Д. А. ЕЛИУСИЗОВ

О СВОДИМОСТИ ОДНОГО КЛАССА ЭКСТРЕМАЛЬНЫХ КОМБИНАТОРНЫХ ЗАДАЧ НА МНОЖЕСТВЕ ПЕРЕСТАНОВОК К ПЕРЕБОРУ ПОДМНОЖЕСТВ

(Представлена академиком НАН РК К. К. Блиевым)

При помощи метода динамического программирования [1] осуществляется сводимость класса задач переборного типа на перестановках к перебору всевозможных подмножеств. Дается оценка сложности и эффективности метода.

Введение. Многие задачи, при решении которых используется переборный алгоритм на перестановках, показывают удовлетворительный результат при очень малых значениях входных данных. Описанный подход позволяет добиться улучшения времени работы алгоритма и размера входных данных почти в два раза.

Определение. Рассмотрим класс задач, функционал которых определен на некотором множестве $U = \{1,2,...,n\}$, а решением задачи является некоторая перестановка этого множества.

Определим исходную задачу как задачу минимизации функции:

$$F : \sigma_n \to A, \tag{1}$$

где σ_n – множество всех перестановок множества U; A – множество возможных значений искомого функционала.

Общая схема решения. Для решения задач типа (1), определенных на множестве перестановок, рассмотрим следующую схему:

Перестановка →
→ множество + фиксированные элементы

Здесь знак + означает, что мы добавляем информацию о фиксированных элементах. Таким образом, вместо хранения всей перестановки на каждом шаге решения будем лишь оставлять для рассмотрения подмножество (где не важен порядок) и некоторые фиксированные элементы.

Пример. $(1,3,4,2) \to \{1,2,3,4\} + (\{2\}$ – последний элемент). Таким образом, перестановку $(1,3,4,2)$ мы рассматриваем как множество $\{1,2,3,4\}$, а также добавляем дополнительную информацию о том, что элемент 2 является последним в исходной перестановке.

Представление множеств. Для реализации наиболее удобно хранить множество в виде двоичных чисел, если соответствующий номер определенного бита равен 1, то элемент с таким номером (такое число) присутствует в подмножестве и нет, если бит равен 0.

Пример. Множество $\{1,6,12,3\}$ можно кодировать в виде следующего бинарного числа: $2^{12} + 2^6 + 2^3 + 2^1 = (1000001001010)_2$.

1. Построение функции для алгоритма решения задач типа (1). Введем новую функцию

$$D : (S \subseteq U) \times U \to A,$$

которая имеет тот же смысл, что и функция F, но определена на декартовом произведении двух множеств. Такое определение функции D на множестве подмножеств помогает заменить определение функции F на множестве перестановок.

Рекуррентное соотношение для нахождения значений функции D имеет следующий вид:

$$D(S \cup x, x)_{x \in U \setminus S} = opt_{j \in S}\{D(S, j) \oplus \alpha(j, x)\}, \tag{2}$$

где opt означает задачу нахождения экстремума функции (например, минимума или максимума), $\oplus$ – операцию изменения нового значения функционала после “фактора влияния” отношения $\alpha(j,x)$, когда новый элемент x становится фиксированным. Отношение $\alpha(j,x)$ может быть рассмотрено значением перехода от элемента j к элементу x. К примеру, расстояние от вершины j до вершины x.

Заметим, что в качестве обобщения (2), которое в нашем случае применимо при решении задачи о назначениях, можно рассматривать и функцию

$$D': S_1 \times S_2 \times ... S_k \times U^r \to A,$$

где $S_i \subseteq U$ для $i = 1,...,k$, а r – количество фиксированных элементов, U^r – степень множества U.

2. Примеры решения комбинаторных оптимизационных задач. Два из описанных ниже примеров относятся к классу NP-полных задач, а задача о назначениях показана лишь для лучшего описания алгоритма (как известно, венгерский алгоритм [2] имеет сложность $O(n^3)$).

Задача коммивояжера. В качестве функции D для задачи коммивояжера мы рассмотрим $D(S, x)$ – наименьшее возможное значение затрат, достигаемое при обходе всех городов подмножества S, если при этом маршрут был окончен в городе с номером x.

Тогда рекуррентное соотношение запишется следующим образом:

$$D(S \cup x, x)_{x \in U \setminus S} = \min_{j \in S}\{D(S, j) + c(j, x)\},$$

$c(j, x)$ – расстояние от города j до города x.

Ответом же к задаче является $\min_i\{D\{1,2,...,n\}, i\}$.

Псевдокод алгоритма:

$D \to \infty$; D[1][0]:=0; \\set initial values

for all subsets $V \subseteq [n]$ **do**

for all vertices $x \in [n]$ **do**

if $D[V][x] < \infty$ **then**

for all vertices $i \notin V$ **do**

if $D[V \cup i][i] > D[V][x] + c[x][i]$ **then**

$D[V \cup i][i] > D[V][x] + c[x][i]$

Сложность алгоритма $O(2^n n^2)$.

Задача о поиске гамильтонова цикла. В данном примере мы определим аналогичную функцию, как и в задаче коммивояжера, но принимающую только значения true или false.

Пусть $D(S, x)$ – означает: можно ли обойти все города подмножества S и при этом маршрут был окончен в городе с номером x.

Рекуррентное соотношение запишется следующим образом:

$$D(S \cup x, x)_{x \in U \setminus S} = \max_{j \in S}\{D(S, j) \,\&\, c(j, x)\},$$

где $c(j, x) = 1$, если существует ребро от j до x и $c(j, x) = 0$ в противном случае.

Сложность алгоритма $O(2^n n^2)$.

Задача о назначениях. Заданы два конечных множества A, B и матрица $a[A][B]$ (можно считать, что $|A| = |B|$). Целью является найти взаимно однозначное соответствие $f : A \to B$, которое минимизирует сумму

$$G(a[A][B]) = \sum_k a[k][f(k)].$$

Для решения этой задачи мы введем обобщенный вариант функции D.

Пусть $D(S_1, S_2, i, j)$ – минимальное значение $G[a[S_1][S_2]]$, если при этом $f(i) = j$.

Сложность алгоритма составит $O(4^n n^4)$.

Описанный метод можно также применить при решении задачи сегментации программ [3, 4].

ЛИТЕРАТУРА

1. *Thomas H. Cormen, Charles E.* Leiserson. Introduction to algorithms. 2-nd edition. USA: MIT Press, 2001. 1180 p.

2. *Harold W. Kuhn.* // The Hungarian Method for the assignment problem. Naval Research Logistics Quarterly. 2: 83-97. 1955.

3. *Дюйсембаев А.Е.* Математические модели сегментации программ. М.: Физматлит, 2001. 207 с.

4. *Елиусизов Д.А.* О комбинаторных особенностях задачи сегментации программ с WS стратегией обмена // Сб. тез. 13-ой междун. науч. конф. молодых ученых «Ломоносов 2006», Апрель 2006. МоиН РФ. МГУ им. Ломоносова, изд. Факультета ВмиК. 20-21-с.

Резюме

Динамикалық бағдарламалау әдісі арқылы терімділік типтегі ауыстырылымдар барлық мүмкін болатын жиыншалардың ауыстырылымына жинақталатыны көрсетіледі. Әдістің қиындылығы мен тиімділігіне баға берілді.

Summary

Using the Dynamic Programming method the class of exhaustive problems on all permutations is reduced to enumeration of all subsets. The estimate of complexity and effectiveness is given.

УДК 681.3

Казахстанско-Британский технический университет, г. Алматы

Поступила 25.10.10г.