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:

$$F : \sigma_n \to A, \qquad (1)$$

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:

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

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