

## ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

DOI 10.17223/20710410/10/8

УДК 519.857

### НОВЫЕ АЛГОРИТМЫ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ РЕСУРСА

В. И. Струченков

*Московский государственный институт радиотехники, электроники и автоматики,  
г. Москва, Россия*

**E-mail:** str1942@mail.ru

Предложены новые алгоритмы решения задачи оптимального распределения ресурса, использующие множества Парето и двусторонние прогностические оценки оптимума, получаемые по методу ветвей и границ.

**Ключевые слова:** *оптимальное распределение ресурса, множество Парето, метод ветвей и границ.*

#### 1. Постановка задачи

Рассмотрим задачу: найти минимум (или максимум) суммы

$$\sum_{i=1}^n g_i(x_i) \quad (1)$$

при ограничениях

$$\sum_{i=1}^n f_i(x_i) \leq R, \quad x_i \in X_i, \quad i = 1, 2, \dots, n, \quad (2)$$

где  $X_i$  — конечные множества;  $f_i(x_i) \geq 0$ ;  $g_i(x_i) \geq 0$  и  $R > 0$ . Целочисленность функций  $f_i(x_i)$  и  $g_i(x_i)$  необязательна. Предполагается, что множество (2) допустимых решений не пусто. В таком виде могут быть записаны различные задачи, сводящиеся к распределению заданного ресурса  $R$  между  $n$  потребителями. В задаче на максимум функции  $f_i(x_i)$  характеризуют ресурс, а  $g_i(x_i)$  — эффективность его использования [1]. При  $f_i(x_i) = p_i x_i$  и  $g_i(x_i) = c_i x_i$  получается задача оптимальной загрузки транспортного средства предметами, веса которых равны  $p_i$ , стоимости —  $c_i$ , а количества —  $x_i$  ( $x_i = 0, 1, 2, \dots$ ) [1]; если при этом  $x_i \in \{0, 1\}$ , то получается известная задача о ранце. В задаче на минимум функции  $f_i(x_i)$  — ресурс, а  $g_i(x_i)$  — затраты. Например,  $f_i(x_i)$  — ресурс времени, а  $g_i(x_i)$  — затраты на осуществление некоторого проекта. Или (как при проектировании оптимальной защиты поверхности [2])  $f_i(x_i)$  — допускаемый ущерб от неполной защиты  $i$ -го элемента поверхности, а  $g_i(x_i)$  — затраты на его защиту.

#### 2. Новые алгоритмы

Для решения задачи оптимального распределения ресурса предложено несколько алгоритмов [1–3], в частности, в [2] излагается следующий алгоритм, который является обобщением алгоритма Немхаузера — Ульмана, предназначенного для решения простейшей задачи о ранце [4].

Далее рассматривается задача (1), (2) на максимум. Пусть  $Q_m = \{(x_1, \dots, x_m) : (x_1, \dots, x_m) \in X_1 \times \dots \times X_m, \sum_{i=1}^m f_i(x_i) \leq R\}$ . Каждому вектору  $x = (x_1, \dots, x_m) \in Q_m$  сопоставляется пара чисел  $(F_m(x), G_m(x))$  вида  $F_m(x) = \sum_{i=1}^m f_i(x_i)$ ,  $G_m(x) = \sum_{i=1}^m g_i(x_i)$ . Из полученной совокупности пар, образующей мультимножество (так как могут существовать такие  $x, y \in Q_m$ , что  $(F_m(x), G_m(x)) = (F_m(y), G_m(y))$ ), выделяется его носитель  $N$  и на нём вводится следующее отношение частичного порядка:

$$(a, b) \succ (c, d) \Leftrightarrow (a \leq c \text{ \& } b \geq d).$$

Подмножество Парето в  $N$  составляют пары, максимальные относительно этого порядка, также называемые паретовскими точками. Алгоритм состоит в последовательном построении множеств Парето  $S_0, S_1, \dots, S_n$ , где  $S_0 = \{(0, 0)\}$ , а  $S_m = \{(F_m(x), G_m(x)) : x = (x_1, \dots, x_m) \in Q_m, (F_{m-1}(x_1, \dots, x_{m-1}), G_{m-1}(x_1, \dots, x_{m-1})) \in S_{m-1} \text{ и для любой пары } (F_m(y), G_m(y)) \in S_m \text{ либо } F_m(y) < F_m(x) \text{ и } G_m(y) < G_m(x), \text{ либо } F_m(x) < F_m(y) \text{ и } G_m(x) < G_m(y)\}\}$ .

Технически, для каждой пары  $(F_{m-1}(x), G_{m-1}(x)) \in S_{m-1}$  и таких значений  $a \in X_m$ , что  $(x \circ a) \in Q_m$ , где  $\circ$  — операция конкатенации, строятся векторы длины  $m$  и соответствующие им пары  $(F_{m-1}(x) + f_m(a), G_{m-1}(x) + g_m(a))$ . Если для очередной полученной пары в множестве  $S_m$  нет равных или больших нее, то она также добавляется в  $S_m$ , при этом все пары, которые меньше нее, исключаются из  $S_m$ . Решением задачи (1), (2) на максимум является вектор  $x \in Q_n$ , которому соответствует пара  $(F_n(x), G_n(x)) \in S_n$  с максимальным значением  $G_n(x)$ .

Для решения задач большой размерности также актуальна дополнительная отбраковка тех векторов, которым соответствуют бесперспективные паретовские точки. Предлагаемый в данной работе алгоритм (алгоритм отбраковки паретовских точек) делает это, используя идеи метода ветвей и границ.

Далее рассматривается задача (1), (2) на минимум. Обозначим затраты, соответствующие некоторому начальному приближенному решению, через  $E$  (в методе ветвей и границ они называются рекордом), а их нижнюю границу через  $H$ . Пусть теперь построено частичное решение  $x' = (x'_1, \dots, x'_m) \in Q_m$  и соответствующая ему точка  $(F_m(x'), G_m(x'))$  принадлежит  $S_m$ , т.е. является паретовской. Для этой точки рассмотрим задачу  $(P_m) : \sum_{i=m+1}^n f_i(x_i) \leq R - F_m(x'_1, \dots, x'_m), \sum_{i=m+1}^n g_i(x_i) \rightarrow \min$ . Пусть  $E_m$  —

затраты  $(\sum_{i=m+1}^n g_i(x_i))$ , соответствующие некоторому приближенному решению  $z$  задачи  $(P_m)$ , а  $H_m$  — нижняя граница затрат в решении  $(P_m)$ . Тогда условие отбраковки этой точки, ввиду ее очевидной бесперспективности, примет вид  $G_m(x') + H_m \geq E$ . Если же  $G_m(x') + E_m < E$ , то новое значение рекорда равно  $G_m(x') + E_m$ . Соответственно изменяется и вектор, задающий рекорд. Если  $E_m = H_m$ , то это значит, что приближенное решение  $z$  задачи  $(P_m)$  является точным, и её дальнейшее рассмотрение не требуется. Если при этом  $G_m(x') + E_m < E$ , то вектор  $(x' \circ z)$  запоминается и хранится до тех пор, пока не будет получено значение рекорда, меньшее чем  $G_m(x') + E_m$ . Если «рекорд устоит», то соответствующее ему решение и является оптимальным. Если при некотором  $m$  не останется ни одной паретовской точки, то рекорд является решением.

При каждом  $m$  можно корректировать нижнюю границу, вычисляя значение  $h = \min(G_m(x) + H_m)$ , где минимум берётся по всем оставшимся паретовским точкам из  $S_m$ , и заменяя  $H$  на  $h$  при  $h > H$ . Если не требуется точное решение, то счёт

прекращается при  $E - H < \varepsilon H$ , где  $\varepsilon$  определяется требуемой точностью решения задачи. В этом случае рекорд  $E$  и соответствующий ему вектор  $x$  дают приближённое решение задачи (1), (2). Заметим, что оценки  $H_m$  и  $E_m$  нужно получать для каждой паретовской точки для  $m = 1, 2, \dots, n-1$ . Эти оценки, как правило, получаются решением вспомогательной задачи. Для эффективности алгоритма в целом существенное значение имеют и качество этих оценок (их близость к оптимуму), и быстродействие алгоритма решения вспомогательной задачи. В задачах большой размерности число паретовских точек может быть велико, и время, затрачиваемое на дополнительные вычисления, может сводить к нулю эффект от дополнительной отбраковки паретовских точек.

### 3. Построение начального приближения и двусторонних оценок

Начальное приближение строится следующим образом: элементы  $x_i^j$  для  $i = 1, 2, \dots, n$  пронумеруем так, чтобы выполнялось условие  $f_i^1 < f_i^2 < \dots < f_i^{k_i}$ . Здесь  $x_i^j$  для  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, k_i$  — элементы конечных множеств  $X_i$ ;  $k_i = |X_i|$ ;  $f_i^j = f_i(x_i^j)$  и  $g_i^j = g_i(x_i^j)$  — соответствующие значения ресурса и затрат (в задаче на минимум). Другими словами, для каждого  $i$  значения  $x_i^j$  упорядочиваем по возрастанию ресурса; соответственно затраты должны строго убывать.

Получим:  $g_i^1 > g_i^2 > \dots > g_i^{k_i}$ . Если условие монотонности нарушается, то соответствующая точка  $(f_i(x_i^j), g_i(x_i^j))$  исключается из дальнейшего рассмотрения как бесперспективная (непаретовская) ещё на этапе предварительной обработки данных [5, с. 239].

Откладывая  $f_i^j$  по оси абсцисс, а  $g_i^j$  по оси ординат, получаем последовательность точек, которая определяет для каждого  $i$  строго монотонно убывающую кусочно-линейную функцию. Будем считать, что такие функции построены для всех  $i$ . Множество их графиков (ломанных линий) обозначим через  $L_1$ . Те из ломаных, которые не являются выпуклыми, заменим их выпуклыми оболочками. Полученное множество ломаных линий  $w_i(z_i)$ ,  $i = 1, 2, \dots, n$ , обозначим через  $L_2$ . Поскольку при замене невыпуклой ломаной её выпуклой оболочкой уменьшается число вершин, введём новые обозначения:  $z_i$  вместо  $f_i$  и  $w_i$  вместо  $g_i$ .

В результате получаем оценочную задачу: найти минимум суммы

$$\sum_{i=1}^n w_i(z_i) \quad (3)$$

при ограничениях

$$\sum_{i=1}^n z_i(x_i) \leq R, \quad x_i \in X_i, \quad i = 1, 2, \dots, n. \quad (4)$$

Крайние точки ломаных из  $L_2$  совпадают с крайними точками соответствующих ломаных из  $L_1$ . Абсолютные величины угловых коэффициентов звеньев ломаных  $w_i(z_i)$  будем называть уклонами. В силу выпуклости для каждой ломаной из  $L_2$  последовательность уклонов строго монотонно убывающая. Абсциссы концов звеньев ломаных из  $L_2$  обозначим через  $b_i^j$ , их множества — через  $B_i$ , а уклоны звеньев — через  $u_i^j$ .

Рассматриваем непрерывную задачу: на каждой ломаной из  $L_2$  найти точку с абсциссой  $z_i^*$  и ординатой  $w_i^*$  так, чтобы сумма абсцисс не превосходила заданный ресурс  $R$ , а сумма ординат была минимальной.

Оптимум непрерывной задачи (3), (4), который, очевидно, не более оптимума задачи (1), (2), примем в качестве искомой нижней границы. Решение непрерывной задачи (3), (4) можно найти следующим образом.

1. Фиксируем точку с минимальными значениями всех  $z_i$ , то есть на каждой ломаной берём начальную точку. Ей соответствует значение целевой функции

$$\sum_{i=1}^n w_i(f_i^1) = \sum_{i=1}^n g_i^1.$$

Это максимальное значение целевой функции при допустимых значениях переменных, то есть искомым абсцисс. Предстоит в  $n$ -мерном пространстве построить траекторию спуска так, чтобы, не нарушая ограничений, максимально уменьшить целевую функцию, то есть спуститься как можно ниже. Фиксируем остаток ресурса

$$T = R - \sum_{i=1}^n f_i^1.$$

2. Из всех звеньев всех ломаных из  $L_2$  выбираем звено с максимальным уклоном. Пусть это будет звено с номером  $j$  ломаной с номером  $s$ . На первом шаге  $j = 1$  в силу выпуклости ломаных. Будем менять только переменную  $z_s$ . Её увеличение даёт уменьшение целевой функции с наибольшей скоростью. Вычисляем шаг движения  $c = \min(b_s^{j+1} - b_s^j, T)$  и меняем  $z_s$  на  $z_s + c$ . Целевая функция уменьшится на  $u_s^j c$ , а оставшийся ресурс — на  $c$ .

3. Из оставшихся звеньев всех ломаных из  $L_2$  выбираем звено с максимальным уклоном и повторяем п. 2, пока оставшийся ресурс  $T$  не будет исчерпан.

Отметим, что в точке минимума все переменные  $z_s$ , кроме, быть может, той, которая менялась на последнем шаге, примут значения из соответствующих множеств  $B_s$ .

Если при выборе звена с максимальным уклоном таких звеньев оказывается несколько, то приоритет отдаётся звену с максимальной длиной, которое оставшийся ресурс позволяет использовать полностью (расход ресурса не превышает его текущий остаток), то есть  $b_s^{j+1} - b_s^j \leq T$ . При отсутствии такого звена среди звеньев с равными максимальными уклонами используется любое из них, например звено с наименьшей длиной. Попытка отказаться от использования максимальных уклонов в этой ситуации и разместить полностью какой-либо другой элемент приводит к неоправданному усложнению алгоритма и увеличению времени счёта

Если окажется, что и последняя из изменяемых переменных  $z_k$  в точке минимума примет значение из множества  $B_k$  ( $b_k^{j+1} - b_k^j = T$ ), то решение непрерывной задачи совпадает с решением исходной дискретной задачи, и на этом расчёт заканчивается. В противном случае полученное решение непрерывной задачи является начальным приближением, а значение целевой функции в точке минимума является искомой нижней границей  $H$ . Если последним рассматривалось звено ломаной, которая изначально была выпуклой, то приближённое решение исходной дискретной задачи получаем, аннулируя последний шаг (этим объясняется стремление сделать его как можно меньше). В противном случае по оптимальному значению  $z_k^*$  определяется абсцисса ближайшей слева вершины  $k$ -й ломаной из  $L_1$ . Ординату этой вершины обозначим через  $g_k^*$ . Нижняя граница  $H$  не изменяется, а рекорд становится равным  $E = H + g_k^* - w_k(z_k^*)$ .

Изложенный алгоритм даёт оптимальное решение непрерывной задачи за счёт выбора на каждом шаге максимального из всех оставшихся уклонов, то есть оптимального варианта использования ресурса. Чтобы в этом убедиться, рассмотрим первый шаг алгоритма. Если максимальный уклон  $u_s^1$  не использовать, то соответствующий ему ресурс  $b_s^2 - b_s^1$  будет исчерпан каким-то другим способом (изменится одна или несколько других переменных вместо  $z_s$ ), но при этом снижение затрат будет меньше.

На всех прочих шагах также нет смысла игнорировать максимальный уклон, так как очевидно, что именно он даёт наибольшее снижение затрат.

Другими словами, предположив, что в точке минимума непрерывной задачи  $z_s = z_s^1$ , приходим к противоречию, так как уменьшение абсциссы любой другой точки при соответствующем увеличении  $z_s$ , очевидно, даёт уменьшение целевой функции. Изложенный алгоритм можно использовать и при построении границ затрат на оставшихся шагах для каждой паретовской точки. Вместо  $R$  нужно взять оставшийся ресурс, начальная точка имеет координаты  $f_i^1$ ,  $i = r, r+1, \dots, n$ , где  $r$  — номер очередного шага. Значение целевой функции в этой точке равно

$$\sum_{i=1}^n w_i(f_i^1).$$

Для реализации алгоритма существенное значение имеет способ поиска наибольших уклонов. Простой способ состоит в сортировке всех уклонов всех звеньев ломаных из  $L_2$  как единого массива в порядке невозрастания. Возможны и более экономные способы, но мы не будем их рассматривать.

Аналогично может быть решена и задача (1), (2) на максимум. При этом алгоритм несколько изменяется: используются вогнутые оболочки вместо выпуклых, решение непрерывной задачи становится нижней границей и т. д. Однако, чтобы не разрабатывать новую компьютерную программу, можно преобразовать задачу на максимум в эквивалентную ей задачу на минимум, взяв в качестве целевой функции вместо

$$\sum_{i=1}^n g_i(x_i)$$

функцию

$$\sum_{i=1}^n (g_{i_{\max}} - g_i(x_i)),$$

где  $g_{i_{\max}}$  — максимальная эффективность для  $i$ -го потребителя ресурса. В вышеупомянутых частных случаях рассматриваемой задачи вычисления существенно упрощаются, так как при  $f_i(x_i) = p_i x_i$  и  $g_i(x_i) = c_i x_i$  для каждой ломаной все звенья имеют один уклон, а при  $x_i \in \{0, 1\}$  каждая ломаная состоит из одного звена.

Экспериментально установлено, что для задачи (1), (2) общего вида, особенно при вещественных значениях функций  $f_i(x_i)$ , алгоритм с отбраковкой паретовских точек по методу ветвей и границ даёт существенное снижение времени счёта по сравнению с отбраковкой только непаретовских точек.

Так, уже упоминавшаяся задача о защите поверхности [5] при числе элементов  $n = 400$  и числе способов защиты различных элементов  $k_i$ , равном 13, 42 или 45, решалась без отбраковки непаретовских точек по алгоритму «киевский веник» за 65 мин, при отбраковке только непаретовских точек за 12 с, а при дополнительной отбраковке паретовских точек по методу ветвей и границ за 2 с при снижении максимального (по шагам) числа точек с 11954 до 2609. Решение по методу регулярной сетки [1] с той же точностью потребовало более 3 ч, что объясняется необходимостью введения малого дискрета и соответственно большим числом дискретов сетки при вещественных и неравноотстоящих значениях ресурса. При этом из-за недостаточности оперативной памяти (512 Мбайт) подключается внешняя память, обмен с которой занимает много времени.

Дополнительно были выполнены расчёты на моделях с числом элементов 600 и способов защиты до 70. Установлено, что и в этом случае алгоритм дополнительной

отбраковки паретовских точек по методу ветвей и границ даёт существенное сокращение времени счёта по сравнению с отбраковкой только непаретовских точек, но это сокращение в значительной степени зависит от конкретных числовых данных при неизменных  $n$  и  $k_i$ .

Исследования свойств алгоритмов продолжаются, однако уже сейчас в целом можно констатировать, что новый алгоритм позволяет решать задачи распределения ресурса (1), (2) большой размерности много быстрее, чем традиционный алгоритм регулярной сетки [1] и алгоритм «киевский веник» [3] за счёт отбраковки бесперспективных вариантов (непаретовских и части паретовских точек) и всех их продолжений.

#### ЛИТЕРАТУРА

1. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. М.: Наука, 1965. 458 с.
2. Струченков В. И. Динамическое программирование с использованием множеств Парето // Дискрет. анализ и исслед. опер. 2008. Т. 15. № 6. С. 58–62.
3. Михалевич В. С. Последовательные алгоритмы оптимизации и их применение // Кибернетика. 1965. № 1. С. 45–46.
4. Кузюрин Н. Н., Фомин С. А. Эффективные алгоритмы и сложность вычислений. М.: МФТИ, 2007. 210 с.
5. Струченков В. И. Методы оптимизации в прикладных задачах. М.: Солон-Пресс, 2009. 310 с.