

УДК 004.02+519.1

## **Систематизация точных методов дискретной оптимизации**

**Овчинников В. А.<sup>1,\*</sup>**

[\\*yaovchinnikov@gmail.com](mailto:yaovchinnikov@gmail.com)

<sup>1</sup>МГТУ им. Н.Э. Баумана, Москва, Россия

---

На основе анализа учебно-методической литературы и научных публикаций, в которых описываются и используются точные методы комбинаторной оптимизации, выполнена систематизация указанных методов. Описаны особенности и сущность классических методов, таких как жадный выбор, поиск в ширину и в глубину с возвращением, ветвей и границ, Дейкстры, Форда-Фалкерсона и динамического программирования, даны характеристики трудоемкости их реализации. Сформулированы признаки задач, позволяющие применить для их решения соответствующий метод. Приведена таблица практических задач, в которой указаны методы их решения и вычислительная сложность алгоритмов.

**Ключевые слова:** дискретная оптимизация, точные методы, структурный синтез, декомпозиция, алгоритм, верхняя и нижняя границы, схема метода, вычислительная сложность

---

### **Введение**

Задачи дискретной оптимизации решаются при разработке практически любых сложных объектов и систем. В связи с этим, освоение методов решения комбинаторно-оптимизационных задач структурного синтеза становится одним из основных компонентов подготовки специалистов. В то же время литературе по этим методам присущ ряд существенных недостатков, затрудняющих как преподавание, так и освоение этих методов, а также использование их при решении новых задач.

Цель работы – систематизировать описанные в различных источниках комбинаторные методы дискретной оптимизации, охарактеризовать их возможности и указать свойства задач, для решения которых могут быть использованы соответствующие методы.

Анализ учебно-методической литературы [2– 4, 7–12], в которой описываются точные методы комбинаторной оптимизации, монографий и научных статей [1, 5, 6, 13– 26], связанных с их использованием, позволяет сделать следующие выводы:

– непрерывно расширяется круг применения классических методов дискретной оптимизации при возрастании размерности проектируемых систем;

– во многих источниках не излагается сущностная схема метода, а приводится его иллюстрация на алгоритме решения конкретной задачи;

– отсутствует систематизация точных методов решения комбинаторно-оптимизационных задач;

– как правило, не сформулированы признаки задач, позволяющие применить для их решения соответствующий метод;

– нередко авторы не делают различия между терминами метод и алгоритм. Примером может служить «жадный алгоритм» (метод), на основе которого разработано значительное число алгоритмов решения различных задач из разных предметных областей;

– оценки вычислительной сложности алгоритмов, реализующих точные методы решения комбинаторно-оптимизационных задач, разбросаны по различным, нередко трудно доступным, источникам;

– в ряде случаев используемая терминология представляется не совсем удачной. Например, методы поиска в глубину и в ширину не предназначены для поиска оптимального решения. Их следовало бы отнести к методам декомпозиции пространства решений. В то же время методы поиска оптимального решения, основанные на указанных стратегиях декомпозиции, в литературе практически не описаны.

Все это в целом затрудняет специалистам выбор метода решения новых прикладных задач структурного синтеза и обучение курсам дискретной оптимизации.

## **1. Стратегии поиска решения комбинаторно-оптимизационных задач**

Можно выделить две основные стратегии поиска решения задач дискретной оптимизации. Первая из них основана на двух идеях [7, 8, 10]:

– *декомпозиция* пространства решений;

– *исследование* множества возможных решений, т. е. поиск в процессе декомпозиции оптимального решения на основе некоторой оценки его качества.

Вторая стратегия реализует следующие три идеи [7]:

– *разбиение* задачи на подзадачи;

– *оценка* значения локального критерия качества решения подзадач и *определение* оптимального варианта подзадач;

– *получение* решения композицией (объединением) этих подзадач.

Декомпозицию пространства решений используют такие базовые методы решения комбинаторно оптимизационных задач как жадный выбор, поиск в ширину и в глубину с возвращением, метод ветвей и границ, а также методы Дейкстры и Форда-Фалкерсона. Вторую стратегию реализует метод динамического программирования.

## 2. Методы декомпозиции пространства решений

В процессе декомпозиции множество  $M$  возможных вариантов решения задачи в соответствии с принципом формирования результата разбивается на подмножества  $M_i$  такие, что  $\cup M_i = M$  [8, 10]. Далее, используя тот же принцип, полученные подмножества вновь разбиваются на подмножества, включающие в себя меньшее число вариантов. После некоторого шага разбиения каждое множество содержит по одному варианту решения.

Сопоставив каждому подмножеству вершину графа и соединив ребром вершины, соответствующие подмножествам  $M_i$  и  $M_j$ , если подмножество  $M_j$  получено непосредственным разбиением подмножества  $M_i$ , получим *дерево решений или дерево декомпозиции*.

Существует две стратегии декомпозиции, отличающиеся порядком получения вершин дерева решений [2 – 5, 7, 9, 10]:

- в ширину;
- в глубину с возвращением.

*Декомпозиция в ширину.* Все множество возможных вариантов решения  $M$  в соответствии с принципом формирования результата разбивается на подмножества первого уровня  $M_i^1$  такие, что

$$\cup M_i^1 = M.$$

Затем каждое подмножество первого уровня  $M_i^1$  разбивается на подмножества второго уровня  $M_j^2$  такие, что

$$\cup M_j^2 = M_i^1.$$

Процесс продолжается до тех пор, пока каждое подмножество не будет соответствовать одному варианту решения.

*Декомпозиция в глубину с возвращением.* Все множество возможных вариантов решения  $M$  разбивается на подмножества  $M_1$  и  $M \setminus M_1$ . Затем подмножество  $M_1$  разбивается на подмножества  $M_2$  и  $M_1 \setminus M_2$ . Продвигаемся по этой ветви пока не получим подмножество, соответствующее одному варианту решения. Возвращаемся по ветви вверх, пока не придем в вершину, из множества вариантов которой можно выделить некоторое подмножество в соответствии с принятым принципом формирования решений. Начиная с этого подмножества, строим следующую ветвь до получения подмножества, соответствующего одному варианту решения. Процесс повторяется, пока не будут получены все варианты.

Из описания процесса декомпозиции пространства решений методами в ширину и в глубину с возвращением видно, что решения (не всегда одни и те же) получаются последовательно, но в разном порядке. Таким образом, рассмотренные методы

декомпозиции могут использоваться для решения задач, в которых необходимо найти все варианты.

Методы декомпозиции в ширину и в глубину с возвращением используются для решения многих задач дискретной математики. Укажем некоторые из них.

*Построение глубинного или  $d$ -дерева.* В неориентированном  $G(X, U)$  или ориентированном  $G^{\rightarrow}(X, U)$  графе методом декомпозиции в глубину с возвращением строится дерево *просмотра* вершин графа. Здесь  $X$  – множество вершин и  $U$  – множество ребер графа. Вершины и ребра графа включаются в дерево (просматриваются) один раз. Вид глубинного дерева зависит от начальной вершины и порядка перечисления смежных вершин. Для одной компоненты связности в виде неориентированного графа  $G(X, U)$  глубинное дерево является остовным.

Глубинное дерево может строиться и в ориентированном графе. Построение этого дерева вводит на множестве вершин ориентированного графа отношение частичного порядка:  $x_i < x_j$ , если  $x_i$  является предком вершины  $x_j$ .

Отношение частичного порядка на множестве вершин ориентированного графа позволяет на основе построения глубинного остовного дерева решать, например, такие задачи как:

- распознавание сильной связности ориентированного графа, т. е. существования в нем пути из каждой вершины в любую другую [2];
- отыскание блоков, мостов и расщепляющих вершин [2, 7];
- установления ацикличности ориентированного графа [7];
- отыскание всех гамильтоновых циклов графа (2).

*Построение дерева в ширину или  $b$ -дерева.* Это дерево строится в соответствии со стратегией декомпозиции в ширину. Вершины и ребра графа включаются в дерево (просматриваются) один раз. Дерево может быть построено как для неориентированного, так и для ориентированного графа. Вид дерева зависит от назначения вершины-корня и порядка перечисления вершин в образах вершин-предков. Дерево содержит все достижимые из корня вершины. Для каждой вершины путь до нее из корня является одним из кратчайших, в смысле числа включенных в него ребер, путей в графе. Для неориентированного графа  $b$ -дерево – остовное.

Просмотр графа в ширину составляет основу алгоритмов решения, например таких задач как:

определения длины кратчайшего пути (в указанном выше смысле) от некоторой вершины до каждой из достижимых [7];

построение остовного дерева минимального веса [7] и др.

Просмотр графа как в глубину с возвращением, так и ширину требует  $(n + m)$  операций, где  $n = |X|$ ,  $m = |U|$ . Если локальная степень вершин неориентированного графа (или сумма полустепеней исхода и захода в ориентированном) ограничена константой, то асимптотическая оценка построения  $d$ - и  $b$ -деревьев будет равна  $O(n)$ .

### 3 Отсечение или выбор вариантов

В практике проектирования задачи нахождения всех возможных решений довольно редки. Обычно необходимо найти одно решение, наилучшее в смысле некоторого функционала (целевой функции). Поиск такого решения во всем множестве вариантов, полученных по методу в ширину или в глубину с возвратом, подразумевает вычисление значений целевой функции для каждого варианта, сравнение их и выбор оптимального, т.е. приводит к полному перебору. Полный перебор требует слишком много времени либо просто нереализуем при достаточно большой размерности входа задачи.

Существование некоторой оценки, которая с той или иной степенью достоверности позволяет судить о том, содержит ли подмножество вариантов оптимальное решение, обеспечило бы возможность избежать полного перебора [7, 8, 10]. В общем случае оценка – это значение функции  $F(M_i)$  на вершинах дерева решений, как правило, исключая корень, равная в его конечных вершинах значению целевой функции для соответствующего варианта решения, а в остальных – нижней  $F_n$  или верхней  $F_v$  границе целевой функции для вариантов, входящих в это множество.

Нижняя и верхняя границы означают, что значение целевой функции для всех вариантов, порожаемых множеством  $M_i$ , не меньше и не больше числа  $F(M_i)$  соответственно. Вид дерева решений зависит от оценочной функции.

При полной достоверности оценки того, что подмножество вариантов *не содержит оптимального решения*, данное подмножество можно исключить из процесса разбиения. В противном случае такая оценка может служить для обоснования очередности разбиения подмножеств, т.е. выбора на каждом шаге построения дерева решений наиболее «перспективной» вершины, которая с большей вероятностью, чем другие, может содержать оптимальное решение. В дальнейшем будем на этом основании различать *отсекающую оценку* и *оценку перспективности*.

Отсекающей является также оценка выбора подмножества, гарантированно содержащего оптимальный вариант решения. В этом случае отсекаются все или некоторые подмножества вариантов, которые порождаются другими вершинами дерева решений. Отметим, что все точные комбинаторные методы решения задач дискретной оптимизации используют идею отсечения.

Для оценки перспективности лучшей является такая функция, при которой разность между оценками подмножеств наибольшая, а сами оценки вычисляются с наибольшей точностью в смысле их близости к значению целевой функции для оптимального варианта подмножества.

## 4 Методы, использующие декомпозицию пространства решений

### 4.1 Метод жадного выбора

Этот метод осуществляет поиск решения задачи в процессе декомпозиции по методу в ширину. Он заключается в том, что *на каждом уровне* дерева решений в соответствии с некоторой оценкой, определяемой спецификой задачи, *выбирается одно подмножество, отсекая все остальные*, до тех пор, пока не получим решение. Таким образом, здесь формируется только одна ветвь дерева решений. Отметим, что для этого метода часто используют термин «жадный алгоритм» [3, 7, 9].

Свойства задач, определяющие возможность получения точного решения данным методом:

- задача может быть разбита на подзадачи, имеющие альтернативные варианты решения;
- для подзадачи уровня справедлив принцип оптимальности, т. е. для вершин одного уровня дерева декомпозиции существует отсекающая оценка;
- подзадачи можно сформировать только после выбора подзадачи предыдущего уровня.

*Метод обеспечивает получение точного решения за полиномиальное время.* Этот метод лежит в основе, например, алгоритма Прима определения минимального остовного дерева. Асимптотическая оценка вычислительной сложности этого алгоритма не хуже  $O(n^2)$  [5].

### 4.2 Поиск в ширину и в глубину с возвращением

Последовательное получение решений при декомпозиции как в ширину, так и в глубину с возвращением, позволяет реализовать поиск оптимального решения в процессе декомпозиции и, возможно, избежать полного перебора для *NP*-полных задач [9, 10].

Метод целесообразно применять, если:

- задача может быть разбита на подзадачи, которые можно сформировать только после выбора подзадачи предыдущего уровня;
- отсекающей оценкой является опорное решение, т.е. значение целевой функции уже полученного варианта решения, или условие, связанное со структурой либо свойствами решения.

В ходе решения, если возможно, уточняется значение отсекающей оценки. Оптимальное решение будет найдено, когда в задаче на минимум целевой функции ее значение для некоторой конечной вершины меньше нижней границы  $F_n$  для всех висячих вершин и меньше значения целевой функции для всех остальных конечных вершин (в задаче на поиск максимума целевой функции соответственно «больше»).

Нижняя  $F_n$  или верхняя  $F_v$  границы наиболее просто вычисляются в тех задачах, в которых необходимо найти оптимальное решение по минимуму или максимуму суммы весов ребер. Примером такой задачи является симметричная или несимметричная задача

коммивояжера (поиск гамильтонова цикла) [2– 4, 7]. Как известно, для ее решения не существует алгоритма с полиномиальной оценкой вычислительной сложности. В связи с этим отсечение вариантов на основе использования опорного решения при поиске в ширину или глубину с возвращением может оказаться весьма полезным.

Степень сокращения числа перебираемых вариантов при применении данных методов непредсказуема – решение может быть найдено и за полиномиальное время и в результате полного перебора. *Важным является то, что оба эти метода могут применяться для точного решения NP полных задач.*

Невозможно дать рекомендации по использованию в тех или иных случаях методов поиска в ширину и в глубину с возвращением и порядка просмотра вершин. Обычно при ветвлении по методу поиска в ширину исследуется меньшее число вершин дерева решений, но хранить приходится данные о большем числе построенных вершин, чем при ветвлении по методу поиска в глубину с возвращением. Однако, здесь многое зависит от конкретной задачи и значений исходных данных.

#### 4.3 Метод ветвей и границ

Метод ветвей и границ является универсальным, хотя и достаточно сложным, методом точного решения широкого круга NP-полных комбинаторно-оптимизационных задач. Метод реализует *направленный* перебор вариантов, используя разумную эвристику – выполнять ветвление вершин не в заданной последовательности, а более перспективных [2, 5, 8, 10, 12]. Вершины выбираются на основе оценки, которая с той или иной степенью достоверности позволяет сделать заключение, что подмножество вариантов, сопоставленных данной вершине дерева декомпозиции, может содержать оптимальное решение. Метод наиболее эффективен, если оценка монотонна.

В худшем случае метод может вылиться в полный перебор. В общем случае выполняется частичный перебор. Сокращение числа просмотренных вариантов достигается за счет:

- *организации ветвления* – разбиения множества вариантов на подмножества в наиболее перспективной вершине;
- *отсечения* подмножеств вариантов, не содержащих оптимального.

Декомпозиция, т.е. разбиение множества вариантов на подмножества в исходной вершине и в выбранных перспективных вершинах выполняется *по методу в ширину*. Вычисление условий отсечения существенно зависит от вида задачи, а степень сокращения перебора – от ее конкретных данных.

*Схема метода.* Организация ветвления подразумевает выбор в процессе разбиения наиболее перспективной вершины, т.е. в разбиении того подмножества вариантов, к которому вероятнее всего принадлежит оптимальный. В качестве оценки перспективности ветви используется верхняя или нижняя границы целевой функции, вычисляемые для каждого подмножества вариантов.

Рассмотрим процесс ветвления. Сначала выполняется разбиение всего множества вариантов, т.е. строятся на следующем от корня уровне все или часть вершин его потомков. Затем на каждом шаге выбирается вершина ветвления по минимуму нижней границы (максимуму верхней для задачи на максимум целевой функции) и разбивается соответствующее ей подмножество вариантов. В ходе ветвления используется, если это возможно, тот или иной способ отсечения ветвей и вершин, например по результатам сравнения нижней или верхней границы целевой функции с опорным решением.

Оптимальное решение будет найдено при выполнении тех же условий, которые были указаны для поиска в ширину и в глубину с возвращением. Для получения опорного решения возможна комбинация ветвления в ширину и метода жадного выбора.

Рассматриваемый метод по сравнению с другими методами точного решения *NP* полных задач (поиск в ширину и глубину с возвращением) в большей степени направлен на сокращение полного перебора, однако более трудоемок.

#### **4.4 Метод Дейкстры**

В настоящее время в специальной литературе [2, 3, 5, 7, 10] с разной степенью детализации описан алгоритм Дейкстры поиска маршрута минимальной длины и большинством авторов как метод не трактуется. Поскольку идеи этого алгоритма могут быть использованы для решения довольно широкого круга задач дискретной оптимизации, его общую схему целесообразно рассматривать как метод.

Метод Дейкстры использует схему метода ветвей и границ и позволяет найти за полиномиальное время точное решение таких задач, для которых характерны следующие особенности:

- решение задачи можно разбить на подзадачи (в задаче определения кратчайшего пути подзадачами являются части пути, приходящие в вершину);
- существуют подзадачи как имеющие, так и не имеющие альтернативной реализации;
- для подзадач с альтернативной реализацией есть отсекающая оценка, однако она не распространяется на другие подзадачи;
- подзадачи не могут быть сформированы заранее, так как зависят от структуры исследуемого графа и порождаются в ходе реализации метода выбранным вариантом решения подзадачи предыдущего уровня;
- число вариантов решения задачи экспоненциально зависит от размера входа задачи, а число подзадач, среди которых выполняется выбор перспективной, – полиномиально.

Декомпозиция множества решений при получении вершин первого уровня графа решений происходит по методу в ширину, далее ветвление выполняется в перспективной вершине.

Типичной задачей, обладающей указанными выше свойствами является поиск маршрута (простой цепи) минимальной длины из некоторой исходной точки в заданную

конечную. Метод может быть использован для определения дополняющего пути максимальной пропускной способности в задаче о максимальном потоке в сети, в задаче поиска максимального паросочетания в двудольном графе и др. [2, 7].

#### 4.5 Метод Форда-Фалкерсона

Этот метод является обобщением широко известного алгоритма тех же авторов. Метод является основой алгоритмов решения задач о максимальном потоке в сети, о максимальном паросочетании в двудольном графе и др. [2, 7]. Сетью является, например, система, состоящая из некоторого множества объектов, связанных каналами передачи сообщений. Один из этих объектов является только источником сообщений ( $s$ ), другой – только приемником ( $t$ ). Остальные объекты выполняют прием и дальнейшую передачу сообщений без потерь. Каналы передачи сообщений имеют ограниченную целочисленную пропускную способность  $c$ . В процессе итерационного поиска решения состояние каждого канала характеризуется парой величин – значение передаваемого потока  $f$  и пропускная способность  $c$ .

Метод Форда-Фалкерсона гарантирует нахождение максимального потока только в сетях с целочисленными и рациональными пропускными способностями (последние могут быть приведены к целочисленным). Для случая иррациональных пропускных способностей итерационный процесс поиска решения может не сходиться.

Основой метода является определение дополняющего пути в остаточной сети. Остаточная сеть для некоторого состояния основной сети строится следующим образом:

- если пропускная способность канала больше передаваемого потока, то она уменьшается на значение потока и добавляется обратный канал с пропускной способностью, равной передаваемому потоку;
- если пропускная способность канала равна передаваемому потоку, то данный канал удаляется и добавляется обратный канал с пропускной способностью, равной передаваемому потоку.

Дополняющий путь – это простая цепь из истока  $s$  в сток  $t$  в остаточной сети. В классическом изложении метода [7] способ определения дополняющего пути не декларируется.

Метод в целом заключается в следующем:

- 1) Задается начальное состояние сети: значение передаваемого потока  $f$  для всех каналов устанавливается равным нулю. Полученная сеть является остаточной.
- 2) Методом поиска в глубину с возвращением или в ширину в остаточной сети ищется первый по порядку построения дополняющий путь, пропускная способность которого  $\Delta f$  равна минимуму пропускных способностей составляющих его каналов.
- 3) Увеличивается на  $\Delta f$  суммарный поток сети  $F$ . Значения передаваемого потока  $f$  каналов сети изменяется в соответствии с правилом:
  - если канал дополняющего пути является прямым, т.е. каналом основной сети, то поток через него увеличивается на  $\Delta f$ ;

– если канал дополняющего пути является обратным, то поток в сети по соответствующему прямому каналу уменьшается на  $\Delta f$ .

4) Определяется остаточная сеть.

Процесс повторяется с п. 2 до тех пор, пока существует дополняющий путь.

Вычислительная сложность алгоритмов, построенных на основе этого метода, не хуже  $O(n m^2)$  [7].

## 5 Метод динамического программирования

Динамическое программирование применим для решения задач, для которых характерны следующие свойства [3, 7]:

– задачу можно разбить на независимые подзадачи, эти подзадачи на более мелкие и так далее;

– вид подзадач можно определить заранее, анализируя структуру задачи;

– оценка качества решения подзадачи является локальным критерием оптимальности;

– для задачи в целом справедлив принцип оптимальности;

– число подзадач полиномиально зависит от размера входа задачи, а число вариантов их объединения – экспоненциально.

Особенности (схема) метода:

– решения получаются объединением решений предыдущих шагов, *которые в общем случае входят в разные варианты допустимых*;

– объединяемые подзадачи (решения предыдущих шагов) выбираются в соответствии с локальным критерием оптимальности;

– оценки рассчитываются для всех допустимых решений (подзадач) *один раз по полученным ранее оценкам решения объединяемых подзадач* и используются в процессе решения. Для запоминания оценок качества решения подзадач обычно используют специальные таблицы. Способ их организации определяется правилом вычисления оценок последующих решений по оценкам предыдущих, их составляющих.

*Метод обеспечивает получение* точного решения за полиномиальное время.

Вычислительная сложность алгоритмов, реализующих метод динамического программирования, не хуже  $O(n^2)$ . Метод динамического программирования используется при решении задачи оптимальной триангуляции выпуклого многоугольника, наибольшей общей подпоследовательности и др. [7].

## 6 Примеры задач дискретной оптимизации и вычислительная сложность алгоритмов, реализующих точные методы их решения

В таблице 1 приведены некоторые задачи дискретной оптимизации, соответствующие примеры практических задач и вычислительная сложность некоторых описанных в литературе алгоритмов их решения. Вычислительная сложность алгоритмов,

построенных на основе методов с полиномиальной оценкой трудоемкости, в зависимости от используемых оптимизирующих преобразований может существенно отличаться от указанной выше.

Таблица 1.

| № | Задача дискретной оптимизации                             | Пример прикладной задачи                                      | Модели объектов                                                       | Класс задач | Метод точного решения              | Вычислительная сложность [источник]                                                                     |
|---|-----------------------------------------------------------|---------------------------------------------------------------|-----------------------------------------------------------------------|-------------|------------------------------------|---------------------------------------------------------------------------------------------------------|
| 1 | 2                                                         | 3                                                             | 4                                                                     | 5           | 6                                  | 7                                                                                                       |
| 1 | Позиционирование                                          | Размещение компонентов в монтажной области                    | Структуры системы – гиперграф, монтажного пространства – граф решетки | <i>NP</i>   | Ветвей и границ                    | Экспоненциальная                                                                                        |
| 2 | Определение минимального остовного дерева                 | Установление порядка соединения попарно связанных объектов    | Взвешенный неориентированный граф                                     | <i>P</i>    | Жадного выбора                     | $O(n^2) - [5]$ ,<br>$O(m \log \log n) - [1]$                                                            |
| 3 | Поиск простой цепи с минимальной суммой весов ребер       | Определение маршрута минимальной длины                        | Взвешенный неориентированный (ориентированный) граф                   | <i>P</i>    | Дейкстры                           | Для разреженных графов:<br>$O(m \log n) - [7]$ ,<br>$O(m \log \log D) - [1]$ , $D$ – максимальная длина |
| 4 | Поиск гамильтонова цикла с минимальной суммой весов ребер | Определение замкнутого маршрута минимальной длины             | Взвешенный неориентированный (ориентированный) граф                   | <i>NP</i>   | Ветвей и границ                    | Экспоненциальная                                                                                        |
| 5 | Определение наибольшего паросочетания                     | Распределение преподавателей по учебным курсам                | Двудольный неориентированный (ориентированный) граф                   | <i>P</i>    | Форда - Фалкерсона                 | $O(n \cdot m) - [7]$ ,<br>$O(n^{1/2} \cdot m) - [1]$ ,                                                  |
| 6 | Построение глубинного дерева с одинаковыми весами ребер   | Выделение из системы подсистемы с заданным приоритетом связей | Взвешенный ориентированный граф                                       | <i>P</i>    | Декомпозиция в глубину с возвратом | $O(n)$ или $O(m) - [7, 11]$                                                                             |
| 7 | Разрезание гиперграфа                                     | Декомпозиция структуры сложной системы                        | Гиперграф                                                             | <i>NP</i>   | Ветвей и границ                    | Экспоненциальная                                                                                        |

|    |                                                      |                                                                                              |                                            |      |                                                                        |                                                                           |
|----|------------------------------------------------------|----------------------------------------------------------------------------------------------|--------------------------------------------|------|------------------------------------------------------------------------|---------------------------------------------------------------------------|
| 8  | Поиск дополняющего пути в сети                       | Выявление каналов увеличения потока информации в системе                                     | Взвешенный ориентированный граф            | $P$  | Поиск в ширину                                                         | $O(n^2) - [7]$                                                            |
| 9  | Установление изоморфизма графов                      | Определение идентичности структур систем                                                     | Взвешенные ультра-или ориентированный граф | $NP$ | Модификация метода ветвей и границ                                     | Экспоненциальная                                                          |
| 10 | Нахождение наибольшего независимого множества вершин | Определение максимального количества параллельных процессов                                  | Неориентированный граф                     | $NP$ | Поиск в глубину с возвратом                                            | Экспоненциальная                                                          |
| 11 | Отыскание наименьшего доминирующего множества вершин | Выявление минимально необходимого количества обслуживающих аппаратов и объектов их установки | Неориентированный (ориентированный) граф   | $NP$ | Ветвей и границ                                                        | Экспоненциальная                                                          |
| 12 | Определение максимального потока                     | Определение максимально возможного потока информации в системе                               | Взвешенный ориентированный граф            | $P$  | Форда - Фалкерсона                                                     | $O(n m^2) - [7]$ ,<br>$O(n^2 m^{1/2}) - [1]$ ,<br>$O(n m \log^2 n) - [1]$ |
| 13 | Установление планарности графа                       | Установление возможности укладки коммуникаций в одной плоскости без их пересечения           | Неориентированный граф                     | $P$  | Поиск в глубину с возвратом (в ширину), динамического программирования | $O(n^3) - [7]$                                                            |

## Заключение

Основными результатами работы являются систематизированное изложение точных методов дискретной оптимизации, их сущности, возможностей и особенностей задач, для решения которых они могут быть использованы. Автор надеется, что предлагаемая статья облегчит научным работникам использование точных методов дискретной оптимизации

при решении новых комбинаторно-оптимизационных задач и даст полезный материал преподавателям курса дискретной математики для технических специальностей.

### Список литературы

1. Нечепуренко М.И., Попков В.К., Майнагашев С.М., Кауль С.Б., Проскуряков В.А., Кохов В.А., Грызунов А.Б. Алгоритмы и программы решения задач на графах и сетях. Новосибирск: Наука. Сиб. отд-ние, 1990. 515 с.
2. Асанов М.О., Баранский В.А., Расин В.В. Дискретная математика: графы, матроиды, алгоритмы. Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001. 288 с.
3. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы: пер. с англ. М.: Издат. дом Вильямс, 2001. 384 с.
4. Галкина В.А. Дискретная математика: комбинаторная оптимизация на графах: учеб. пособие по специальностям "Компьютер. безопасность", "Комплекс. обеспечение информ. безопасности автоматизир. систем" и "Информ. безопасность телекоммуникац. систем". М.: Гелиос АРВ, 2003. 232 с.
5. Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов: пер. с англ. М.: Мир, 1981. 368 с.
6. Дивеев А.И., Северцев Н.А. Метод выбора оптимального варианта технической системы. М.: ВЦ РАН, 2003. 106 с.
7. Кормен Т., Лейзерсон Ч., Риверст Р. Алгоритмы: построение и анализ: пер. с англ. М.: МЦНМО, 2000. 960 с.
8. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. М.: Энергоатомиздат, 1988. 480 с.
9. Новиков Ф.А. Дискретная математика для программистов. СПб.: Питер, 2001. 304 с.
10. Овчинников В.А. Алгоритмизация комбинаторно-оптимизационных задач при проектировании ЭВМ и систем: учеб. для вузов. М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. 288 с.
11. Овчинников В.А. Графы в задачах анализа и синтеза структур сложных систем. М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. 423 с.
12. Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: учеб. пособие. М.: ФИЗМАТЛИТ, 2002. 240 с.
13. Артюхин В.В. Прогнозирование чрезвычайных ситуаций с помощью дискретной оптимизации и современных программных средств // Технологии гражданской безопасности. 2014. Т. 11, № 1 (39). С. 86-91.
14. Ведерников Ю.В., Гарькушев А.Ю., Сазыкин А.М. Математическая формализация задачи оптимального построения информационно-управляющего комплекса мониторинга критически важных объектов // Вопросы оборонной техники. Сер. 16: Технические средства противодействия терроризму. 2014. № 1-2. С. 26-31.
15. Джамрад М., Романченко О.А., Толстикова О.Н. Размещение объекта обслуживания населения на основе метода дискретной оптимизации // Управление большими системами: сб. тр. 2006. № 14. С. 123-134.

16. Домке Э.Р., Жесткова С.А., Акимова В.Ю. Особенности решения задачи маршрутизации транспорта методом «ветвей и границ» // Вестник Московского автомобильно-дорожного государственного технического университета (МАДИ). 2012. № 2. С. 76-79.
17. Дюбин Г.Н., Корбут А.А. Асимптотика жадных методов для задач о ранце: обзор и новые результаты // Обозрение прикладной и промышленной математики. 2008. Т. 15, № 4. С. 744а-746.
18. Забиняко Г.И., Котельников Е.А. Параллельные вычисления в некоторых задачах дискретной оптимизации // Математическое моделирование. 2009. Т. 21, № 9. С. 99-107.
19. Иванко Е.Е. Метод динамического программирования в минимаксной задаче распределения заданий с равноценными исполнителями // Вестник Южно-Уральского государственного университета. Сер. Математическое моделирование и программирование. 2013. Т. 6, № 1. С. 124-133.
20. Колпаков Р.М., Посыпкин М.А. Верхняя и нижняя оценки трудоемкости метода ветвей и границ для задачи о ранце // Дискретная математика. 2010. Т. 22, № 1. С. 58-73.
21. Мещеряков Г., Дроженко В. Применение методов дискретной оптимизации для синтеза структур систем менеджмента // РИСК: Ресурсы, информация, снабжение, конкуренция. 2012. № 1. С. 39-41.
22. Посыпкин М.А., Си Ту Тант Син. Сравнительный анализ эффективности различных вариантов метода динамического программирования для решения оптимизационных задач на этапе размещения элементов микросхем // Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)»: сб. тр. Ч. 2. 2014. С. 97-100.
23. Салий Я.В. Влияние условий предшествования на вычислительную сложность решения маршрутных задач методом динамического программирования // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. 2014. № 1. С. 76-86.
24. Табуров Д.Ю. Решение задач оптимизации вычислительного процесса в локальных сетях информационно-измерительных и управляющих систем с использованием метода ветвей и границ // Приборы. 2010. № 7. С. 50-56.
25. Dzieliński A., Przemysław M. Computer algorithms for solving optimization problems for discrete-time fractional systems // Proc. of the 2013 European Control Conference (ECC), July 17-19, 2013, Zürich, Switzerland. P. 4009-4014.
26. Pardalos P.M., Rodgers G.P. Computational aspects of a branch and bound algorithm for quadratic zero-one programming // Computing. 1990. Vol. 45, iss. 2. P. 131-144. DOI: [10.1007/BF02247879](https://doi.org/10.1007/BF02247879)

## Systematization of Accurate Discrete Optimization Methods

V.A. Ovchinnikov<sup>1,\*</sup>

[\\*yaovchinnikov@gmail.com](mailto:yaovchinnikov@gmail.com)

<sup>1</sup>Bauman Moscow State Technical University, Moscow, Russia

---

**Keywords:** discrete optimization, accurate methods, structural synthesis, decomposition, algorithm, upper and lower bounds, method scheme, computational complexity

---

The object of study of this paper is to define accurate methods for solving combinatorial optimization problems of structural synthesis. The aim of the work is to systemize the exact methods of discrete optimization and define their applicability to solve practical problems.

The article presents the analysis, generalization and systematization of classical methods and algorithms described in the educational and scientific literature.

As a result of research a systematic presentation of combinatorial methods for discrete optimization described in various sources is given, their capabilities are described and properties of the tasks to be solved using the appropriate methods are specified.

### References

1. Nechepurenko M.I., Popkov V.K., Mainagashev S.M., Kaul' S.B., Proskuryakov V.A., Kokhov V.A., Gryzunov A.B. *Algoritmy i programmy resheniya zadach na grafakh i setyakh* [Algorithms and software for solving problems on graphs and networks]. Novosibirsk, Nauka Publ., 1990. 515 p. (in Russian).
2. Asanov M.O., Baranskii V.A., Rasin V.V. *Diskretnaya matematika: grafy, matroidy, algoritmy* [Discrete Mathematics: graphs, matroids, algorithms]. Izhevsk, SIC "Regular and chaotic dynamics" Publ., 2001. 288 p. (in Russian).
3. Aho A.V., Hopcroft J.E., Ullman J.D. *Data Structures and Algorithms*. Addison Wesley, Reading, 2000. (Russ. ed.: Aho A.V., Hopcroft J.E., Ullman J.D. *Struktury dannykh i algoritmy*. Moscow, Publishing House "Vil'yams", 2001. 384 p.).
4. Galkina V.A. *Diskretnaya matematika: kombinatornaya optimizatsiya na grafakh* [Discrete mathematics: combinatorial optimization on graphs]. Moscow, Gelios ARV Publ., 2003. 232 p. (in Russian).
5. Goodman S., Hedetniemi S. *Introduction to the Design and Analysis of Algorithms*. N.Y., McGraw-Hill, 1977. 371 p. (Russ. ed.: Goodman S., Hedetniemi S. *Vvedenie v razrabotku i analiz algoritmov*. Moscow, Mir Publ., 1981. 368 p.).

6. Diveev A.I., Severtsev N.A. *Metod vybora optimal'nogo varianta tekhnicheskoi sistemy* [Method of selecting of optimal variant of technical system]. Moscow, CC of RAS Publ., 2003. 106 p. (in Russian).
7. Cormen T.H., Leiserson C.E., Rivest R.L. *Introduction to Algorithms*. MIT Press, 1990. (Russ. ed.: Cormen T.H., Leiserson C.E., Rivest R.L. *Algoritmy: postroyeniye i analiz*. Moscow, MTsNMO Publ., 2000. 960 p.).
8. Kuznetsov O.P., Adel'son-Vel'skii G.M. *Diskretnaya matematika dlya inzhenera* [Discrete Mathematics for Engineers]. Moscow, Energoatomizdat Publ., 1988. 480 p. (in Russian).
9. Novikov F.A. *Diskretnaya matematika dlya programmistov* [Discrete mathematics for computer programmers]. St. Petersburg, Piter Publ., 2001. 304 p. (in Russian).
10. Ovchinnikov V.A. *Algoritmizatsiya kombinatorno-optimizatsionnykh zadach pri proektirovani EVM i system* [Algorithmization of combinatorial and optimization problems in the design of computers and systems]. Moscow, Bauman MSTU Publ., 2001. 288 p. (in Russian).
11. Ovchinnikov V.A. *Grafy v zadachakh analiza i sinteza struktur slozhnykh system* [Graphs in analysis and synthesis of structures of difficult systems]. Moscow, Bauman MSTU Publ., 2014. 423 p. (in Russian).
12. Sigal I.Kh., Ivanova A.P. *Vvedenie v prikladnoe diskretnoe programmirovaniye: modeli i vychislitel'nye algoritmy* [Introduction to applied discrete programming: models and computational algorithms]. Moscow, Fizmatlit Publ., 2002. 240 p. (in Russian).
13. Artyukhin V.V. Forecasting of Emergencies with Discrete Optimization and Modern Software. *Tekhnologii grazhdanskoi bezopasnosti = Civil security technologies*, 2014, vol. 11, no. 1 (39), pp. 86-91. (in Russian).
14. Vedernikov Yu.V., Gar'kushev A.Yu., Sazykin A.M. Mathematical formalization of the task optimum creation of the management information complex of monitoring crucial objects. *Voprosy oboronnoi tekhniki. Ser. 16: Tekhnicheskie sredstva protivodeistviya terrorizmu = Defense Engineering Problems. Ser. 16. Technical means to combat terrorism*, 2014, no. 1-2, pp. 26-31. (in Russian).
15. Dzhamrad M., Romanchenko O.A., Tolstikova O.N. Placement of object of services of population on the basis of discrete optimization method. *Upravlenie bol'shimi sistemami = Large-scale Systems Control*, 2006, no. 14, pp. 123-134. (in Russian).
16. Domke E.R., Zhestkova S.A., Akimova V.Yu. Inaccuracy of "the branch-and-bound" method in solving traveling salesman problem. *Vestnik MADI*, 2012, no. 2, pp. 76-79. (in Russian).
17. Dyubin G.N., Korbut A.A. Asymptotics of greedy method for the knapsack problem: a review and new results. *Obozrenie prikladnoi i promyshlennoi matematiki*, 2008, vol. 15, no. 4, pp. 744a-746. (in Russian).
18. Zabinyako G.I., Kotel'nikov E.A. Parallel calculations in some problems of discrete optimization. *Matematicheskoe modelirovaniye*, 2009, vol. 21, no. 9, pp. 99-107. (in Russian).

19. Ivanko E.E. Dynamic Programming Method in Bottleneck Tasks Distribution Problem with Equal Agents. *Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Ser. Matematicheskoe modelirovanie i programmirovaniye = Bulletin of the South Ural State University. Ser. "Mathematical Modelling, Programming and Computer Software"*, 2013, vol. 6, no. 1, pp. 124-133. (in Russian).
20. Kolpakov R.M., Posypkin M.A. Upper and lower bounds for the complexity of the branch and bound method for the knapsack problem. *Diskretnaya matematika*, 2010, vol. 22, no. 1, pp. 58-73. (English version of journal: *Discrete Mathematics and Applications*, 2010, vol. 20, no. 1, pp. 95-112. DOI: [10.1515/dma.2010.006](https://doi.org/10.1515/dma.2010.006) ).
21. Meshcheryakov G., Drozhenko V. Adaptation the methods of Discretion optimization for formation structures of management systems. *RISK: Resursy, informatsiya, snabzhenie, konkurentsia*, 2012, no. 1, pp. 39-41. (in Russian).
22. Posypkin M.A., Si Tu Tant Sin. Comparative analysis of efficiency of different variants of the dynamic programming method for solving the problem of optimal placing of elements on the chip. *Vserossiiskaya nauchno-tekhnicheskaya konferentsiya "Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh system (MES)": sb. tr. Ch. 2* [Proc. of the All-Russia Science and Technology Conference "Problems of Advanced Micro- and Nanoelectronic Systems Development". Pt. 2]. Moscow, Institute for Design Problems in Microelectronics of Russian Academy of Sciences Publ., 2014, pp. 97-100. (in Russian).
23. Salii Ya.V. On the effect of precedence constraints on computational complexity of dynamic programming method for routing problems. *Vestnik Udmurtskogo universiteta. Matematika. Mekhanika. Komp'yuternye nauki = The Bulletin of Udmurt University. Mathematics. Mechanics. Computer Science*, 2014, no. 1, pp. 76-86. (in Russian).
24. Taburov D.Yu. The decision of problems of optimization of the computational process in local networks of information-measuring and control systems using the method of branches and borders. *Pribory*, 2010, no. 7, pp. 50-56. (in Russian).
25. Dzieliński A., Przemysław M. Computer algorithms for solving optimization problems for discrete-time fractional systems. *Proc. of the 2013 European Control Conference (ECC)*, July 17-19, 2013, Zürich, Switzerland, pp. 4009-4014.
26. Pardalos P.M., Rodgers G.P. Computational aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, 1990, vol. 45, iss. 2, pp. 131-144. DOI: [10.1007/BF02247879](https://doi.org/10.1007/BF02247879)