

## ОБРАБОТКА ИНФОРМАЦИИ

УДК 519.8

DOI: 10.17223/19988605/39/3

Д.П. Бураков, М.И. Гарина

АЛГОРИТМИЧЕСКИЕ АСПЕКТЫ ИССЛЕДОВАНИЯ СТРУКТУРЫ  
ПРЕДПОЧТЕНИЙ ЛПР С ИСПОЛЬЗОВАНИЕМ  
ТИПОВЫХ ОБОБЩАЮЩИХ ФУНКЦИЙ

Предлагается алгоритм сокращенного перебора точек, соответствующих объектам, характеризующимся несколькими признаками, для построения линейной выпуклой оболочки и Парето-оболочки множества этих объектов. Данный алгоритм используется при исследовании структуры предпочтений ЛПР с использованием типовых обобщающих функций.

**Ключевые слова:** линейная выпуклая оболочка; многомерное пространство; Парето-доминирование.

В задаче определения условий оптимальности некоторой альтернативы из дискретного множества, рассмотренной в предыдущем докладе авторов, решаются две нетривиальные подзадачи. Первая подзадача решается для определения потенциальной оптимальности альтернативы и заключается в определении её принадлежности  $\text{conv}_P(Y)$  – Парето-оболочке множества  $Y$ , получаемой объединением  $2^n$  множеств Парето  $Y_P$  для всех возможных сочетаний направлений оптимизации  $n$  критериев. Здесь  $Y$  – множество векторных оценок множества альтернатив  $X$ :  $y(x) = (f_1(x), f_2(x), \dots, f_n(x))$ ,  $x \in X$ ,  $y(x) \in Y$ ,  $|Y| = |X| = N$ . Тогда векторная оценка представляет собой точку в  $n$ -мерном пространстве (если все  $f_j(x)$  – числовые, то  $Y \subset \mathbb{R}^n$ ). Точки, входящие в  $\text{conv}_P(Y)$ , принадлежат поверхности тела (необязательно выпуклого) минимального объема, заключающего в себе множество  $Y$  целиком. В простейшем случае это тело представляет собой невыпуклый многогранник, вершинами которого являются точки  $\text{conv}_P(Y)$ .

Вторая подзадача решается для определения типа обобщающей функции  $F_O(w; y)$  и вектора весовых коэффициентов  $w$ , применение которых делает альтернативу  $y$  оптимальной, и заключается в определении принадлежности альтернативы  $\text{conv}_L(Y)$  – линейной выпуклой оболочке  $Y$ , которая представляет собой выпуклый многогранник минимального объема, обрамляющий  $Y$ . При этом в обе оболочки как минимум входят все точки  $y_{ex}$ , имеющие экстремальное значение хотя бы одной координаты:  $y_{ex} \in \text{conv}_L(Y) \cap \text{conv}_P(Y) \Leftrightarrow \exists j = 1, \dots, n: (f_j(x) = \max_{x \in X} (f_j(x)) \vee f_j(x) = \min_{x \in X} (f_j(x)))$ , кроме того,  $\text{conv}_L(Y) \subseteq \text{conv}_P(Y)$ .

Тогда возникает вопрос, как выбрать вершины и определить геометрические параметры этих многогранников и можно ли использовать их и методы аналитической геометрии для определения потенциальной оптимальности альтернативы, типа обобщающей функции и вектора весовых коэффициентов. Рассмотрению данных вопросов и уделяется внимание в данной статье.

## 1. Формирование Парето-оболочки

Авторы считают, что построение Парето-оболочки не может быть осуществлено алгоритмом меньшей сложности, чем сокращенный.

1. Для каждого из  $2^n$  вариантов направлений оптимизации  $n$  критериев нужно рассчитать множество  $Y_P$  Парето-точек путем их попарного сравнения. Вычислительная сложность определения множества Парето в худшем случае, при необходимости полного попарного сравнения, составляет  $n \cdot N \cdot (N - 1)/2 \sim n \cdot N^2$ , где  $N = |Y|$ . Это предполагает несравнимость всех имеющихся точек по Парето, и в результирующее множество

$Y_P$  при этом войдут все  $N$  точек. Однако в общем случае возможно сокращение перебора: если одна точка доминирует над другой, то доминируемую точку имеет смысл исключить из дальнейшего перебора и не использовать для сравнения с остальными. Таким образом, в лучшем случае вычислительная сложность составит  $n \cdot (N-1) \sim n \cdot N$ . При этом только одна доминирующая альтернатива будет сравниваться с остальными. Для максимального сокращения рекомендуется начинать перебор с точек, вошедших в множество  $\{y_{ex}\}$ , т.е. имеющих экстремальные значения хотя бы по одной координате.

2. Объединить множества  $Y_P$ . Вычислительной сложностью  $(N-1) \cdot n \sim n \cdot N$  можно пренебречь.

Таким образом, вычислительная сложность предлагаемого алгоритма построения  $\text{conv}_P(Y)$  составляет в худшем случае  $2^n \cdot n \cdot N^2$ , а в лучшем –  $2^n \cdot n \cdot N$ .

## 2. Определение потенциальной оптимальности альтернативы

Чтобы некоторая альтернатива обладала потенциальной оптимальностью, она должна принадлежать Парето-оболочке  $\text{conv}_P(Y)$ , т.е. находиться либо в одной из вершин, либо лежать на одной из её граней. В первом случае определение оптимальности тривиально и заключается в последовательном сравнении точки с вершинами, формирующими оболочку. Во втором случае следует определить уравнение каждой грани (гиперплоскости) и, подставляя координаты точки, выяснить, удовлетворяют ли они одному из них. Задача осложняется тем, что не каждые  $n$  точек  $\text{conv}_P(Y)$  формируют её грань и не каждая грань лежит только с одной стороны многогранника, так как он невыпуклый. Поэтому алгоритмически проще во время формирования многогранника Парето не исключать из множества вершин точки, лежащие точно на гранях. В этом случае оба варианта определения потенциальной оптимальности сводятся к однократному перебору вершин  $\text{conv}_P(Y)$ .

## 3. Формирование линейной выпуклой оболочки

Для построения выпуклой линейной оболочки  $\text{conv}_L(Y)$  в двумерном пространстве используют алгоритм Грэхема, Чана и Джарвиса (алгоритм «заворачивания подарка») и их модификации [1, 2], но в пространствах более высокой размерности они неприменимы. В литературе отсутствуют описания обобщения алгоритмов поиска выпуклых оболочек для случая  $R^n$  при  $n > 2$ .

Если бы заранее не была построена оболочка Парето  $\text{conv}_P(Y)$ , то, поскольку  $\text{conv}_L(Y)$  является минимальным выпуклым многогранником, содержащим  $Y$ , для его поиска следовало бы перебрать все гиперплоскости, проходящие через  $n$  точек из  $Y$ , т.е. сочетания по  $n$  точек из  $Y$ . После выбора очередной  $n$ -ки точек из  $Y$  необходимо проверить каждую из оставшихся  $N - n$  точек на то, с какой стороны от выбранной гиперплоскости она находится. Если для текущей гиперплоскости все точки  $Y$  находятся с одной стороны, то образующая её  $n$ -ка точек входит в  $\text{conv}_L(Y)$ . Взаимное положение гиперплоскости, проходящей через  $n$  точек, и  $n + 1$ -й определяется знаком определителя  $n$ -го порядка, построенного из разностей этих  $n + 1$  точек. Если все проверяемые  $N - n$  точек расположены с одной стороны от рассматриваемой гиперплоскости, то эти определители имеют одинаковый знак. Если же для  $n + 1$ -й точки этот определитель равен 0, то это означает, что она лежит в этой же гиперплоскости. При этом в самом худшем случае для построения  $\text{conv}_L(Y)$  необходимо вычислить  $C_N^n \cdot (N - n)$  определителей порядка  $n$ , а сложность вычисления каждого определителя  $\sim n^3$ .

Поэтому для построения  $\text{conv}_L(Y)$  предлагается алгоритм сокращенного перебора вершин многогранника  $\text{conv}_P(Y)$ , использующий тот факт, что  $\text{conv}_L(Y) \subseteq \text{conv}_P(Y)$ .

1. Для каждого из  $2^n$  вариантов направлений оптимизации  $f_j(x) \rightarrow \{\min, \max\}$ ,  $j = 1, \dots, n$ :

а. Выбирается  $n$  точек из  $\text{conv}_P(Y)$ , имеющих экстремальные значения хотя бы по одному критерию, и строится гиперплоскость  $L$ , проходящая через них. Сложность –  $n \cdot (N + n^2) \sim n^3$ . Кубическая компонента появляется в связи с необходимостью вычисления определителя для построения гиперплоскости.

б. Выбирается (или формируется искусственно) «предельная» точка  $y_{extr}$ , имеющая экстремальные значения по всем  $f_j(x)$ , и определяется её ориентированное расстояние  $d$  до гиперплоскости  $L$  [3]. Сложность –  $n \cdot (N + 1) \sim N \cdot n$ .

с. Путем перебора формируется множество точек  $Y^* \subseteq \text{conv}_P(Y)$ , ориентированное расстояние от которых до гиперплоскости имеет тот же знак, что  $d$ , а остальные отбрасываются. Сложность –  $N \cdot n$ .

d. Множество  $Y_i \subseteq Y^*$  формируется путем повторения пунктов а–с для всех наборов по  $n$  точек из  $Y^*$ . Если  $|Y^*| \leq n + 1$ , то  $Y_i = Y^*$ . Сложность в худшем случае составляет  $C_N^n \cdot (n^3 + N \cdot n)$ , а в лучшем – 0.

2. Для получения  $\text{conv}_L(Y)$  множества  $Y_i$  объединяются. Вычислительной сложностью можно пренебречь.

Получается, что вычислительная сложность предлагаемого алгоритма построения  $\text{conv}_L(Y)$  в худшем случае составляет  $2^n \cdot C_N^n \cdot n \cdot (n^2 + N)$ , а в лучшем –  $2^n \cdot n \cdot (n^2 + N)$ .

#### 4. Поиск вектора весовых коэффициентов, обеспечивающих оптимальность альтернативы

Для аддитивной и мультипликативной обобщающих функций, как показано в предыдущей работе, задача определения вектора весовых коэффициентов включает в себя поиск вектора, перпендикулярного гиперплоскости, являющейся гранью  $\text{conv}_L(Y)$  или  $\text{conv}_L(\ln(Y))$ .

Известно, что всякая гиперплоскость в  $\mathbf{R}^n$  проходит через  $n$  точек [3]. Уравнение гиперплоскости определяется как  $C_1 \cdot y_1 + \dots + C_n \cdot y_n + D = 0$ , где  $C_1, \dots, C_n$  – координаты нормали данной гиперплоскости. С другой стороны, гиперплоскость также описывается уравнением  $|(y - a_1 \dots a_{n-1} - a_1)^T| = 0$ , где  $a_1, \dots, a_n$  – точки, входящие в гиперплоскость, а  $y$  – произвольная точка пространства, тоже находящаяся на гиперплоскости. Разрешив это уравнение относительно  $y$ , найдем  $C_1, \dots, C_n$ . Для этого необходимо разложить указанный определитель по первой строке:

$$\begin{vmatrix} y - a_1 \\ \vdots \\ a_{n-1} - a_1 \end{vmatrix} = \begin{vmatrix} y_1 - a_{1,1} & \dots & y_n - a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n-1,1} - a_{1,1} & \dots & a_{n-1,n} - a_{1,n} \end{vmatrix} = (y_1 - a_{1,1}) \cdot A_{1,1} + \dots + (y_n - a_{1,n}) \cdot A_{1,n} = \\ = y_1 \cdot A_{1,1} + \dots + y_n \cdot A_{1,n} - (a_{1,1} \cdot A_{1,1} + \dots + a_{1,n} \cdot A_{1,n}) = 0.$$

Следовательно, алгебраические дополнения первой строки и есть компоненты нормального вектора. При этом они вычисляются через координаты точек  $a_1, \dots, a_{n-1}$ . Далее обратим внимание, что у всякой гиперплоскости существует не один нормальный вектор, а целое семейство векторов, задаваемое выражением  $\alpha \cdot \mathbf{n}$ , где  $\alpha \neq 0$ , а  $\mathbf{n}$  – любой из векторов, перпендикулярный гиперплоскости (например, вектор с координатами  $(A_{1,1}, \dots, A_{1,n})$ , как показано выше). Однако для правильного определения направления, важного для определения направлений оптимизации, нас интересует не всякий вектор, нормальный к указанной грани, а только такой, который выходил бы изнутри многогранника наружу перпендикулярно к указанной плоскости.

Таким образом, встают две задачи: 1) найти точку  $y_0$ , принадлежащую внутренности многогранника; 2) найти точку  $y_g$ , принадлежащую указанной грани, такую, что вектор  $(y_g - y_0)$  был бы перпендикулярен грани. Первую часть задачи можно решить двумя способами: если множество  $Y \setminus \text{conv}_L(Y)$  не пусто, в качестве  $y_0$  можно взять любую точку из него. В противном случае в качестве  $y_0$  можно взять точку, являющуюся арифметическим средним всех точек из  $\text{conv}_L(Y)$ : в силу выпуклости многогранника она будет всегда принадлежать его внутренности. Для нахождения точки  $y_g$  нужно решить систему уравнений (поиск точки пересечения прямой и гиперплоскости, при условии что прямая задана в параметрической форме, а ее направляющий вектор является нормальным вектором гиперплоскости):

$$\begin{cases} y_{g1} = y_{01} + A_{1,1} \cdot t, \\ \dots \\ y_{gn} = y_{0n} + A_{1,n} \cdot t, \\ (A_{1,1}^2 + \dots + A_{1,n}^2) \cdot t - (A_{1,1} \cdot a_{1,1} + \dots + A_{1,n} \cdot a_{1,n}) = 0. \end{cases}$$

Здесь  $y_{g1}, \dots, y_{gn}$  – координаты искомой точки  $y_g$ ;  $y_{01}, \dots, y_{0n}$  – координаты внутренней точки  $y_0$ ;  $a_{1,1}, \dots, a_{1,n}$  – координаты точки  $a_1$ , являющейся вершиной грани. Числа  $A_{1,1}, \dots, A_{1,n}$  – координаты нормального

вектора гиперплоскости, найденные выше. Найдя число  $t$ , найдем и координаты искомой точки  $y_g$ , а вычитанием  $(y_g - y_0)$  найдем искомый вектор.

Однако всякая вершина многогранника принадлежит одновременно  $k \geq 1$  гиперплоскостям. Поэтому если нас интересует такой вектор  $w$ , который доставляет этой точке наибольшее значение, то этот вектор должен находиться во внутренности конуса, образованного нормальными векторами этих гиперплоскостей. Но тогда  $w$  должен представляться как их выпуклая комбинация. В простейшем случае будем брать средний вектор, т.е. для  $k$  граней  $w = (w_{\perp gr1} + \dots + w_{\perp grk}) / k$ . Тем самым мы можем перейти от систем неравенств к уравнениям, ищущим для каждой гиперплоскости нормальный вектор, а зная для точки, в какие грани она входит, образовать из них конус и взять «средний» внутренний вектор-луч этого конуса.

## Заключение

В работе показано, что задача поиска вектора весовых коэффициентов  $w$ , оптимизирующего указанную альтернативу по значению аддитивной или мультипликативной обобщающей функции (при условии ее принадлежности паретовской или линейной оболочке множества), алгоритмически разрешима. Кроме того, приведены алгоритмы, позволяющие построить эти оболочки для произвольного конечного множества в  $R^n$  эффективнее, чем полным перебором.

## ЛИТЕРАТУРА

1. Микони С.В. Теория принятия управленческих решений. СПб. : Лань, 2015. 448 с.
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. 3-е изд. СПб. : Вильямс, 2015. 1328 с.
3. Ильин В.А., Позняк Э.Г. Аналитическая геометрия. М. : ФИЗМАТЛИТ, 2002. 240 с.

**Бураков Дмитрий Петрович**, канд. техн. наук. E-mail: bds@yandex.ru

**Гарина Марина Игоревна**, канд. техн. наук. E-mail: migarina@gmail.com

Петербургский государственный университет путей сообщения Императора Александра I

Поступила в редакцию 14 марта 2016 г.

*Burakov Dmitry P., Garina Marina I.* (Saint-Petersburg State Transport University, Russian Federation).

**Algorithmic aspects to analysis of decision maker's preferences structure with standard aggregative functions.**

**Keywords:** linear convex envelope; multidimensional space; Pareto dominance.

DOI: 10.17223/19988605/39/3

There are two non-trivial tasks in the considered in the previous paper algorithm of checking if some object is optimal. First task is about potential ability for an object to be optimal. It solves by checking if an object belongs to  $\text{conv}_P(Y)$  – Pareto-envelope of  $Y$  set, obtained as a union of  $2^n$  Pareto-sets  $Y_P$  for all combinations of directions for  $n$  criteria. Second task is about to determine the type of aggregative function and weight vector which make an object optimal. It contains the checking if an object belongs  $\text{conv}_L(Y)$  – linear convex envelope  $Y$ . Here  $Y$  is a set of vectors which characterizes the set of objects  $X$ :  $y(x) = (f_1(x), f_2(x), \dots, f_n(x))$ ,  $x \in X$ ,  $y(x) \in Y$ . Any vector is a point in  $n$ -dimensional space  $R^n$ , and envelopes  $\text{conv}_P(Y)$  and  $\text{conv}_L(Y)$  are the general form polytope and the convex polytope, framing  $Y$ . Both of polytopes contains at least all points  $y_{ex}$  having extreme value of at least one of coordinates:  $y_{ex} \in \text{conv}_L(Y) \cap \text{conv}_P(Y) \Leftrightarrow \exists j = 1, \dots, n: (f_j(x) = \max_{x \in X} (f_j(x)) \vee f_j(x) = \min_{x \in X} (f_j(x)))$ . And also  $\text{conv}_L(Y) \subseteq \text{conv}_P(Y)$ . We declare then the Pareto envelope cannot be found with an algorithm less complex then reduced search:

1. Calculating the Pareto set for any of  $2^n$  combinations of directions of  $n$  criteria using the pairwise comparisons. The computational complexity of calculating the Pareto set is  $n \cdot N \cdot (N-1)/2 \sim n \cdot N^2$  at worse case,  $N = |Y|$ . That worse case is if all objects are Pareto incomparable. In that case, all  $N$  objects will be in Pareto set  $Y_P$ . However, in general, there is a possibility of reducing the calculating: if one object is dominated by another, we must exclude that dominated object from the further search. Therefore, in the best case the computational complexity will be  $n \cdot (N-1) \sim n \cdot N$ . In this case, we have one best object, which we will compare with others. We recommend to start with objects from the set  $\{y_{ex}\}$  in order to provide the maximal reducing.

2. Unioning of  $Y_P$ . Computational complexity  $(N-1) \cdot n \sim n \cdot N$  is less then above and it can be neglected.

So approximately computational complexity of  $\text{conv}_P(Y)$  construction algorithm is  $2^n \cdot n \cdot N^2$  at worse case and  $2^n \cdot n \cdot N$  at best case. For linear convex envelope  $\text{conv}_L(Y)$  construction in two-dimension space we can use the Graham scan and its modifications, but with quantity of dimensions  $n > 2$  it is not applicable. Instead of Graham scan we consider the following reducing search algorithm using the fact that  $\text{conv}_L(Y) \subseteq \text{conv}_P(Y)$ .

1. For all of  $2^n$  combination of directions  $f_j(x) \rightarrow \{\min, \max\}$ ,  $j = 1, \dots, n$ :

- a. Choose one object from  $\text{conv}_P(Y)$ , having extreme value at least of one criteria and construct the hyperplane  $L$  passing through them. The complexity is  $n \cdot (N + n^2) \sim n^3$ . The cubical component is from need to calculate the determinant for hyperplane construction.
  - b. Choose or form the new “limited” point  $\mathbf{y}_{\text{extr}}$ , having extreme values of all  $f_j(x)$  with given combination of directions and calculate the oriented distance  $d$  between it and the hyperplane. The complexity is  $n \cdot (N + 1) \sim N \cdot n$ .
  - c. Form the  $Y^* \subseteq \text{conv}_P(Y)$  from objects for which the oriented distance to hyperplane has the same sign that  $d$ , another neglected. The complexity is  $N \cdot n$ .
  - d. The set  $Y_t \subseteq Y^*$  forming by iterative repeating  $a-c$  for all sets of  $n$  points from  $Y^*$ . If  $|Y^*| \leq n + 1$ , then  $Y_t = Y^*$ . The complexity at the worse case is  $C_N^n \cdot (n^3 + N \cdot n)$ , and at the best – 0.
2. The envelope  $\text{conv}_L(Y)$  is union of all  $Y_t$ . The complexity is less then above and can be neglected.
- So approximately computational complexity of  $\text{conv}_L(Y)$  constructing algorithm is  $2^n \cdot C_N^n \cdot n \cdot (n^2 + N)$  at the worse case and  $2^n \cdot n \cdot (n^2 + N)$  at the best.

## REFERENCES

1. Mikoni, S.V. (2015) *Teoriya prinyatiya upravlencheskikh resheniy* [Theory of Managerial Decision-Making]. Saint-Petersburg: Lan.
2. Kormen, T., Leisserson, C., Rivest, R. & Stein, K. (2015) *Algoritmy. Postroenie i analiz* [Introduction to Algorithms, CLRS]. Translated from English by I.V. Krasikov. St. Petersburg: Williams.
3. Ilyin, V.A. & Poznyak, E.G. (2002) *Analiticheskaya geometriya* [Analitical Geometry]. Moscow: PhisMathLit.