ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2019, том 59, № 5, с. 905–910

УДК 519.658

БЫСТРЫЙ АЛГОРИТМ РЕШЕНИЯ ПРОСТЕЙШЕЙ ЗАДАЧИ ПОИСКА

© 2019 г.
В. Н. Малозёмов[1,*], Г. Ш. Тамасян[1,**]

([1] 199034 С.-Петербург, Университетская наб., 7/9, С.-Пб гос. ун-т, Россия)

*e-mail: v.malozemov@spbu.ru;

**e-mail: g.tamasyan@spbu.ru

Поступила в редакцию 23.03.2018 г.
Переработанный вариант 11.01.2019 г.
Принята к публикации 11.01.2019 г.

Предлагается быстрый алгоритм решения задачи Данскина. Анализируется зависимость решения от параметров. Библ. 6. Фиг. 4. Табл. 1.

Ключевые слова: простейшая задача поиска, лемма Гиббса, быстрый алгоритм, зависимость решения от параметров.

DOI: 10.1134/S0044466919050107

1. ВВЕДЕНИЕ

Рассматривается простейшая задача поиска в следующей постановке. Пусть имеется X единиц времени для обнаружения некоторого объекта. Объект может находиться в одном из n мест, причем известна вероятность его нахождения в каждом из этих мест. Известна также вероятность обнаружения объекта, если он находится в i-м месте при всех i от 1 до n. Разумеется, эта вероятность зависит от времени поиска. Требуется разбить общее время поиска X на n частей (по числу мест возможного нахождения объекта) так, чтобы максимизировать вероятность обнаружения объекта. Частный случай этой задачи рассмотрел Дж. М. Данскин в своей книге, посвященной теории максимина [1, с. 22–24]. С помощью леммы Гиббса он вывел необходимое условие максимума и в общих чертах описал метод решения задачи. В данной заметке предложен быстрый алгоритм решения задачи Данскина и проанализирована зависимость решения от параметров.

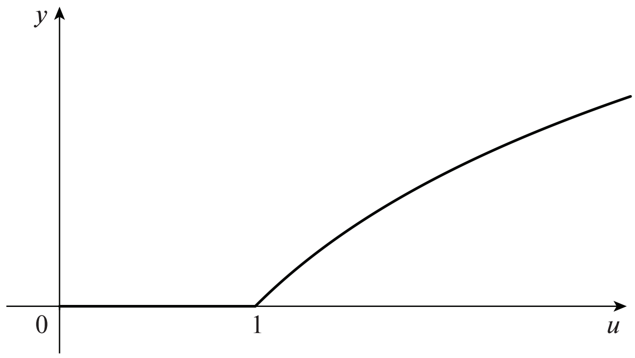
2. ПОСТАНОВКА ЗАДАЧИ

Пусть в нашем распоряжении имеется X единиц времени на поиск некоторого объекта. Этот объект может находиться в одном из n мест. Если объект находится в i-м месте, то, потратив x_i единиц времени, мы можем обнаружить его с вероятностью $1 - e^{-b_i x_i}$. Параметр b_i можно интерпретировать как уровень условий, благоприятствующих поиску объекта в i-м месте. Вероятность того, что объект находится в i-м месте, равна a_i. Как разбить общее время поиска X на сумму x_i так, чтобы максимизировать вероятность обнаружения интересующего нас объекта?

Приходим к экстремальной задаче, которую будем называть *задачей Данскина*:

$$\begin{gathered} F(x) := \sum_{i=1}^{n} a_i(1 - e^{-b_i x_i}) \to \max, \\ \sum_{i=1}^{n} x_i = X, \quad x_i \geq 0 \quad \forall i \in 1:n. \end{gathered} \tag{1}$$

Здесь a_i, b_i, X — положительные параметры.



Фиг. 1. График функции $y = \ln_+(u)$.

Целевая функция $F(x)$ строго вогнута на $\mathbb{R}^n$, а множество планов (векторов, удовлетворяющих ограничениям экстремальной задачи) компактно. Это гарантирует существование и единственность решения задачи (1).

3. УСЛОВИЯ ОПТИМАЛЬНОСТИ. РЕШЕНИЕ ЗАДАЧИ

По лемме Гиббса [1] (см. также препринт [2]) план $x = (x_1, \ldots, x_n)$ задачи (1) будет оптимальным тогда и только тогда, когда при некотором вещественном λ выполняются соотношения

$$a_i b_i e^{-b_i x_i} = \lambda, \quad \text{если} \quad x_i > 0, \tag{2}$$

$$a_i b_i \le \lambda, \quad \text{если} \quad x_i = 0. \tag{3}$$

Из (2) и (3) следует, что

$$x_i > 0, \quad \text{если} \quad a_i b_i > \lambda, \tag{4}$$

$$x_i = 0, \quad \text{если} \quad a_i b_i \le \lambda. \tag{5}$$

Действительно, при $a_i b_i \le \lambda$ необходимо $x_i = 0$, ибо в противном случае в силу (2) получим $a_i b_i > \lambda$, что противоречит исходному условию. При $a_i b_i > \lambda$ будет $x_i > 0$, ибо иначе получим противоречие с (3).

На основании (2) формулу (4) можно переписать в виде

$$x_i = \frac{1}{b_i} \ln \frac{a_i b_i}{\lambda}, \quad \text{если} \quad a_i b_i > \lambda. \tag{6}$$

Равенство (2) гарантирует положительность λ, поэтому условия $a_i b_i > \lambda$ и $a_i b_i \le \lambda$ равносильны тому, что $\frac{a_i b_i}{\lambda} > 1$ и $\frac{a_i b_i}{\lambda} \in (0,1]$. Данное замечание позволяет объединить формулы (6) и (5) следующим образом:

$$x_i = \frac{1}{b_i} \ln_+ \frac{a_i b_i}{\lambda}, \tag{7}$$

где

$$\ln_+(u) = \begin{cases} \ln(u) & \text{при} \quad u > 1, \\ 0 & \text{при} \quad u \in (0,1]. \end{cases}$$

График функции $y = \ln_+(u)$ представлен на фиг. 1.

Обозначим $c_i = \ln a_i b_i$, $u = \ln \lambda$ и покажем, что

$$x_i = \frac{1}{b_i}(c_i - u)_+, \quad i \in 1 : n. \tag{8}$$

Условия $\frac{a_i b_i}{\lambda} > 1$ и $\frac{a_i b_i}{\lambda} \in (0,1]$ равносильны соответственно неравенствам $c_i - u > 0$ и $c_i - u \le 0$. Согласно (7) получаем

$$x_i = \frac{1}{b_i}(c_i - u) \quad \text{при} \quad c_i - u > 0,$$
$$x_i = 0 \quad \text{при} \quad c_i - u \le 0,$$

что равносильно (8). Параметр u найдем из уравнения

$$\varphi(u) := \sum_{i=1}^{n} \frac{1}{b_i}(c_i - u)_+ = X. \tag{9}$$

Таким образом, решение задачи (1) сводится к решению скалярного уравнения (9) и вычислению компонент оптимального плана по формуле (8).

4. ОПИСАНИЕ АЛГОРИТМА

Опишем быстрый алгоритм решения уравнения (9).

Начнем с того, что упорядочим пары (b_i, c_i), $i \in 1:n$, по невозрастанию второй компоненты. Получим последовательность

$$(\hat{b}_1, \hat{c}_1), (\hat{b}_2, \hat{c}_2), \ldots, (\hat{b}_n, \hat{c}_n),$$

где $\hat{c}_1 \ge \hat{c}_2 \ge \ldots \ge \hat{c}_n$. Очевидно, что

$$\varphi(u) = \sum_{i=1}^{n} \frac{1}{\hat{b}_i}(\hat{c}_i - u)_+. \tag{10}$$

В частности,

$$\varphi(u) = 0 \quad \text{при} \quad u \ge \hat{c}_1; \tag{11}$$

при $u \in [\hat{c}_{k+1}, \hat{c}_k]$, $k \in 1:n-1$, имеем

$$\varphi(u) = \sum_{i=1}^{k} \frac{1}{\hat{b}_i}(\hat{c}_i - u) = \sum_{i=1}^{k} \frac{\hat{c}_i}{\hat{b}_i} - \left(\sum_{i=1}^{k} \frac{1}{\hat{b}_{ii}}\right) u; \tag{12}$$

наконец,

$$\varphi(u) = \sum_{i=1}^{n} \frac{\hat{c}_i}{\hat{b}_i} - \left(\sum_{i=1}^{n} \frac{1}{\hat{b}_i}\right) u \quad \text{при} \quad u \le \hat{c}_n. \tag{13}$$

В силу (12) получаем

$$\varphi(u) - \varphi(\hat{c}_k) = \left(\sum_{i=1}^{k} \frac{1}{\hat{b}_i}\right)(\hat{c}_k - u).$$

Обозначив

$$p_k = \sum_{i=1}^{k} \frac{1}{\hat{b}_i},$$

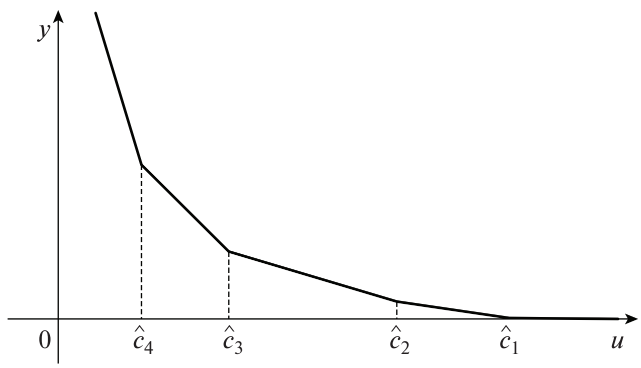
придем к формуле

$$\varphi(u) = \varphi(\hat{c}_k) + p_k(\hat{c}_k - u), \quad u \in [\hat{c}_{k+1}, \hat{c}_k], \quad k \in 1:n-1. \tag{14}$$

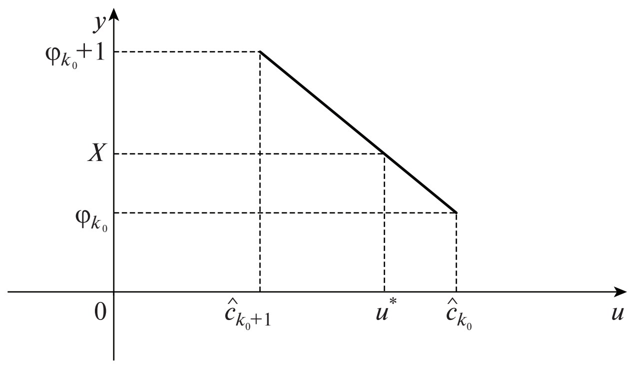
Аналогично из (13) следует, что

$$\varphi(u) = \varphi(\hat{c}_n) + p_n(\hat{c}_n - u), \quad u \le \hat{c}_n. \tag{15}$$

На основании (10), (11), (14) и (15) заключаем, что функция $\varphi(u)$ является непрерывной ломаной, выпуклой и строго убывающей от $+\infty$ до 0 на полуоси $(-\infty, \hat{c}_1)$ (см. фиг. 2).



Фиг. 2. График ломаной $y = \varphi(u)$.



Фиг. 3. Локализация решения u^* уравнения $\varphi(u) = X$.

Отметим, что ломаная $\phi(u)$ на отрезке $[\hat{c}_n, \hat{c}_1]$ полностью определяется своими значениями $\varphi(\hat{c}_k)$ в узлах $\hat{c}_k$, $k \in 1:n$. В силу (14) эти значения связаны формулой

$$\varphi(\hat{c}_{k+1}) = \varphi(\hat{c}_k) + p_k(\hat{c}_k - \hat{c}_{k+1}), \quad k \in 1:n-1.$$

Обозначив $\varphi_k = \varphi(\hat{c}_k)$, получим для последовательности $\varphi_1, \ldots, \varphi_n$ рекуррентное соотношение

$$\begin{gathered} \varphi_1 = 0, \quad p_0 = 0, \quad p_k = p_{k-1} + \frac{1}{\hat{b}_k}, \\ \varphi_{k+1} = \varphi_k + p_k(\hat{c}_k - \hat{c}_{k+1}), \quad k \in 1:n-1. \end{gathered} \tag{16}$$

Нас интересует решение u^* уравнения $\varphi(u) = X$. Из свойств функции $\varphi(u)$ следует, что такое решение существует и единственно. Для его нахождения будем последовательно вычислять значения $\varphi_1, \varphi_2, \ldots$ по формулам (16), пока не встретим индекс k_0, на котором

$$\varphi_{k_0} < X \le \varphi_{k_0+1}.$$

В этом случае u^* принадлежит отрезку $\left[\hat{c}_{k_0+1}, \hat{c}_{k_0}\right]$ (см. фиг. 3).

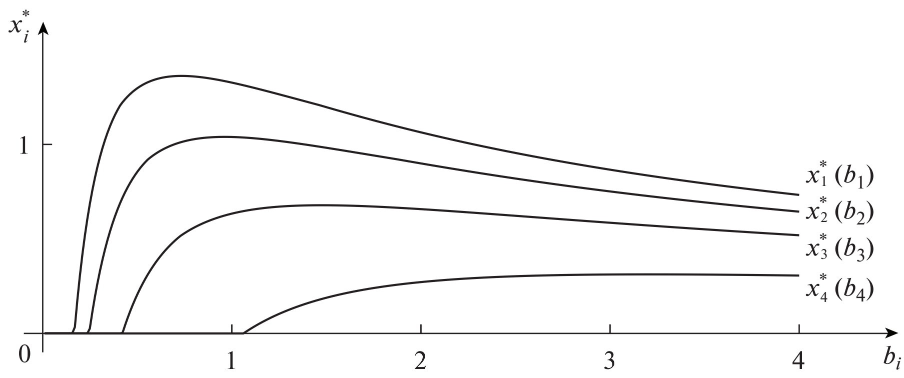
Согласно (14) имеем

$$u^* = \hat{c}_{k_0} - \frac{1}{p_{k_0}}(X - \varphi_{k_0}). \tag{17}$$

Если $\varphi_n < X$, то $u^* < \hat{c}_n$. В силу (15), u^* вычисляется по той же формуле (17) при $k_0 = n$.

На основании (8) для компонент решения $x^* = (x_1^*, \ldots, x_n^*)$ задачи (1) получаем представление

$$x_i^* = \frac{1}{b_i}(c_i - u^*)_+, \quad i \in 1:n.$$



Фиг. 4. Графики зависимости x_i^* от b_i.

Замечание 1. Описанный алгоритм сходится не более, чем за n шагов. И в общем случае требует сортировку элементов массива длиной n и не более $6n-2$ арифметических операций, т.е. имеет сложность порядка $O(n\log_2(n))$.

Замечание 2. Используя специфику алгоритма, можно минимизировать его трудоемкость. Действительно, для последовательного нахождения значений φ_k, $k = 1:n$, (см. (16)) необязательно иметь всю отсортированную по невозрастанию последовательность $\hat{c}_1 \geq \hat{c}_2 \geq \ldots \geq \hat{c}_n$. Достаточно находить лишь одно значение из этой последовательности для каждой следующей итерации (см. [3]).

Замечание 3. В работах [4], [5] предложены алгоритмы решения скалярного уравнения вида (9), имеющие линейный порядок сложности [6]. Однако численные эксперименты показали их малую эффективность с точки зрения временных затрат (см. [3]).

5. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

Напомним, что параметр b_i интерпретируется как уровень условий, благоприятствующих поиску объекта в i-м месте. Представляет интерес проанализировать влияние параметров b_i на оптимальное распределение $\{x_i^*\}$.

Возьмем конкретные данные. Пусть $n = 4$, $a_1 = 0.4$, $a_2 = 0.3$, $a_3 = 0.2$, $a_4 = 0.1$, $X = 3$. При $b_1 = b_2 = b_3 = b_4 = 1$ оптимальное распределение $x^* = (x_1^*, x_2^*, x_3^*, x_4^*)$ имеет вид

$$x^* = (1.327, 1.039, 0.634, 0).$$

Теперь при каждом фиксированном $i \in 1:4$ будем изменять значения b_i, сохраняя остальные b_k равными единице. На фиг. 4 изображены четыре графика, которые показывают, как изменяются x_i^* в зависимости от b_i.

Таблица 1

b_2	x_1^*	x_2^*	x_3^*	x_4^*
0.245	1.693	0	1	0.307
0.484	1.416	0.832	0.723	0.029
0.723	1.342	1.009	0.649	0
0.962	1.327	1.039	0.634	0
1.5	1.356	0.982	0.662	0
2.5	1.420	0.819	0.727	0.034
4	1.479	0.644	0.785	0.092

На каждом графике выделяются два значения параметра b_i:

$$b_i^0 \text{ – значение, до которого } x_i^* = 0;$$

$$b_i^1 \text{ – значение, после которого } x_i^* \text{ начинает убывать.}$$

В рассматриваемом случае имеем

$$b_1^0 = 0.167, \quad b_2^0 = 0.245, \quad b_3^0 = 0.421, \quad b_4^0 = 1.061,$$
$$b_1^1 = 0.734, \quad b_2^1 = 0.962, \quad b_3^1 = 1.475, \quad b_4^1 = 3.201.$$

В табл. 1 представлена более полная информация об изменении всего оптимального распределения x^* при изменении b_2.

СПИСОК ЛИТЕРАТУРЫ

1. *Данскин Дж.М.* Теория максимина и ее приложение к задачам распределения вооружения. Пер. с англ. М.: Сов. радио, 1970. 200 с.
2. *Малоземов В.Н., Тамасян Г.Ш.* Лемма Гиббса и ее приложения // Семинар “CNSA & NDO”. Избранные доклады. 10 октября 2017 г. URL: http://www.apmath.spbu.ru/cnsa/reps17.shtml#1010 (дата обращения: 01.03.2018).
3. *Tamasyan G., Prosolupov E.* Orthogonal projection of a point onto the standard simplex: algorithms analysis // Proc. 2015 Int. Conf. “Stability and Control Processes” in Memory of V. I. Zubov (SCP) (St. Petersburg, Russia, Oct. 5–9, 2015). IEEE, 2015. P. 353–356. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7342137
4. *Brucker P.* An $O(n)$ algorithm for quadratic knapsack problems // Oper. Res. Lett. 1984. V. 3. № 3. P. 163–166.
5. *Maculan N., Galdino de Paula G., Jr.* A linear-time median-finding algorithm for projecting a vector on the simplex of $\mathbb{R}^n$ // Oper. Res. Lett. 1989. V. 8. № 4. P. 219–222.
6. *Просолупов Е.В., Тамасян Г.Ш.* Оценка трудоемкости алгоритма по поиску нуля одной выпуклой кусочно-линейной функции // Дискрет. анализ и исслед. операций. 2018. Т. 25. № 2. С. 82–100.