$\overline{\gamma^-(x)}$ компактно, а значит, множество $L^-(x)$ непусто компактно инвариантно и принадлежит Y в силу того, что Y замкнуто. Далее, по построению $L^-(x)\setminus M \subset A_Y(M)$ и поэтому $\forall y \in L^-(x)\setminus M$ имеем: $d(yt, M) \to 0$ при $t \to +\infty$, а поскольку $L^+(y)$ замкнуто, $L^+(y) \cap M \neq \varnothing$. Однако $L^+(y) \subset L^-(x)$, что соответствует условию $L^-(x) \cap M \neq \varnothing$. Последнее, согласно лемме 1, противоречит Y-устойчивости M, что и завершает доказательство теоремы 1.

Теорема 2. Окрестность H множества M в Y является областью притяжения асимптотически устойчивого относительно Y множества M в том и только в том случае, когда выполняются два условия:

1) $\forall x \in H$ $L^+(x)$ — компактное подмножество H;

2) $\forall x \in H\setminus M$ множество $L^-(x) \cap H$ не содержит компактных инвариантных подмножеств.

Доказательство. Пусть выполнены требования 1), 2). Тогда, по теореме 1, множество M является Y-асимптотически устойчивым. Пусть $x \in H$. Тогда из 1) следует, что $L^+(x)$ компактно и содержится в H, так как Y замкнуто. Легко видеть, что в силу 2) и инвариантности множества предельных точек $L^+(x) \subset M$ (см., например, доказательство теоремы 1). Таким образом, $H = A_Y(M)$.

Обратно, пусть M является Y-асимптотически устойчивым и H — его область притяжения относительно Y. В этом случае ясно, что 1) выполняется, поскольку M компактно. Предположим, что 2) не выполняется. Тогда для некоторого элемента $x \in X\setminus M$ существует компактное инвариантное подмножество N, принадлежащее $L^-(x) \cap H$. Следовательно, $\forall y \in N$ имеем: $y \in H$, а значит, $L^+(y) \subset M$. Отсюда в силу замкнутости множеств предельных точек $L^+(y) \subset L^-(x)$ и, в частности, $L^-(x) \cap M \neq \varnothing$, что по лемме 1 противоречит Y-устойчивости M.

Таким образом, условие 2) также выполняется. Из доказанного утверждения следует

Теорема 3. Множество M является глобально асимптотически устойчивым относительно Y тогда и только тогда, когда выполняются условия:

1) $\forall x \in Y\setminus M$ — множество $L^+(x)$ компактно;

2) $\forall x \in Y\setminus M$ — множество $L^-(x) \cap Y$ не содержит компактных инвариантных подмножеств.

Замечание. Приведенные исследования можно распространить путем соответствующих модификаций на дискретные динамические системы [1], а также на динамические системы с многомерным временем [5].

Список литературы

1. Сибирский К. С. Введение в топологическую динамику. Кишинев, 1970.
2. Bhathia N. P., Sregö G. P. Stability Theory of Dynamical Systems. Berlin, 1970.
3. Kalitine B.// R.A.I.R.O. Automatique / Systems Analysis and Control. 1982. V. 16, № 3. P. 275—286.
4. Калитин Б. С.// Дифференц. уравнения. 1986. Т. 22. № 2. С. 187.
5. Гайшун И. В. Вполне разрешимые многомерные дифференциальные уравнения. Минск, 1983.

Поступила в редакцию 31.01.85.

УДК 519.1

П. В. ГЛЯКОВ, М. М. КОВАЛЕВ, В. М. КОТОВ

СЕРИЯ ГРАДИЕНТНЫХ АЛГОРИТМОВ ДЛЯ ЗАДАЧИ О РАЗБИЕНИИ ДЕРЕВА

Рассмотрим следующую задачу. Пусть $T = (V, E)$ — дерево с множеством вершин V и множеством ребер E. На множестве вершин и ребер заданы функции $c: V \to R_+$, $w: E \to R_+$, которые определяют вес вершины и стоимость ребра соответственно. Разбиением P дерева T назо-

вем набор $(P_1, P_2, \ldots, P_k)$ непустых, непересекающихся поддеревьев P_i таких, что каждая вершина дерева T содержится точно в одном поддереве P_i. Поддеревья будем называть кластерами, если они удовлетворяют условию связности, т. е. если вершины j, k принадлежат некоторому поддереву P_i, то в нем содержится путь, соединяющий j и k.

Определим стоимость и вес кластера $P_i = (V_i, E_i)$ следующим образом: $w(P_i) = \sum_{e_j \in E_i} w(e_j)$, $c(P_i) = \sum_{v_j \in V_i} c(v_j)$, а стоимость разбиения как сумму стоимостей кластеров $w(P) = \sum_{i=1}^{k} w(P_i)$. Тогда задача о разбиении дерева T может быть сформулирована следующим образом:

$$\max\{w(P) \mid P = (P_1, \ldots, P_k),\ c(P_i) \leqslant B,\ 1 \leqslant i \leqslant k\}, \qquad (1)$$

где $B \in R_+$ — некоторая заданная величина. Разбиение P, удовлетворяющее условию $c(P_i) \leqslant B$, $1 \leqslant i \leqslant k$, будем называть допустимым.

Задача о разбиении дерева возникает при решении задач классификации, а также при разбиении любой иерархической структуры на минимальное число сегментов, когда существует ограничение на размер сегмента. Например, часто требуется программы и большие системы баз данных разбить на страницы фиксированного размера или разместить на нескольких внешних устройствах. В этом случае ожидаемое время обработки или число устройств должно быть минимизировано. Наконец, пусть дана сеть распределенных пунктов обслуживания. Возникает вопрос, как разместить некоторые из этих пунктов на одном центральном пункте так, чтобы доход от этого размещения был максимальным.

Известно, что задача о разбиении дерева в рассмотренной постановке является NP-трудной. Для ее решения Лакс [1] предложил алгоритм динамического программирования с оценкой сложности $O(nB^2)$, а Шрадер [2] построил быстрый ε-приближенный алгоритм. В данной работе приводится субоптимальный алгоритм для решения задачи (1), основанный на схеме релаксаций и разбиений, идея которой состоит в следующем [3]. На первом этапе условия исходной задачи ослабляются и находится решение полученной задачи. На втором этапе это решение разбивается на ряд решений, допустимых для исходной задачи, и лучшее из них принимается за приближенное решение исходной задачи.

В дальнейшем будем предполагать, что $c(v) \leqslant B$ для всех $v \in V$, так как в противном случае задача (1) не имеет решения. Будем также считать, что в T нет ни одной пары смежных вершин i, j, для которых выполняется условие $c(i) + c(j) > B$. Если бы такие пары вершин существовали, то исходное дерево необходимо было бы разбить на поддеревья удалением из него ребер (i, j) и решать исходную задачу для каждого поддерева в отдельности.

Алгоритм

1. Выбрать в дереве T в качестве корня произвольную вершину и ввести ориентацию ребер от корня к листьям. Перенумеровать вершины дерева по уровням. Положить $i = 1$, $L = \varnothing$, $Z = \varnothing$.

2. Если вершина i — лист, перейти к п. 4. Пусть $v_1, \ldots, v_{h_i}$ — сыновья вершины i. Если выполняется условие

$$c(i) + \sum_{j=1}^{h_i} c(v_j) > B, \qquad (2)$$

перейти к п. 3, иначе положить $Z_i = \{(i, v_1), \ldots, (i, v_{h_i})\}$, $Z = Z \cup \{Z_i\}$ и перейти к п. 4.

3. Переупорядочить вершины $v_1, \ldots, v_{h_i}$ следующим образом:

$$\frac{w(i, v_1)}{c(v_1)} \geqslant \frac{w(i, v_2)}{c(v_2)} \geqslant \ldots \geqslant \frac{w(i, v_{h_i})}{c(v_{h_i})}.$$

Найти k_i, для которого выполняются условия

$$c(i) + \sum_{j=1}^{k_i} c(v_j) \leqslant B, \quad c(i) + \sum_{j=1}^{k_i+1} c(v_j) > B. \tag{3}$$

Положить $Z_i = \{(i, v_1), \ldots, (i, v_{k_i})\}$, $Z = Z \cup \{Z_i\}$. Если выполняется условие

$$c(i) + \sum_{j=1}^{k_i} c(v_j) < B, \tag{4}$$

положить $L = L \cup \{(i, v_{k_i+1})\}$.

4. Положить $i = i + 1$. Если $i \leqslant n$, перейти к п. 2. Разбить множество звезд Z на два подмножества R_1 и R_2 следующим образом: в множество R_1 включить те звезды из Z, у которых центры находятся на четных уровнях, а в множество R_2 поместить все остальные звезды из Z. Аналогично разбить множество ребер L на два подмножества R_3 и R_4. В множество R_3 поместить те ребра, у которых исходящие вершины находятся на четных уровнях, а в множество R_4 поместить все остальные ребра из L. Дополнив множества $R_1, \ldots, R_4$ недостающими вершинами из V, построить разбиения $P^1, \ldots, P^4$ соответственно. В качестве искомого выбрать разбиение $P \in \{P^1, \ldots, P^4\}$ с наибольшей стоимостью.

Утверждение 1. Разбиения $P^1, \ldots, P^4$ являются допустимыми для дерева T.

Доказательство. Поскольку дерево T является ориентированным, а в множество R_1 попали только те звезды, центры которых принадлежат четным уровням, то эти звезды не будут иметь общих вершин. Следовательно, каждый кластер разбиения P^i, $1 \leqslant i \leqslant 4$ является либо вершиной, либо ребром, либо звездой. Поэтому в силу условия (3) вес каждого кластера меньше либо равен B.

Утверждение 2. Для разбиений $P^1, \ldots, P^4$, генерируемых алгоритмом, справедливо соотношение $\sum_{i=1}^{4} w(P^i) \geqslant w(P^*)$, где P^* — оптимальное разбиение.

Доказательство. Пусть $P^* = (P_1^*, \ldots, P_k^*)$ — оптимальное разбиение дерева T. Каждая вершина $i \in V$ входит в некоторый кластер $P_{s_i}^*$ разбиения P^*. Обозначим через E_i^* множество ребер, исходящих из вершины i в ориентации, введенной в алгоритме, и вошедших в кластер $P_{s_i}^*$. Рассмотрим два случая.

1. Пусть для вершины i не выполняется условие (2). Тогда, очевидно, справедливо соотношение

$$w(Z_i) \geqslant w(E_i^*). \tag{5}$$

2. Пусть теперь для вершины i условие (2) выполняется. Если выполняются условия (3) и (4), то в Z_i включим ребро $(i, v_{k_i+1}) \in L$. Введем обозначения: $b = B - c(i)$, $c_j = w(i, v_j)$, $a_j = c(v_j)$ для всех $j \in \{1, \ldots, h_i\}$. Тогда п. 3 алгоритма есть не что иное, как градиентный алгоритм решения задачи о рюкзаке с булевыми переменными $\max\{\sum_{j=1}^{h_i} c_j x_j \mid \sum_{j=1}^{h_i} a_j x_j \leqslant b,\ x_j \in \{0, 1\},\ j \in \{1, \ldots, h_i\}\}$, поэтому, распространяя известные результаты о работе градиентного алгоритма на звезду Z_i, будем иметь

$$w(Z_i) \geqslant w(\widehat{E_i^*}), \tag{6}$$

где $\widehat{E}_i^*$ — множество ребер, исходящих из вершины i и отвечающих оптимальному заполнению емкости $B - c(i)$. Но поскольку вершины, соответствующие E_i^*, заполняют не обязательно полностью и оптимально емкость $B - c(i)$, то отсюда следует, что

$$w(\widehat{E}_i^*) \geqslant w(E_i^*). \tag{7}$$

Из неравенств (5), (6), (7) получаем

$$\sum_{i \in V} w(Z_i) \geqslant \sum_{i \in V} w(E_i^*). \tag{8}$$

Выражение, стоящее в левой стороне неравенства (8), задает стоимость всех ребер, вошедших в звезды Z_i и множество L, а в правой стороне — стоимость всех ребер, вошедших в оптимальное разбиение P^*.

Алгоритм назовем субоптимальным, если он для любого дерева T находит допустимое разбиение, удовлетворяющее условию $w(P) \geqslant \alpha w(P^*)$, где α — некоторая константа $(0 < \alpha \leqslant 1)$. Далее величину α будем называть гарантированной оценкой субоптимального алгоритма.

Из утверждения 1 и 2 следует

Теорема. Алгоритм является субоптимальным с гарантированной оценкой $\alpha = 1/4$.

Следствие. Если T — цепь или в T не существует вершины i, для которой выполняется условие (2), то гарантированная оценка алгоритма α равна 1/2.

Оценим сложность алгоритма. Однократное выполнение п. 3 алгоритма требует $O(h_i \log h_i)$ времени. Поскольку $\sum_{i=1}^{n} h_i = n - 1$, то общее время работы алгоритма — $O(n \log n)$. Очевидно, что объем требуемой памяти — $O(n)$.

Если изменить п. 3 алгоритма так, как это сделано в работе [4], используя линейный алгоритм поиска медианы, то получим линейный алгоритм по времени и памяти.

Список литературы

1. Lukes J. A.//IBM J. Res. Developmeht. 1974. № 18. P. 217.
2. Schrader R.// Discr. Appl. Math. 1983. V. 6. № 3. P. 301.
3. Ковалев М. М., Котов В. М.// Вестн. Белорусского ун-та. Сер. 1, физ., мат. и мех. 1986. № 3. С. 44.
4. Kundu S., Misra J.// SIAM J. Comput. 1977. V. 6. № 1. P. 151.

Поступила в редакцию 19.02.85.

УДК 517.948

Т. Г. ПАВЛОВА

ЯВНОЕ РЕШЕНИЕ ИНТЕГРАЛЬНОГО УРАВНЕНИЯ ПЕРВОГО РОДА С АЛГЕБРАИЧЕСКИМ ЯДРОМ СПЕЦИАЛЬНОГО ВИДА

Рассмотрим интегральное уравнение

$$\frac{1}{2\pi} \int_{\alpha}^{\beta} \left(1 + \frac{\sqrt{1-t^2}}{\sqrt{1-x^2}}\right) \frac{\varphi(t)}{t-x}\, dt = f(x), \quad \alpha < x < \beta, \tag{1}$$

в классе функций, интегрируемых вблизи точек α, β, ± 1, H-непрерывных на остальной части отрезка $[\alpha, \beta]$. В случае, когда $-1 \leqslant \alpha < \beta \leqslant 1$,