Тема 3. ИСЧЕРПЫВАЮЩИЙ ПОИСК

3.1. Поиск с возвратом

3.1.4. Оценка сложности выполнения поиска с возвратом

Обычно поиск с возвратом приводит к алгоритмам, экспоненциальным по своим параметрам. Это является следствием того, что если все решения имеют длину не более n, то исследованию подлежат приблизительно $\prod_{i=1}^{n} |A_i|$ вершин дерева поиска, где $|A_i|$ – мощность множества A_i. Даже широкое использование ограничений и склеиваний позволяет в большинстве случаев добиться только того, что $|A_i|$ становится константой; при этом получаются деревья примерно с C^n вершинами для некоторой константы $C > 1$. Поскольку размеры дерева растут так быстро, можно попытаться определить возможность практического осуществления поиска (т. е. за приемлемое время) путем оценки числа вершин в дереве.

Аналитическое выражение для оценки удается получить редко, так как трудно предсказать, как взаимодействуют различные ограничения по мере появления их при продвижении в глубь дерева поиска. В подобных случаях, когда построение аналитической модели является трудной или вовсе неосуществимой задачей, можно применить *метод Монте-Карло* (метод статистических испытаний). Смысл этого метода в том, что исследуемый процесс моделируется путем многократного повторения его случайных реализаций. Каждая случайная реализация называется *статистическим испытанием*.

Рассмотрим применение метода Монте-Карло для экспериментальной оценки размеров дерева поиска. Идея метода состоит в проведении нескольких испытаний, при этом каждое испытание представляет собой поиск с возвратом со случайно выбранными значениями a_i. Предположим, что имеется частичное решение $(a_1, a_2, \ldots, a_{k-1})$ и что число выборов для a_k, основанное на том, вводятся ли ограничения или осуществляется склеивание, равно $x_k = |S_k|$. Если $x_k \neq 0$, то a_k выбирается случайно из S_k и для каждого элемента вероятность быть выбранным равна $1/x_k$. Если $x_k = 0$, то испытание заканчивается. Таким образом, если $x_1 = |S_1|$, то $a_1 \in S_1$ выбирается случайно с вероятностью $1/x_1$; если $x_2 = |S_2|$, то при условии, что a_1 было выбрано из S_1, $a_2 \in S_2$ выбирается случайно с вероятностью $1/x_2$ и т. д. Математическое ожидание $x_1 + x_1x_2 + x_1x_2x_3 + x_1x_2x_3x_4 + \ldots$ равно числу вершин в дереве поиска, отличных от корня, т. е. оно равно числу случаев, которые будут исследованы алгоритмом поиска с возвратом. Существует доказательство этого утверждения [21].

Общий алгоритм поиска с возвратом легко преобразуется для реализации таких испытаний; для этого при $S_k = \varnothing$ вместо возвращения просто заканчивается испытание. Алгоритм 3.4 оценки размера дерева поиска осуществляет N испытаний для подсчета числа вершин в дереве. Операция $a_k \leftarrow rand(S_k)$ реализует случайный выбор элемента a_k из множества S_k.

Таким образом, каждое испытание представляет собой продвижение по дереву поиска от корня к листьям по случайно выбираемому на каждом уровне направлению. Поскольку в методе Монте-Карло отсутствует возврат, оценка размеров дерева выполняется за полиномиальное время.

$$
\begin{array}{l}
\text{определить } S_1 \subseteq A_1 \\
count \leftarrow 0 \\
k \leftarrow 1 \\
\textbf{while } k > 0 \textbf{ do} \left\{ \begin{array}{l}
\textbf{while } S_k \neq \varnothing \textbf{ do} \left\{ \begin{array}{l}
\text{// продвижение} \\
a_k \leftarrow \text{элемент из } S_k \\
S_k \leftarrow S_k - \{a_k\} \\
count \leftarrow count + 1 \\
\textbf{if } (a_1, a_2, \ldots, a_k) - \text{решение} \\
\quad \textbf{then} \text{ записать его} \\
k \leftarrow k + 1 \\
\text{определить } S_k \subseteq A_k
\end{array} \right. \\
k \leftarrow k - 1 \quad \text{// возвращение}
\end{array} \right. \\
\text{// все решения найдены}
\end{array}
$$

$$
\begin{array}{l}
count \leftarrow 0 \quad \text{// суммарное число вершин в дереве} \\
\textbf{for } i \leftarrow 1 \textbf{ to } N \textbf{ do} \\
\quad \left\{ \begin{array}{l}
sum \leftarrow 0 \quad \text{// число вершин при одном испытании} \\
product \leftarrow 1 \quad \text{// накапливаются произведения} \\
\text{определить } S_1 \subseteq A_1 \\
k \leftarrow 1 \\
\textbf{while } S_k \neq \varnothing \textbf{ do} \left\{ \begin{array}{l}
product \leftarrow product * |S_k| \\
sum \leftarrow sum + product \\
a_k \leftarrow rand\,(S_k) \\
k \leftarrow k + 1 \\
\text{определить } S_k \subseteq A_k
\end{array} \right. \\
count \leftarrow count + sum
\end{array} \right. \\
average \leftarrow count / N \quad \text{// среднее число вершин в дереве}
\end{array}
$$

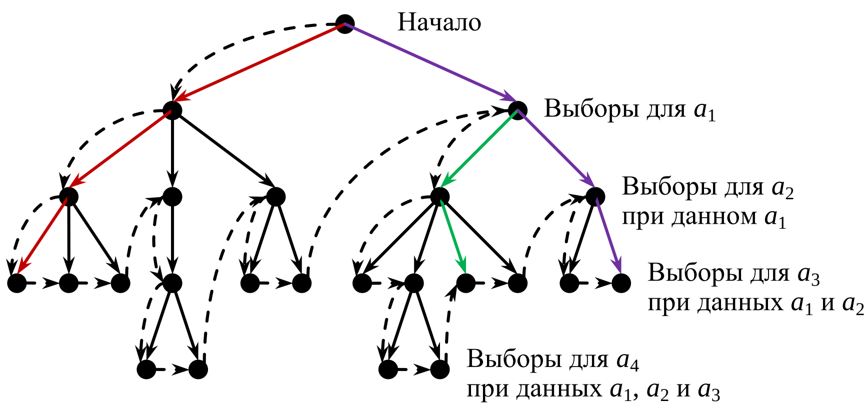

Рис. 3.1. Дерево поиска (24 вершины)

Формула: $x_1 + x_1x_2 + x_1x_2x_3 + x_1x_2x_3x_4 + \ldots$

Проведем 3 эксперимента: самый левый путь (красный), самый правый путь (лиловый) и 4-й справа путь (зеленый)

1) 2 + 2*3 + 2*3*3 = 2+6+18 = 26

2) 2 + 2*2 + 2*2*2 = 2+4+8 = 14

3) 2 + 2*2 + 2*2*4 = 2+4+16 = 22

Среднее $\lceil(26+14+22)/3\rceil = \lceil 20{,}67 \rceil = 21$