УДК 519.85

ОПТИМАЛЬНАЯ ФИЛЬТРАЦИЯ МНОЖЕСТВА ПАРЕТО

© 2004 Ю. В. Бугаев

Воронежская государственная технологическая академия

Решается задача оптимальной фильтрации набора Парето-оптимальных альтернатив, т.е. построения подмножества максимально удаленных друг от друга точек. Исследуется возможность точного решения задачи, предлагается эффективный приближенный алгоритм.

ВВЕДЕНИЕ

Задачи многокритериальной оптимизации решаются обычно в два этапа. На первом происходит построение некоторого конечного набора эффективных альтернатив (конечная аппроксимация множества Парето), из которого на следующем этапе с помощью неформальных методов выбирают оптимальное решение. Численные методы построения множества неулучшаемых решений непрерывных задач можно разбить на две группы. Процедуры первой группы осуществляют поочередный поиск отдельных точек, как решений вспомогательных скалярных задач (методы скаляризации). Методы второй группы призваны обеспечить аппроксимацию множества неулучшаемых оценок в целом.

Применение методов второй группы не дает возможности контролировать густоту полученной сети. Приемлемая точность аппроксимации достигается при очень большом числе решений. Использование же методов скаляризации допускает получение совпадающих или очень близких точек в пространстве критериев. Подобная ситуация весьма затрудняет сравнение и выбор альтернатив на втором этапе решения, поскольку эксперт не в состоянии различить две точки, близкие в пространстве оценок. В связи с этим, после нахождения аппроксимирующего набора эффективных точек, возникает необходимость его фильтрации.

Пусть имеем конечное множество U, $|U|=n$, на элементах которого определена метрика $\rho(x,y)$. Задача оптимальной фильтрации, т.е. построение набора максимально удаленных друг от друга точек, изложенная в [1], имеет две формальных постановки.

1. При заданной величине порога H построить подмножество $U' \subset U$ максимальной мощности, элементы которого удаленны друг от друга в метрике ρ не менее чем на H.

2. Построить подмножество $U' \subset U$ заданной мощности K, для элементов которого минимальное удаление друг от друга в метрике ρ максимально.

Рассмотрим следующую распознавательную задачу [2] (обозначим ее задача L) : существует ли при заданном натуральном H на конечном множестве U такое подмножество $U' \subset U$ мощности K, что выполняется условие

$$\forall x, y \in U' \rho(x,y) \geq H.$$

Ранее автором было доказано [3], что задача L является NP-полной. Поскольку она не проще оптимальной фильтрации, то существование эффективного алгоритма точного решения последней, как и любой NP-полной задачи, проблематично, и на практике необходимо использовать приближенные методы, либо следует выделить класс подзадач, допускающих точное решение эффективными методами.

1. ФИЛЬТРАЦИЯ МНОЖЕСТВА ПАРЕТО. ТОЧНОЕ РЕШЕНИЕ ЗАДАЧИ.

При небольшой мощности исходного набора эффективных точек задача оптимальной фильтрации может быть решена точно с использованием известных методов построения максимального независимого множества на графе, например, с помощью алгоритма Брона—Кэрбоша (алгоритм БК) [4], относящегося к классу алгоритмов с возвратом [5].

Пусть U — исходное множество альтернатив, $|U|=n$, и H — величина порога. Построим соответствующий граф $G=(U,E)$, где ребра определяются по следующему правилу: для любой пары вершин $x,y\in U$ положим

$$(x,y)\in E, \text{ если } \rho(x,y)\le H;$$

$$(x,y)\notin E \text{ в противном случае.}$$

При прямом ходе алгоритма БК строится последовательность вложенных множеств $Q_k\subset U$, из которых на каждом шаге отбираются несмежные между собой вершины. При обратном ходе часть вершин возвращается в Q_k для построения новой ветви с другой последовательностью множеств Q_k. Упрощенная версия алгоритма БК имеет следующий вид:

```
Begin k := 0; S := ∅; Q := U;
  While Q[0] <> ∅ Do
  Begin While Q[k] <> ∅ Do
    Begin x := top(Q_k); x → S;
      Q[k+1] := Q[k] \ (СПИСОК [x] ∪ {x});
      k := k+1;
    End;
    y ← S; k := k−1;
    Q[k+1] := Q[k] \ {y}; top(Q[k]) → S;
  End;
End.
```

Здесь $\text{top}(Q[k])$ — первый элемент множества Q_k; операция $x\to S$ означает запись элемента x в стек S. После каждого завершения внутреннего цикла в S будут находиться вершины максимального независимого множества.

Рассмотрим процесс построения самого первого множества S при первом выполнении внутреннего цикла. Назовем этот фрагмент полного алгоритма БК алгоритмом БК1. Очевидно, его вычислительная сложность имеет порядок $O(n)$. Исследуем возможность его использования для построения максимального независимого множества на графе частного вида.

Сделаем допущение (**Допущение 1**), что для любых вершин $u,v,w\in U$, таких, что $u<v<w$ из включения $(u,v)\in E$ следует $(u,v)\in E$ и $(v,w)\in E$. Вопрос о корректности такого допущения рассмотрим позже.

Пусть $Q\subseteq U$ — текущее множество вершин, полученное алгоритмом БК1 на некотором шаге, $E(Q)\subseteq E$ — множество ребер, таких, что из $(u,v)\in E(Q)$ следует $u,v\in Q$. Выберем какую-либо вершину $u\in Q$, исключим из Q вершины с номерами меньшими, чем u и рассмотрим u в качестве начальной вершины, используемой для построения максимального независимого множества на подграфе $(Q,E(Q))$ с помощью алгоритма БК1. Назовем построенное множество S траекторией из u по множеству Q, длиной траектории назовем мощность S.

Лемма 1. Пусть $u<v$. Тогда траектория из u по любому множеству Q не может быть короче траектории из v по тому же множеству Q.

Доказательство. Проведем его по индукции. Пусть траектория из v по Q состоит из единственной точки и, следовательно, имеет длину 1. Очевидно, что длина траектория из u не может быть меньше, а, значит, лемма верна.

Предположим, что лемма верна, когда длина траектория из v меньше, или равна k. Докажем ее справедливость при длине траектории $k+1$.

Пусть v — начальная вершина траектории. Сделаем один шаг алгоритма БК1, и пусть следующей вершиной траектории будет v'. Аналогично, пусть u' — следующая вершина траектории, если u — начальная.

Ясно, что $u'\le v'$, в противном случае имеем $u<v<v'\le u'-1<u'$. Вершина с номером $u'-1$ должна быть исключена при построении траектории с началом в u, а, значит, в подграфе $(Q,E(Q))$ существует дуга $(u,\ u'-1)$. В силу сделанного допущения, должны существовать дуги (u,v') и $(v',u'-1)$. Следовательно, v' смежна с v и при построении траектории с началом в v, должна быть исключена алгоритмом БК1.

Но длина траектория из v' равна k, а, следовательно, по предположению индукции, траектория из u' не может быть короче. Отсюда длина траектория из u не может быть меньше $k+1$. Лемма доказана.

При реализации полной версии алгоритма БК траектории прокладываются по различным множествам Q, полученных с помощью операции $Q := Q \setminus \{y\}$, которую назовем модификацией множества Q. Рассмотрим две различных траектории из произвольной вершины x по множествам $Q1$ и $Q2$, где $Q2$ получено с помощью модификации $Q1$ в каком-нибудь поколении. Иными словами, $Q1$ неоднократно было подвержено модификации, прежде чем было получено $Q2$. В результате в $Q1$ есть вершины, отсутствующие в $Q2$. Очевидно, их номера больше, чем x, в противном случае траектории из x по $Q1$ и $Q2$ не отличались бы друг от друга.

Лемма 2. Траектория из x по $Q1$ не короче траектории из x по $Q2$.

Доказательство. Пусть $Q1$ и $Q2$ отличаются единственным элементом — y. Он мог быть исключен из $Q1$, если через него была проложена какая-нибудь траектория. После исключения y при модификации $Q1$ точкой траектории при ветвлении будет выбрана какая-либо вершина y', $y < y'$. Очевидно, что траектория из y' по $Q2$ ничем не отличается от траектории из y' по $Q1$, т.к. после y множества $Q1$ и $Q2$ полностью совпадают. Следовательно, по лемме 1, траектория из y по $Q1$ не может быть короче траектории из y' по $Q2$.

Поскольку начала траекторий $(x,\ldots,y)$ и $(x,\ldots,y')$ отличаются единственной вершиной и содержат одинаковое число элементов, то траектория из x по $Q1$ не короче траектории из x по $Q2$. Значит, если $Q1$ и $Q2$ отличаются единственной точкой, лемма верна.

Предположим, она верна в том случае, когда $Q1$ и $Q2$ отличаются k элементами. Рассмотрим случай, когда эти множества отличаются $(k+1)$ элементами. Пусть y — последний элемент, исключенный при модификации. Сравним два множества: $Q1$ и $(Q2 \cup \{y\}) = Q'$. Они отличаются k элементами, следовательно, траектория из x по $Q1$ не короче траектории из x по Q'. В свою очередь, траектория из x по Q' не короче траектории из по $Q2$, т.к. эти множества различаются одним элементом. Отсюда следует, что при различии в $(k+1)$ элементов лемма также верна.

Теорема 1. Если граф (V,E) удовлетворяет допущению 1, то алгоритм БК1 строит на нем максимальное независимое множество наибольшей мощности.

Доказательство. Выберем $x = \mathrm{top}(V)$ в качестве начальной вершины и применим полный вариант алгоритма БК. Тогда, в силу леммы 2, траектория из x по первоначальному множеству $Q1 = V$ не будет короче, чем траектории из x по любой модификации $Q1$, в которых вершина x присутствует.

Пусть теперь модификация затронула вершину x. Тогда новая ветвь начнется из вершины z, $z > x$. Очевидно, соответствующая траектория не отличается от траектории из z по немодифицированному $Q1$. Следовательно, по лемме 1, эта траектория также не может быть длиннее самой первой траектории из x. Все последующие траектории из z пройдут уже по модифицированному варианту $Q1$, и, следовательно, по лемме 2 также не могут быть длиннее.

Таким образом, самой длинной является первая траектория из x, которая была построена алгоритмом БК1. Теорема доказана.

Рассмотрим случай двухкритериальной задачи оптимизации и пусть U — конечный набор паретовских точек в пространстве критериев. Перенумеруем точки множества U по возрастанию, например, первой координаты и рассмотрим три произвольные последовательные точки $x, y, z \in U$. Для них $x_1 \le y_1 \le z_1$, следовательно, $x_2 \le y_2 \le z_2$. Предположим, $\rho(x,z) \le H$, т.е. в соответствующем графе выполняется включение $(x,z) \in E$. Тогда несложно показать, что по крайней мере, для евклидовой метрики и l-метрики $\rho(x,z) = \max_i |x_i - z_i|$ выполняются неравенства $\rho(x,y) \le H$, $\rho(y,z) \le H$, а, значит, $(x,y) \in E$ и $(y,z) \in E$.

Следовательно, для данных метрик, построенный на множестве U граф удовлетворяет допущению 1, и алгоритм БК1 позволяет построить для него независимое множество максимальной мощности, т.е. получить точное решение задачи оптимальной фильтрации при любом количестве точек в исходном наборе.

2. ПРИБЛИЖЕННОЕ РЕШЕНИЕ ЗАДАЧИ ФИЛЬТРАЦИИ МНОЖЕСТВА ПАРЕТО

При большем числе критериев задача перенумерации вершин в соответствии с допущением 1, по-видимому, является сложной в вычислительном плане и поэтому от точного решения задачи оптимальной фильтрации следует отказаться. Это означает, что нужно искать эффективный приближенный метод решения. Поскольку мощность множества U, особенно при использовании алгоритмов второй группы, может быть достаточно большой, порядка нескольких тысяч, то вычислительная сложность метода фильтрации не должна быть выше $O(|U|^2)$.

В настоящее время известно много эффективных приближенных методов построения максимального независимого множества графа или решения близкой к ней задачи о минимальном покрытии [2, 4, 6—11]. Однако, большинство этих алгоритмов используют матрицу смежности размерности $|U|^2$, а, следовательно, возникает проблема ее хранения. Кроме того, алгоритм, использующий такую матрицу, может иметь сложность выше $O(|U|^2)$.

Несколько эвристических алгоритмов, специально посвященных решению задачи оптимальной фильтрации, приведены в [1]. Однако, численные эксперименты показали невысокую степень совпадения приближенного решения с точным. Причина, видимо, кроется в неудачной стратегии выбора исключаемых точек.

Одно из направлений разработки приближенных алгоритмов дает следующая лемма.

Лемма 3. В любом графе существует упорядоченность вершин, при которой алгоритм БК1 дает оптимальное решение задачи о независимом множестве наибольшей мощности.

Доказательство. Пусть V — множество вершин графа, $U \subseteq V$ — независимое множество наибольшей мощности, полученное с помощью алгоритма БК.

В процессе построения U некоторые вершины были исключены из V при обратном ходе алгоритма. Обозначим M — множество исключенных вершин. Тогда U получено из $V \setminus M$ с помощью БК1. Занумеруем вершины V следующим образом. На первые места поставим элементы $V \setminus M$ в соответствии с порядком их старых номеров, а вершины из M поставим в конец нумерации, присвоив произвольные номера.

Применим к полученному множеству алгоритм БК1. Так как в первую очередь будут рассматриваться элементы $V \setminus M$, то из ее вершин будет построено то же самое независимое множество U. Поскольку оно оптимально, то все элементы M, поставленные в конец списка, будут исключены при прямом ходе, и дополнительных элементов независимого множества не получится. Следовательно, данная нумерация позволит получить оптимальное решение с помощью БК1. Лемма доказана.

Вывод. При построении эффективного приближенного алгоритма фильтрации следует использовать стратегию исключения неподходящих вершин в соответствии с алгоритмом БК1, основное внимание уделить подбору оптимального или близкого к нему упорядочения исходного множества альтернатив.

Обозначим $U(x,h) = \{y \in V \mid y \neq x; \rho(x,y) \leq h\}$ — множество точек, удаленных от x не более, чем на h; Сорт(U) — операция сортировки множества U в соответствии с некоторым правилом. Алгоритм фильтрации, соответствующий данной выше рекомендации, будет иметь следующий вид.

```
Begin V[0] := V; m := 0;
    For k := 1 To p Do
    Begin V[k] := Сорт(V[k-1]); W := V[k];
        While W <> ∅ Do
        Begin
            x := top(W); x → S;
            W := W \ (U(x,h) ∪ {x});
        End;
        If |S| > m Then
        Begin m := |S|; S0 := S; End;
    End;
    Writeln(S0);
End.
```

Здесь p — число повторений сортировки, $p > 1$, если сортировки различны, например, случайны.

Модификация данного алгоритма может состоять в том, что во внутреннем цикле вместо алгоритма БК1 используется укороченный вариант полного алгоритма БК, т.е. вместо организации полного просмотра вариантов применяется просмотр с небольшим фиксированным числом чередований прямого и обратного ходов.

Три варианта предложенного алгоритма были подвергнуты тестовой проверке. Проверялись следующие методы:

1. Алгоритм многократной случайной сортировки. Использовался метод случайного перемешивания [12].

2. Однократная сортировка по возрастанию степени вершин. В соответствии с данной эвристикой в первую очередь просматривались вершины минимальной степени, т.е. имеющие минимальное число смежных вершин.

3. Сортировка по возрастанию некоторой линейной функции координат альтернатив в пространстве критериев. Эвристика данного способа упорядочения заключается в том, чтобы, начав с некоторой граничной точки множества V, постепенно продвигаться вдоль определенного направления, используя стратегию «коврового бомбометания».

Проверка алгоритмов проводилась как на множествах эффективных оценок тестовых задач непрерывной и дискретной оптимизации, так и на выборках точек многомерных выпуклых поверхностей, имитирующих множество Парето в пространстве критериев. По результатам проверки сделан вывод, что наилучшим из рассмотренных методов следует признать алгоритм 3.

СПИСОК ЛИТЕРАТУРЫ

1. *Штойер Р.М.* Многокритериальная оптимизация: теория, вычисления и приложения / Пер. с англ. М.: Радио и связь, 1992. 504 с.

2. *Гэри М.* Вычислительные машины и труднорешаемые задачи / Гэри М., Джонсон Д. М.: Мир, 1982. 416 с.

3. *Бугаев Ю.В.* Алгоритм бисекции в экстраполяции экспертных оценок // Экономика и математические методы, 2002, Т. 38, № 3. С. 121—125.

4. *Кристофидес Р.* Теория графов. Алгоритмический подход / Пер. с англ. М.: Мир, 1978. 432 с.

5. *Липский В.* Комбинаторика для программистов. М.: Мир, 1988. 213 с.

6. *Каплинский А.И.* Построение рандомизированных алгоритмов оптимизации / Каплинский А.И., Лимарев А.Е., Чернышова Г.Д. // Проблемы случайного поиска, 1980. Вып. 8. С. 63—91.

7. *Корбут А.А.* Об эффективности комбинаторных методов в дискретном программировании / А. А. Корбут, И. Х. Сигал, Ю. Ю. Финкельштейн // Современное состояние теории исследования операций / Под ред. Н. Н. Моисеева. М.: Наука, 1979. С. 283—310.

8. *Кузюрин Н.Н.* Задача линейного булева программирования и некоторые комбинаторные приемы // Компьютер и задачи выбора. М.: Наука, 1989. 208 с.

9. *Сергиенко И.В.* Приближенные методы решения дискретных задач оптимизации // И. В. Сергиенко, Т. Т. Лебедева, В. А. Рощин. Киев: Наукова думка, 1980. 273 с.

10. *Чернышева Г.Д.* Вероятностная модификация «жадных» алгоритмов комбинаторной оптимизации // Высокие технологии в технике и медицине. Воронеж: ВГТУ, 1994. С. 47—50.

11. *Шестова И.А.* Вероятностный алгоритм решения задачи о покрытии // Информац. технологии и системы. Научное издание, вып. 4. Воронеж: ВГТА, 2001. С. 187—191.

12. *Кнут Д.* Искусство программирования для ЭВМ. Ч. 2. Получисленные алгоритмы / Пер. с англ. М.: Мир, 1977. 724 с.