

## ОБОБЩЕНИЕ АЛГОРИТМА ФЛОЙДА–УОРШАЛЛА НА СЛУЧАЙ НЕСКОЛЬКИХ КРИТЕРИЕВ

**И.В. Блинов, Ю.В. Бугаев, С.В. Чикунов**

*Кафедра «Информационные технологии моделирования и управления»,  
ГОУ ВПО «Воронежская государственная технологическая академия»;  
mmtc@vgta.vrn.ru*

*Представлена членом редколлегии профессором В.И. Коноваловым*

**Ключевые слова и фразы:** бинарное отношение; вычислительная сложность; граф; функция выбора; эффективные пути.

**Аннотация:** Предложен многокритериальный алгоритм, являющийся обобщением на случай нескольких критериев известного метода Флойда–Уоршалла, позволяющий найти множества недоминируемых путей между всеми парами вершин ориентированного графа произвольного вида.

---

### Введение

Методы, основанные на принципе оптимальности Беллмана, являются эффективными процедурами решения задач оптимизации дискретных многостадийных процессов с аддитивным критерием оптимальности. В последнее время ведутся активные исследования по разработке алгоритмов решения подобных задач с векторным критерием: поиск лексикографических оптимумов, а также определение возможности применения алгоритма линейной свертки критериев (АЛСК).

Однако известно, что АЛСК не позволяет найти все множество нехудших решений [1]. Более надежны методы, использующие принцип прямого обобщения скалярного алгоритма на векторный случай. Так, разработан векторный алгоритм, позволяющий найти в ациклическом графе все пути от некоторой фиксированной вершины до всех остальных вершин, недоминируемые в смысле некоторого бинарного отношения предпочтения [2]. В данной работе предлагается новый, более общий метод решения подобной задачи – векторный вариант алгоритма Флойда–Уоршалла, позволяющий отыскать эффективные пути между всеми парами вершин в ориентированном графе произвольного вида.

### 1. Описание алгоритма

Пусть дан ориентированный граф  $G = (V, E)$ , где  $V = \{1, 2, \dots, n\}$  – множество вершин;  $E$  – множество дуг. Каждая дуга  $e \in E$  характеризуется набором весов  $\rho_u(e)$ ,  $u = \overline{1, m}$ , то есть снабжена векторным (многокритериальным) весом. Вес произвольного пути в графе равен сумме весов составляющих дуг, и на множестве весов дуг и путей задано некоторое бинарное отношение предпочтения  $R$ . Требуется найти между всеми парами вершин  $(i, j)$  все пути, недоминируемые в смысле отношения  $R$ .

Для решения этой задачи предлагается векторный вариант алгоритма Флойда–Уоршалла.

Вспомним скалярный вариант алгоритма [3, 4]. Положим вес дуги  $[i, j]$  равным элементу матрицы  $A[i, j]$ , который равен бесконечности, если в графе нет такой дуги и  $A[i, i] = 0$  для всех  $i = \overline{1, n}$ . Алгоритм базируется на использовании последовательности из  $n$  преобразований (итераций) начальной матрицы весов путей  $D[i, j]$  (первоначально  $D[i, j] = A[i, j]$  для всех  $i = \overline{1, n}, j = \overline{1, n}$ ). При этом на  $k$ -й итерации элементы матрицы представляют собой веса оптимальных путей между каждой парой вершин с тем ограничением, что путь между вершинами  $i$  и  $j$  содержит в качестве промежуточных только вершины из множества  $\{1, 2, \dots, k\}$ .

Для описания многокритериального варианта введем следующие обозначения:

–  $A[i, j]$  – векторный вес дуги  $[i, j]$ ; если в графе нет такой дуги, то все компоненты вектора равны бесконечности и  $A[i, i] = 0$  для всех  $i = \overline{1, n}$ ;

–  $\{D[i, j]\}$  – множество векторных весов возможных путей из вершины  $i$  в вершину  $j$ , полученное в результате работы алгоритма;

–  $C^R(X)$  – выбор на предъявлении  $X$  при использовании функции выбора (ФВ), определяемой механизмом блокировки по заданному бинарному отношению  $R$  [5].

**Определение.** Назовем  $R$ -оптимальным всякий путь в графе  $G$ , недоминируемый в смысле отношения  $R$ .

На первом этапе алгоритма производится поиск  $R$ -оптимальных векторных оценок  $\{D[i, j]\}$  для всех пар  $(i, j)$ .

Для простоты изложения алгоритм удобно описать в виде программы на неформальной версии языка Паскаль.

```
begin
1   for i:=1 to n do
2       for j:=1 to n do {D[i, j]}:=A[i, j];
3           for k:=1 to n do
4               for i:=1 to n do
5                   for j:=1 to n do
6                       for q ∈ {D[i, k]} do
7                           for p ∈ {D[k, j]} do
8                               {D[i, j]}:=CR[ {D[i, j]} ∪ {q+p} ];
end.
```

Алгоритм работает следующим образом.

Производится начальная установка, то есть в каждое множество  $\{D[i, j]\}$  записывается значение, равное весу дуги  $[i, j]$ .

Основная работа алгоритма начинается с 3-й строки, в которой задается последовательность из  $n$  преобразований (цикл по  $k = \overline{1, n}$ ) начальной матрицы весов путей  $\{D[i, j]\}$ .

На каждой  $k$ -й итерации в строке 4 последовательно перебираются все начальные вершины, а в строке 5 – все конечные вершины пути из вершины  $i$  в вершину  $j$ . В 6-й строке из множества  $\{D[i, k]\}$ , а в строке 7 из множества

$\{D[k, j]\}$  выбираются очередные  $R$ -оптимальные значения участков пути из вершины  $i$  в вершину  $j$  через вершину  $k$ . В строке 8 на основании функции выбора  $C^R(\bullet)$  проводится сравнение нового варианта пути, полученного в результате сложения между собой этих значений, с уже существующими вариантами. Процедура повторяется до тех пор, пока не будут рассмотрены все элементы множеств  $\{D[i, k]\}$  и  $\{D[k, j]\}$  (строки 6 и 7). Затем рассматриваются следующие пары вершин, и процедура выбора вновь повторяется.

После выполнения  $k$ -й итерации множества  $\{D[i, j]\}$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, n}$  будут содержать все  $R$ -оптимальные пути при условии, что в качестве промежуточных они будут содержать только вершины из множества  $\{1, 2, \dots, k\}$ .

После выполнения  $n$  итераций множества  $\{D[i, j]\}$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, n}$  будут содержать критериальные оценки всех  $R$ -оптимальных путей между всеми парами вершин графа  $G$ .

Второй этап алгоритма состоит в восстановлении самих эффективных путей. В некоторых других методах при построении пути к произвольной вершине  $i$  известна вершина  $u$ , последняя перед  $i$ , и ее можно запоминать. После определения всех оптимальных расстояний можно, начиная с конца, восстановить каждый из найденных оптимальных путей. В методе Флойда–Уоршалла последняя вершина перед данной вершиной  $i$  неизвестна. Это значит, что указанный метод восстановления путей в методе Флойда–Уоршалла не пригоден. Однако, зная матрицы  $D[i, j]$  наборов нехудших векторных расстояний между всеми парами вершин, можно построить сами кратчайшие пути между двумя заданными вершинами  $s$  и  $t$  с помощью алгоритма, скалярный вариант которого имеет вид [3]:

```

write(t);  v := t;
while v <> s do
  for u := 1 to n do
    if ( D[s, v] = D[s, u] + A[u, v] ) and ( u << v ) then
      begin write(u); v := u; break; end;

```

Векторный вариант более сложен, но использует ту же идею: обнаружение последней вершины  $u$  восстанавливаемого пути по совпадению  $D[s, v] = D[s, u] + A[u, v]$ . Алгоритм имеет вид:

```

for t ∈ V do
  for s ∈ V do
    for z ∈ {D[s, t]} do
      begin v := t; СТЕК := t; q := z;
        while v <> s do
          for u ∈ V \ {v} do
            for p ∈ {D[s, u]} do
              if q = p + A[u, v] then
                begin u => СТЕК; q := p; v := u;
                  break ( for u );
                end;
            write(СТЕК);
          end;

```

Вершины восстанавливаемого пути накапливаются в стеке, содержимое которого распечатывается, как только первая вершина пути совпадет с заданной начальной вершиной  $s$ .

## 2. Анализ алгоритма

Докажем корректность этого алгоритма. Обозначим  $\{d(i, j)\}^{(k)}$  – множество всех  $R$ -оптимальных путей из вершины  $i$  в вершину  $j$ , внутренние вершины которых содержатся в множестве  $\{1, 2, \dots, k\}$ . Сделаем следующие предположения.

1. В графе  $G$  каждая пара вершин может соединяться не более, чем одной дугой.

2. Отношение  $R$  транзитивно, асимметрично и не зависит от смещения. Последнее свойство означает, что для любой пары точек  $(x, y)$  и любого вектора  $b$  следующие отношения эквивалентны:  $xRy$  и  $(x + b)R(y + b)$ .

3. Векторный вес любого контура, имеющегося в графе, доминируется нулевым вектором.

**Теорема.** Если выполнены условия 1–3, то предлагаемый алгоритм позволяет найти все  $R$ -оптимальные пути между всеми парами вершин в графе  $G$ , и обратно, все пути, найденные алгоритмом, являются  $R$ -оптимальными, то есть для всех  $k = \overline{1, n}$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, n}$  выполняется равенство

$$\{D[i, j]\}^{(k)} = \{d(i, j)\}^{(k)}, \quad (1)$$

где  $\{D[i, j]\}^{(k)}$  – результат работы алгоритма на  $k$ -й итерации.

При доказательстве нам понадобятся следующие два утверждения.

**Лемма 1 [2].** Если  $X$  – конечное множество, а отношение предпочтения  $R$  транзитивно и асимметрично, то для любого  $x \in X \setminus C^R(X)$  существует элемент, доминирующий над ним в смысле отношения  $R$ , то есть

$$\forall x \in X \setminus C^R(X) \exists y \in C^R(X) : (y, x) \in R.$$

**Лемма 2.** Пусть ФВ  $C(\bullet)$ , определенная на конечном множестве  $T$ , обладает свойствами наследования и отбрасывания. Тогда для любых подмножеств  $X, Y \subseteq T$  справедливо равенство

$$C(X \cup Y) = C(X \cup C(Y)), \quad (2)$$

то есть последовательный выбор на промежуточных множествах совпадает с выбором на их объединении. В частности, это справедливо, если  $C(\bullet) = C^R(\bullet)$ , где  $R$  – асимметричное и транзитивное бинарное отношение.

Напомним, что свойства наследования (**H**) и отбрасывания (**O**) являются фундаментальными свойствами ФВ и формулируется следующим образом:

**H:** для произвольных предъятий  $X$  и  $Y$  выполнено следование

$$(Y \subseteq X) \Rightarrow (C(X) \cap Y \subseteq C(Y));$$

**O:** для любых множеств  $A, B$

$$(C(A) \subseteq B \subseteq A) \Rightarrow (C(A) = C(B)).$$

Этими свойствами обладает, например, ФВ  $C^R(\bullet)$ , где  $R$  – асимметричное и транзитивное бинарное отношение [5].

Известно также [6], что равенство (2) эквивалентно условию Плотта независимости от пути

$$C(X \cup Y) = C(C(X) \cup C(Y))$$

и имеет место только в том случае, когда ФВ обладает свойствами наследования и отбрасывания.

Доказательство теоремы. Предварительно отметим, что оператор в строке 8 алгоритма реализует операцию построения функции выбора после рассмотрения каждых  $p$  и  $q$ , а согласно лемме 2, это построение можно проводить после рассмотрения всех составляющих множеств. Поэтому строки 6, 7, 8 можно записать в виде:

$$\{D[i, j]\}^{(k)} = C^R \left( \{d(i, k)^{(k-1)} + d(k, j)^{(k-1)}\} \right). \quad (3)$$

Доказывать равенство (1) будем индукцией по  $k$ .

1. При  $k = 1$  справедливость равенства (1) очевидна, так как в силу ограничения множество  $\{D[i, j]\}^{(1)}$  может включать либо путь, состоящий из одной дуги  $[i, j]$ , либо путь из двух дуг  $[i, 1]$  и  $[1, j]$ , либо оба пути вместе, то есть

$$\{D[i, j]\}^{(1)} = C^R \left( \{D[i, j]\}^{(0)} \cup \{d(i, 1)^{(0)} + d(1, j)^{(0)}\} \right).$$

2. Предположим, что соотношение (1) выполняется для  $2, 3, \dots, k-1$  итераций, в частности имеем  $\{D[i, j]\}^{(k-1)} = \{d(i, j)\}^{(k-1)}$  для всех  $i = \overline{1, n}, j = \overline{1, n}$  и все внутренние вершины путей из  $i$  в  $j$  содержатся в множестве  $\{1, 2, \dots, k-1\}$ .

3. Докажем справедливость условия (1) для  $k$ -й итерации. Для этого, используя выражение (3), докажем, что

$$\{d(i, j)\}^{(k)} = C^R \left( \{d(i, k)^{(k-1)} + d(k, j)^{(k-1)}\} \right).$$

Пусть  $s$  – такой путь из вершины  $i$  в вершину  $j$ , что  $\rho(s) \in \{d(i, j)\}^{(k)}$ . Возможны два случая:

а) путь  $s$  не включает вершину  $k$ . Тогда множество  $\{D[i, j]\}^{(k)}$  остается неизменным и, согласно предположению индукции,

$$\{D[i, j]\}^{(k)} = \{D[i, j]\}^{(k-1)} = \{d(i, j)\}^{(k-1)};$$

б) путь  $s$  содержит вершину  $k$ . В этом случае обозначим через  $s'$  – часть пути  $s$  от вершины  $i$  до вершины  $k$ , а через  $s''$  – часть пути от вершины  $k$  до вершины  $j$ . Тогда для пути  $s' = s/s''$  должны получить  $\rho(s') \in \{d(i, k)\}^{(k-1)}$ .

Предположим, что это не так, то есть  $\rho(s') \notin \{d(i, k)\}^{(k-1)}$ . Тогда, по лемме 1, существует  $g'$  – путь от вершины  $i$  до вершины  $k$ , все внутренние вершины которого содержатся в  $\{1, 2, \dots, k-1\}$  и такой, что  $\rho(g') R \rho(s')$ . Вследствие того, что  $R$  не зависит от смещения, для пути  $g = g' \cup s''$  выполняется соотношение  $\rho(g) R \rho(s)$ , так как

$$\rho(s) = \rho(s') + \rho(s'') \quad \text{и} \quad \rho(g) = \rho(g') + \rho(s'').$$

Но такого  $g$ , лучшего по  $R$ , чем  $s$ , быть не должно. Отсюда следует, что

$$\rho(s') \in \{d(i, k)\}^{(k-1)},$$

а значит

$$\rho(s) \in \{d(i, k)^{(k-1)} + d(k, j)^{(k-1)}\} = H(i, j)^{(k)}.$$

Отсюда

$$\rho(s) \in C^R(H(i, j)^{(k)}),$$

иначе, по лемме 1, существует путь  $x$  с векторным весом лучшим, чем  $\rho(s)$ . А этого быть не может, так как  $\rho(s)$  принадлежит  $\{d(i, j)\}^{(k)}$  – множеству всех  $R$ -оптимальных путей из  $i$  в  $j$ , все внутренние вершины которых содержатся в множестве  $\{1, 2, \dots, k\}$ .

Итак, мы доказали включение

$$\{d(i, j)\}^{(k)} \subseteq C^R(H(i, j)^{(k)}). \quad (4)$$

Докажем обратное включение, то есть

$$C^R(H(i, j)^{(k)}) \subseteq \{d(i, j)\}^{(k)}.$$

Пусть  $\rho(s) \in C^R(H(i, j)^{(k)})$ . Предположим, что неверно включение  $\rho(s) \in \{d(i, j)\}^{(k)}$ , то есть  $\rho(s) \notin \{d(i, j)\}^{(k)}$ . Тогда, по лемме 1, найдется такой путь  $g$ , что  $\rho(g) \in \{d(i, j)\}^{(k)}$  и  $\rho(g) R \rho(s)$ . По доказанному включению (4), если  $\rho(g) \in \{d(i, j)\}^{(k)}$ , то  $\rho(g) \in C^R(H(i, j)^{(k)})$ . А это означает, что не может выполняться отношение  $\rho(g) R \rho(s)$ , так как оба пути недоминируемы по бинарному отношению  $R$ . Следовательно

$$\rho(s) \in \{d(i, j)\}^{(k)},$$

а значит

$$C^R(H(i, j)^{(k)}) \subseteq \{d(i, j)\}^{(k)}.$$

Отсюда вытекает равенство этих двух множеств, то есть

$$\{D[i, j]\}^{(k)} = \{d(i, j)\}^{(k)}.$$

4. Воспользовавшись предположением индукции, получим выполнение данного соотношения для всех  $k = 1, 2, \dots, n$ . Следовательно, корректность алгоритма доказана, и при  $k = n$  будут найдены все  $R$ -оптимальные пути между всеми парами вершин в ориентированном графе, так как в этом случае все внутренние вершины всех путей будут содержаться в множестве  $\{1, 2, \dots, n\}$ .

Оценим теперь вычислительную сложность данного алгоритма. Очевидно, что число машинных операций, необходимых для выполнения первого этапа алгоритма значительно больше, чем для восстановления путей, поэтому вычислительная сложность всего алгоритма совпадает со сложностью его первой части.

Пусть  $n$  – число вершин графа;  $L = \max_{i, j} |d(i, j)|$  – максимальное количество

$R$ -оптимальных путей среди всех пар вершин  $i$  и  $j$ ;  $m$  – число критериев.

Число повторений каждого из трех внешних циклов (строки 3, 4 и 5 алгоритма) имеет порядок  $O(n)$  или, трех циклов в целом –  $O(n^3)$ , а каждого из двух внутренних циклов (строки 6 и 7) –  $O(L)$  или, в общем,  $O(L^2)$ . В цикле 7 при каж-

дом повторении необходимо не более  $L$  раз провести парное сравнение  $m$ -мерных векторов. Следовательно, вычислительная сложность всего цикла  $\delta$  составит  $O(mL^3)$ . Поскольку в большинстве практических задач значение  $m$  ограничено некоторой константой, то можно принять вычислительную сложность цикла  $\delta$  равной  $O(L^3)$ .

Следовательно, общая вычислительная сложность многокритериального алгоритма Флойда–Уоршалла составляет  $O(L^3 n^3)$ , против  $O(n^3)$  в случае одного критерия.

Дать точную оценку величине  $L$  для произвольного графа весьма сложно, так как заранее найти мощность множеств эффективных путей практически невозможно. Однако численные эксперименты, проведенные авторами и описанные в [7, 8], а также теоретически полученные асимптотические оценки [9], позволяют надеяться, что, в частности, для отношения Парето величина  $L$  растет достаточно медленно по сравнению с числом вершин  $n$  произвольного графа, если он не близок к полному графу. Поэтому данный алгоритм может быть рекомендован для решения практически важных задач.

*Работа выполнена при финансовой поддержке Рособразования, Государственный контракт № П947.*

#### *Список литературы*

1. Кравцов, М.К. Неразрешимость задач векторной дискретной оптимизации в классе алгоритмов линейной свертки критериев / М.К. Кравцов // Дискрет. математика. – 1996. – Т. 8, вып. 2. – С. 89–96.
2. Бугаев, Ю.В. Применение прямого обобщения скалярных алгоритмов в векторной оптимизации на графах / Ю.В. Бугаев // Дискрет. математика. – 2001. – Т. 13, вып. 3. – С. 110–124.
3. Липский, В. Комбинаторика для программистов : пер. с пол. / В. Липский / М. : Мир, 1988. – 213 с.
4. Кристофидес, Р. Теория графов. Алгоритмический подход : пер. с англ. / Р. Кристофидес. – М. : Мир, 1978. – 432 с.
5. Юдин, Д.Б. Вычислительные методы теории принятия решений / Д.Б. Юдин. – М. : Наука, 1989. – 316 с.
6. Айзерман, М.А. Выбор вариантов: основы теории / М.А. Айзерман, Ф.Т. Алескеров. – М. : Наука, 1990. – 240 с.
7. Чикунов, С.В. Структурно-параметрический синтез моделей многокритериального поэтапного выбора решений в технологических системах : дис. ... канд. техн. наук : защищена 16.10.03 : утв. 16.01.04 / Чикунов Сергей Владимирович. – Воронеж, 2003. – 171 с.
8. Бугаев, Ю.В. Экспериментальные исследования алгоритмов многокритериального поэтапного выбора решений в технологических системах / Ю.В. Бугаев, С.В. Чикунов // Теоретические основы проектирования технологических систем и оборудования автоматизированных производств : сб. науч. тр. / Воронеж. гос. технол. акад. – Воронеж, 2007. – Вып. 5, ч. 1. – С. 68–82.
9. Бугаев, Ю.В. Обобщение схемы динамического программирования / Ю.В. Бугаев, С.В. Чикунов // Автоматика и телемеханика. – 2009. – № 2. – С. 90–100.

## Generalization of Floyd–Warshall Algorithm in Case of Several Criteria

I.V. Blinov, Yu.V. Bugaev, S.V. Chikunov

*Department «Information Technologies and Management Modeling»,  
Voronezh State Technological Academy;  
mmtc@vgta.vrn.ru*

**Key words and phrases:** binary relation; computing complexity; graph; choice function; effective ways.

**Abstract:** The paper proposes multi-criteria generalized algorithm in case of several criteria of the well-known Floyd–Warshall method enabling to find a number of non-dominant ways between all the top steams of oriented graph of random type.

---

## Verallgemeinerung des Algorithmus von Floyd–Warshall im Falle einiger Kriterien

**Zusammenfassung:** Es wird der multicriteriale Algorithmus, der die Verallgemeinerung im Falle einiger Kriterien der bekannten Methode von Floyd–Warshall ist, vorgeschlagen. Er erlaubt, die Mengen der nicht dominierten Wege zwischen allen Paaren Gipfel des ausgerichteten Grafen einer willkürlichen Art zu finden.

---

## La généralisation de l'algorithme de Floyd–Warshall Sur le cas de quelques critères

**Résumé:** Est proposé l'algorithme multicritériel étant la généralisation sur le cas de quelques critères de la méthode connue de Floyd–Warshall permettant de trouver les multitudes de voies non dominées entre toutes les vapeurs des sommets du comte orienté de l'aspect arbitraire.

---

**Авторы:** *Блинов Иван Владимирович* – аспирант кафедры «Информационные технологии моделирования и управления»; *Бугаев Юрий Владимирович* – доктор физико-математических наук, профессор кафедры «Информационные технологии моделирования и управления»; *Чикунов Сергей Владимирович* – кандидат технических наук, доцент кафедры «Информационные технологии моделирования и управления», ГОУ ВПО «ВГТА».

**Рецензент:** *Абрамов Геннадий Владимирович* – доктор технических наук, профессор, заведующий кафедрой «Информационные технологии моделирования и управления», ГОУ ВПО «ВГТА».