# Эффективный Алгоритм Вычисления Бинарного Эпсилон Индикатора, Основанный на Поиске Минимума в Ортанте

Васин А. Ю., магистрант кафедры компьютерных технологий Университета ИТМО, vasinandrey2010@gmail.com  
Буздалов М. В., доцент кафедры компьютерных технологий Университета ИТМО, mbuzdalov@gmail.com

**Аннотация**

Бинарный *ε*-индикатор часто используется для оценки качества решений в многокритериальной оптимизации, а также в проведении самих оптимизаций.

Мы представляем эффективный алгоритм вычисления значения *ε*-индикатора, который сводит задачу к серии поисков минимума в ортанте. Для последнего мы рассматриваем две реализации: одна основана на древовидной структуре данных, другая основана на подходе «разделяй и властвуй».

# Введение

# Многие реальные оптимизационные задачи являются многокритериальными, то есть, они требуют максимизации или минимизации нескольких критериев, которые часто конфликтуют. В этой постановке исследователи часто хотят узнать множество Парето-оптимальных решений задачи.

# В многокритериальной оптимизации *ε*-индикатор это функция из одного или более множеств решений в единственное число. Индикаторы в основном используются для двух целей: для оценки качества множества решений, которая особенно полезна при сравнении выходов различных оптимизаторов [1] и для самой оптимизации [2].

Формально *аддитивный бинарный ε-индикатор* это функция от двух множеств точек и , которая возвращает наименьшее значение *ε*, возможно отрицательное, которое нужно прибавить (если критерии оптимизации должны быть максимизированы или иначе вычесть) к каждой координате каждой точки из множества , так что каждая точка множества нестрого Парето-доминирована хотя бы одной точкой .

Одно из привлекательных свойств *ε*-индикатора заключается в том, что его определение подразумевает простой алгоритм с малой константой реализации (мы обозначаем количество критериев оптимизации как ). Однако когда количество точек увеличивается (скажем, ), этот алгоритм становится медленным даже для . Данная работа нацелена на улучшение сложившейся ситуации.

# Определения

Без потери общности предположим, что мы решаем многокритериальную задачу минимизации с количеством критериев равным . В данном случае отношение Парето-доминирования определяется на двух точках в пространстве критериев оптимизации следующим образом:

где называется *строгим доминированием* и — *нестрогим доминированием*.

*Аддитивный бинарный ε-индикатор*, или *ε*-индикатор для краткости, определен на двух множествах точек *M* и *F* и равен наименьшему значению *ε*, на которое нужно сдвинуть *M* в сторону оптимальности так, что каждая точка из *F* нестрого доминирована хотя бы одной точкой из *M*. В случае минимизации сдвиг точек в сторону оптимальности на *ε* будет равносилен вычитанию *ε* из каждой координаты точки. Формальное определение *ε*-индикатора будет иметь следующую форму:

Мы заканчиваем данную секцию необходимыми определениями, касающимися поиска минимума в ортанте. *Ортант* это часть *k*-размерного пространства, которая состоит из пересечения *k* полупространств, где *i*-ое такое полупространство определено неравенством или , где это некоторая константа. Ортант является естественным обобщением луча в одномерном случае, квадранта в двумерном и октанта в трехмерном. В данной статье мы рассматриваем только ортанты, ориентированные в сторону положительной бесконечности, то есть, имеющие вид .

Вычислительная задача *поиска в ортанте* рассматривает множество точек в *k*-размерном пространстве, часто с некоторыми ассоциированными значениями. В данной задаче необходимо отвечать на запросы связанные с ортантами, которые обычно ориентированы в сторону положительной или отрицательной бесконечности. Виды запросов включают в себя поиск *m* произвольных точек, принадлежащих ортанту, или поиск суммы, минимума или максимума значений, ассоциированных с точками, принадлежащими ортанту.

# Сведение к поиску минимума в ортанте

Для эффективного вычисления *ε*-индикатора, заметим, что значение эпсилон для каждой точки из *F* может быть вычислено независимо от других точек:

,

и затем следует найти максимум из них:

Для вычисления , мы можем разбить *M* на произвольные подмножества такие, что , решить задачу независимо для каждого подмножества и взять минимум. Заметим, что разбиение на подмножества может зависеть от *f* произвольно. Мы определяем как множество точек *m* такое, что . Тогда определение может быть переписано как:

Следующим шагом будет являться осознание того, что собой представляют множества . Для выбранной точки *f* и любой точки мы можем записать следующие неравенства на основе определения :

что эквивалентно:

(1)

Определим проекцию следующим образом:

где равная нулю координата не включена в получившуюся точку.

Используя проекцию , мы можем переписать (1) как:

в итоге получаем:

Внутренний минимум ничто иное, как результат поиска минимума в ортанте для множества точек , где значение связано с точкой . Так как проекция строится одинаково для всех точек , мы можем построить одно множество проекций точек для каждого *j* и выполнить поиск для всех *f*.

Описанный алгоритм может работать с любым алгоритмом для поиска минимума в ортанте. Ключевым фактом (который имеет положительное влияние на производительность) является то, что все точки и все запросы известны заранее, так что может быть использован эффективный офлайн алгоритм.

# Алгоритм на основе динамических деревьев

В [3, Теореме 3.3] приведена структура данных для *d*-размерного поиска минимума в ортанте, которая поддерживает только операцию активации для модификации структуры, и она может быть реализована с использованием времени и памяти на предподготовку, суммарного времени активации и времени на единичный запрос. Такая структура данных использует деревья ван Эмде Боаса [4] для достижения времени работы для наименьших измерений. Однако эти деревья или имеют огромный дополнительный расход памяти, или требуют сжатия хранимых значений в небольшие диапазоны целых чисел.

Хотя мы находим упомянутые выше проблемы решаемыми, мы выбрали более простую структуру данных дерево Фенвика [5]. Сложность необходимых операций в нем составляет , и оно весьма оптимально во времени работы и занимаемой памяти на современных компьютерах. Итоговая структура данных требует времени на предподготовку и активацию, памяти и времени на единичный запрос.

Для вычисления значения *ε*-индикатора для *k*-размерных точек, мы производим *k* различных *(k-1)-*размерных офлайн поисков минимума в ортанте. Общее время работы .

# Алгоритм «разделяй и властвуй»

Чтобы понять идею работы подхода «разделяй и властвуй» для данной задачи, рассмотрим рис. 1. В данном двумерном рисунке белые точки соответствуют запросам, а черные точки соответствуют точкам данных. Вертикальная прямая разбивает множество белых точек *Q* на два множества и , и множество черных точек *P* на два множества и таким образом, что:

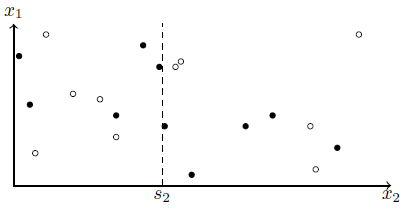


Рисунок 1: Иллюстрация подхода «разделяй и властвуй» для поиска минимума в ортанте

Заметно, что для любой точки запроса , можно рассматривать только точки , так как для каждой верно, что и *p* не покрывается запросом *q*. Так же для каждого запроса и каждой точки верно, что , так что алгоритм может больше не проверять данное условие.

Если мы вызовем нашу процедуру , где *d* текущее измерение, то она будет состоять после поиска медианы (значение в примере выше) из проведения разбиений и трех рекурсивных вызовов, а именно, , , . Если записать время работы как , мы можем оценить его следующим образом:

что при известном дает нам .

# Эмпирическая оценка

Мы эмпирически оценили предложенные алгоритмы с поисками минимума в ортанте, а также наивного алгоритма для вычисления *ε*-индикатора. Мы использовали размерности в интервале . Также мы использовали сдвигаемое и фиксированное множества одинакового размера. Эти размеры были выбраны из набора

Из анализа результатов мы заметили, что предложенный алгоритм с обеими версиями алгоритма поиска минимума в ортанте имеет лучшую асимптотику по сравнению с наивным алгоритмом.

Для размерностей и предложенный алгоритм превосходит наивный алгоритм для всех рассматриваемых размеров задачи, хотя для и времена работы почти совпадают. Для этих размерностей не наблюдается заметной разницы между алгоритмами поиска минимума в ортанте.

Для больших размерностей можно заметить два явления. Во-первых, чем выше размерность, тем хуже поведение реализации предложенного алгоритма с использованием динамического дерева по сравнению с реализацией на основе подхода «разделяй и властвуй». Во-вторых, предложенный алгоритм (здесь мы рассматриваем реализацию на основе подхода «разделяй и властвуй») начинает превосходить наивный алгоритм лишь с какого-то конкретного размера задачи: с приблизительно для , с для , и с для .

# Заключение

Мы представили алгоритм для эффективного вычисления аддитивного бинарного *ε*-индикатора. Данный алгоритм основан на сведении этой проблемы к серии поисков минимума в ортанте. Для проведения последнего мы использовали два подхода: один из них основан на классической древовидной структуре данных для ортанта и запросов на интервале, второй основан на многомерном подходе “разделяй и властвуй”.

Литература

1. Performance assessment of multiobjective optimizers: An analysis and review / E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, V. Grunert da Fonseca. // IEEE Transactions on Evolutionary Computation — 2003 — С. 117–132.
2. Indicator-based selection in multiobjective search / E. Zitzler, S. Künzli // Parallel Problem Solving from Nature PPSN VIII number 3242 in Lecture Notes in Computer Science — 2004 — С. 832–842.
3. Scaling and related techniques for geometry problems / H. N. Gabow, J. L. Bentley, R. E. Tarjan // Proceedings of the sixteenth annual ACM symposium on Theory of computing — 1984 — С. 135–143.
4. Design and implementation of an efficient priority queue / P. Van Emde Boas, R. Kaas, E. Zijlstra // Mathematical Systems Theory — 1976 — С. 99–127.
5. A new data structure for cumulative frequency tables / P. M. Fenwick // Software: Practice and Experience — 1994 — С. 327–336.