решающие поддеревья перечисляются не по отдельности, а блоками в виде поддеревьев И-ИЛИ дерева, в которых все варианты допустимы.

Ключевые слова: Искусственный интеллект, Алгоритм, Перечисление, И-ИЛИ граф, И-ИЛИ дерево, Решающее дерево, Дерево решения, Вариант И-ИЛИ дерева, Стоимость, Ограничение по стоимости

Введение

И-ИЛИ деревья широко применяются в системах искусственного интеллекта [1-4]. На основе И-ИЛИ деревьев строятся, например, методы сведения задач к подзадам [1, 3] и способы описания множества технических решений определенного класса [2].

И-ИЛИ дерево содержит два типа вершин: И-вершины и ИЛИ-вершины. Решающим деревом (деревом решения) [1] называется такое поддерево, которое строится сверху вниз и включает корень, всех сыновей для каждой своей И-вершины и единственного сына для каждой ИЛИ-вершины. Будем для краткости называть решающее дерево вариантом, как это сделано в [4].

Распространенной задачей для И-ИЛИ деревьев является перечисление вариантов с определенными свойствами [1-4]. В [1] рассматривается задача выбора варианта минимальной стоимости. Стоимость варианта определяется суммарной стоимостью его дуг. В [2] количественные показатели приписываются вершинам И-ИЛИ дерева, а не дугам. Мы будем следовать такому представлению. Фактически, оно получается простым переносом

стоимости дуги на вершину, в которую она направлена. При обратном преобразовании стоимость вершины переносится на стоимость дуги, ведущей в данную вершину.

Рассмотрим следующую задачу. В вершинах И-ИЛИ дерева определены значения стоимости. Стоимостью варианта считается суммарная стоимость его вершин. Требуется перечислить все варианты И-ИЛИ дерева, для которых стоимость не превышает заданной константы C .

1. Минимальная оболочка

Итак, имеется И-ИЛИ дерево D с N вершинами. Обозначим заданное значение стоимости в вершине i через w_i . Функция стоимости W определена рекурсивно на множестве вершин варианта путем обхода дерева варианта от листьев к корню. Действительно, для вершины z с сыновьями z_i при $i = 1, 2, \dots, k$ и стоимостью w_z

$$W(z) = w_z + W(z_1) + W(z_2) + \dots + W(z_k).$$

Стоимостью всего варианта является стоимость его корня.

Назовем поддеревом вариантов множество всех вариантов такого поддерева, которое включает корень, всех сыновей для каждой своей И-вершины и не менее одного сына для каждой ИЛИ-вершины. Пересечение поддеревьев вариантов представляет собой множество вариантов подобного И-ИЛИ поддерева, поэтому также является поддеревом вариантов.

Будем считать допустимыми варианты И-ИЛИ дерева, удовлетворяющие ограничению стоимости $W \leq C$. Назовем минимальной оболочкой И-ИЛИ дерева по ограничению $W \leq C$ такое поддерево вариантов, которое

- включает все допустимые варианты исходного И-ИЛИ-дерева;
- не может быть усечено без потери допустимых вариантов.

Из этого определения следует, что каждая вершина минимальной оболочки входит в какой-либо допустимый вариант.

Рассмотрим способ построения минимальной оболочки. Для любой вершины z найдем минимальный отрезок $[a_z, b_z]$, в котором лежат все возможные значения стоимости. Будем называть $[a_z, b_z]$ отрезком возможных значений стоимости в вершине z .

Выполним обход дерева в направлении от листьев к корню r .

Пусть z_i ($i = 1, 2, \dots, k$) – сыновья вершины z , w_z – заданная стоимость вершины z , а $[a_i, b_i]$ – рассчитанные отрезки возможных значений стоимости в вершинах z_i (для листьев $a_i = b_i = z_i$).

Если z – И-вершина, то $a_z = w_z + a_1 + a_2 + \dots + a_k$; $b_z = w_z + b_1 + b_2 + \dots + b_k$.

Если z – ИЛИ-вершина, то $a_z = w_z + \min(a_i)$; $b_z = w_z + \max(b_i)$ для $i = 1, 2, \dots, k$.

Таким образом находятся отрезки возможных значений для всех вершин И-ИЛИ дерева. Если окажется, что в корне $a_r > C$, то минимальная оболочка пуста. Это означает, что допустимые варианты отсутствуют. Если же $b_r \leq C$, то любой вариант И-ИЛИ дерева

является допустимым, а минимальная оболочка совпадает со всем деревом. Будем далее считать, что $a_r \leq C \leq b_r$, что гарантирует наличие допустимых вариантов.

Следующий обход выполняется наоборот от корня дерева к листьям. При тех же обозначениях для любой вершины z_i найдем максимальное значение стоимости h_i , при котором может существовать допустимый вариант, включающий вершину z_i .

Пусть для вершины z со стоимостью w_z и сыновьями z_i ($i = 1, 2, \dots, k$) указанное значение равно h_z (для корня $h_r = C$).

Если z – И-вершина, рассмотрим для каждого i уравнение

$$w_z + a_1 + a_2 + \dots + a_{i-1} + h_i + a_{i+1} + \dots + a_k = h_z,$$

откуда

$$h_i = h_z - (w_z + a_1 + a_2 + \dots + a_{i-1} + a_{i+1} + \dots + a_k).$$

Если z – ИЛИ-вершина, то отсечем те вершины z_i , для которых $w_z + a_i > h_z$. Для остальных вершин $h_i = h_z - w_z$.

Легко видеть, что в результате описанной процедуры получается поддерево решений T , являющееся минимальной оболочкой дерева D по ограничению $W \leq C$. Трудоемкость построения минимальной оболочки оценивается величиной $O(N)$.

2. Допустимые поддеревья вариантов

Будем называть допустимым такое поддерево вариантов, в котором каждый вариант является допустимым. Допустимые поддеревья вариантов можно выделять, распространяя последовательно от корня к листьям ограничения на стоимость.

Пусть для вершины z со стоимостью w_z и сыновьями z_i ($i = 1, 2, \dots, k$) получено ограничение $W(z) \leq d_z$ (для корня $d_r = C$). Наша задача получить ограничения вида $W(z_i) \leq d_i$.

Если z – И-вершина, подберем произвольные значения $a_i \leq d_i \leq b_i$ такие, что

$$w_z + d_1 + d_2 + \dots + d_k = d_z \tag{1}$$

и будем считать, что в i -й вершине получено ограничение $W(z_i) \leq d_i$.

Можно, например, определять величины d_i по формулам $d_i = d_i(t) = a_i + t(b_i - a_i)$, решая уравнение (1) относительно t .

Если z – ИЛИ-вершина, отсечем вершины z_i , для которых $w_z + a_i > d_z$, а для остальных вершин положим $d_i = d_z - w_z$.

В итоге выделяется допустимое поддерево вариантов P . Будем говорить, что P получено в результате спуска ограничения $W \leq C$.

В общем случае, минимальная оболочка не является допустимым поддеревом вариантов, но гарантированно содержит вариант минимальной стоимости. Трудоемкость получения допустимого поддерева вариантов оценивается величиной $O(N)$.

3. Разложение разности минимальной оболочки и допустимого поддерева

Теорема. Пусть T – минимальная оболочка И-ИЛИ дерева, а P – допустимое поддерево вариантов, полученное в результате спуска ограничения $W \leq C$, причем $P \neq T$. Тогда существуют такие поддерева вариантов S_i ($i = 1, 2, \dots, m$; $m \leq N$), что

$$1) T \setminus P = \bigcup_{i=1}^m S_i;$$

$$2) S_i \cap S_j = \emptyset \text{ при } i \neq j;$$

3) каждое поддерево вариантов S_i содержит допустимые варианты.

Доказательство. Пусть X – множество вершин из T , y – произвольная ИЛИ-вершина P . Введем следующие обозначения:

- $Z(X, T)$ – множество вариантов из T , не содержащих вершин множества X ;
- $G(X, T)$ – множество вариантов из T , содержащих хотя бы одну вершину из X ;
- $R(y, P)$ – множество сыновей ИЛИ-вершины $y \in P$, не лежащих в P ;
- $H(P, T)$ – множество ИЛИ-вершин $y_i \in P$, для которых $R(y_i, P) \neq \emptyset$.

Заметим, что если вершина z_i является сыном И-вершины z и $z_i \in X$, то варианты множества $Z(X, T)$ не могут содержать вершину z . Очевидно, что $Z(X, T)$ является поддеревом вариантов и для любых множеств вершин A и B

$$Z(A \cup B, T) = Z(A, T) \cap Z(B, T).$$

Множество $G(R(y, P), T)$ также является поддеревом вариантов. Легко проверяется равенство

$$T \setminus P = \bigcup_{y_i \in H(P, T)} G(R(y_i, P), T). \quad (2)$$

Положим

$$S_i = G(R(y_i, P), T) \cap Z\left(\bigcup_{j < i} G(y_j, T)\right), \quad i = 1, 2, \dots, m.$$

Из равенства (2) и определения S_i следует справедливость первых двух утверждений теоремы. Докажем, что выполняется и третье утверждение.

Зафиксируем номер $i \leq m$. По определению минимальной оболочки, существует допустимый вариант $u \in G(R(y_i, P), T)$. Выберем все вершины y_{i_k} ($k = 1, 2, \dots, q$) такие, что $i_k < i$ и $u \in G(R(y_{i_k}, P), T)$.

Обозначим через u_{i_k} поддерево с корнем y_{i_k} , являющееся частью варианта u , а через t_{i_k} поддерево с корнем y_{i_k} , являющееся частью варианта минимальной стоимости.

По построению P , поддерево t_{i_k} целиком лежит в P . Сформируем новый вариант t , заменив в варианте u все поддерева u_{i_k} на поддерева t_{i_k} ($k = 1, 2, \dots, q$). Значения

стоимости W на варианте t не больше, чем на варианте u , поэтому t также является допустимым вариантом.

Поскольку поддеревья t_{i_k} лежат в P , то $t \in Z(\bigcup_{j < i} G(y_j, T))$. Следовательно, $t \in S_i$ и теорема доказана.

4. Алгоритм перечисления допустимых вариантов

На основании доказанной теоремы построим алгоритм типа ветвей и границ [5] перечисления допустимых поддеревьев вариантов. Сохраняя обозначения теоремы, выделим из исходного И-ИЛИ дерева D минимальную оболочку T , а из T путем спуска ограничения $W \leq C$ найдем допустимое поддерево вариантов P .

Если $P \neq T$, определим множество $H(P, T)$, вершину $y_1 \in H(P, T)$ и поддерево вариантов $S_1 = G(R(y_1, P), T)$. По доказанному, в S_1 имеются допустимые варианты. Выделим из S_1 минимальную оболочку T_1 и допустимое поддерево вариантов P_1 .

Как и ранее, если $P_1 \neq T_1$, найдем множество $H(P_1, T_1)$, вершину $y_{11} \in H(P_1, T_1)$ и поддерево вариантов S_{11} , из которого будет выделено очередное допустимое поддерево вариантов P_{11} .

Продолжим подобный поиск в глубину, пока очередное допустимое поддерево вариантов не совпадет с минимальной оболочкой. Тогда вернемся к предыдущей минимальной оболочке и выделим на ней следующее допустимое поддерево вариантов. Ввиду того, что минимальная оболочка всегда содержит допустимые варианты, очередное допустимое поддерево вариантов находится с трудоемкостью $O(N)$.

Обозначим через $|A|$ число вершин в поддереве вариантов A . Поскольку при любой последовательности номеров i_1, i_2, \dots, i_k выполняется

$$T \supset T_{i_1} \supset T_{i_1 i_2} \supset \dots \supset T_{i_1 i_2 \dots i_k}, \quad (3)$$

то

$$N \geq |T_{i_1}| > |T_{i_1 i_2}| > \dots > |T_{i_1 i_2 \dots i_k}| \quad (4)$$

и цепочка (3) содержит не более N элементов.

Обозначим через E_{i_1} множество вершин из $T \setminus T_{i_1}$, а через $E_{i_1 i_2 \dots i_k}$ при $k > 1$ – множество вершин из $T_{i_1 i_2 \dots i_{k-1}} \setminus T_{i_1 i_2 \dots i_k}$. При реализации алгоритма можно хранить в памяти минимальную оболочку T и текущую цепочку множеств $E_{i_1}, E_{i_1 i_2}, \dots, E_{i_1 i_2 \dots i_k}$. Аналогичным образом сохраняется информация о допустимых поддеревьях вариантов $P, P_{i_1}, P_{i_1 i_2}, \dots, P_{i_1 i_2 \dots i_k}$.

После исчерпания допустимых вариантов из $T_{i_1 i_2 \dots i_k}$ легко восстановить минимальную оболочку $T_{i_1 i_2 \dots i_{k-1}}$, допустимое поддерево вариантов $P_{i_1 i_2 \dots i_{k-1}}$ и перейти к выделению следующего допустимого поддерева вариантов, обновив необходимую информацию в памяти. Как видно из (4), для этого требуется объем памяти порядка N .

Описанный способ организации информации фактически предполагает использование стека. Объем вычислений при этом по-прежнему имеет порядок N .

Отметим в заключение, что алгоритм позволяет выбирать допустимые варианты не

по отдельности, а блоками в виде поддеревьев И-ИЛИ дерева, в которых все варианты допустимы, то есть удовлетворяют ограничению по стоимости $W \leq C$.

Библиография :

1. Нильсон, Н. Искусственный интеллект. Методы поиска решений / Н. Нильсон. – М.: Мир, 1973. – 270 с.
2. Автоматизация поискового конструирования / А. И. Половинкин, Н. К. Бобков, Г. Я. Буш и др. Под редакцией А. И. Половинкина. – М.: Радио и связь, 1981. – 344 с.
3. Братко, И. Программирование на языке Пролог для искусственного интеллекта / И. Братко – М.: Мир, 1990. – 560 с.
4. Кручинин, В.В. Методы построения алгоритмов генерации и нумерации комбинаторных объектов на основе деревьев И/ИЛИ / В.В. Кручинин. – Томск: В-Спектр, 2007. – 200 с.
5. Рейнгольд, Э. Н. Комбинаторные алгоритмы. Теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део. – М.: Мир, 1980. – 476 с.

References:

1. Nil'son, N. Iskustvennyi intellekt. Metody poiska reshenii / N. Nil'son. – M.: Mir, 1973. – 270 s.
2. Avtomatizatsiya poiskovogo konstruirovaniya / A. I. Polovinkin, N. K. Bobkov, G. Ya. Bush i dr. Pod redaktsiei A. I. Polovinkina. – M.: Radio i svyaz', 1981. – 344 s.
3. Bratko, I. Programmirovaniye na yazyke Prolog dlya iskustvennogo intellekta / I. Bratko – M.: Mir, 1990. – 560 s.
4. Kruchinin, V.V. Metody postroeniya algoritmov generatsii i numeratsii kombinatornykh ob'ektov na osnove derev'ev I/ILI / V.V. Kruchinin. – Tomsk: V-Spekt, 2007. – 200 s.
5. Reingol'd, E. N. Kombinatornye algoritmy. Teoriya i praktika / E. Reingol'd, Yu. Nivergel't, N. Deo. – M.: Mir, 1980. – 476 s.