

# УСКОРЕННЫЙ АЛГОРИТМ ОДНОСТОРОННЕГО ВЕТВЛЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С БУЛЕВЫМИ ПЕРЕМЕННЫМИ

**Б.И. Юхименко**

---

Одесский национальный политехнический университет,  
просп. Шевченко, 1, Одесса, 65044, Украина; e-mail: pm1987pm@gmail.com

---

В работе приведен ускоренный алгоритм решения задачи линейного программирования с булевыми переменными. Алгоритм относится к классу алгоритмов метода ветвей и границ с односторонним ветвлением. Особенностью является то, что приведен способ получения исходного варианта решения и способ приоритетного ранжирования компонент вектора решения. Дан пример определения оптимального решения согласно предполагаемому алгоритму.

**Ключевые слова:** отсеивание, приоритетная очередь, одностороннее ветвление.

## Введение

Принятие решений в любой сфере деятельности человека неизбежная процедура. Решение всегда выбирается так, чтобы оно было более эффективным, а в некоторых случаях и оптимальным. Такое решение достигается путем сравнения отдельных альтернативных решений, количество которых может быть как бесконечным, так и конечным. С математических позиций конечное множество решений имеют дискретные задачи оптимизации. Дискретная оптимизация, как область науки, является постоянно развивающейся и привлекает множество специалистов. В настоящее время насчитывается большое количество алгоритмов, методов и подходов решения задач данного класса.

## Цель статьи

Значительное место среди методов решения задач дискретной оптимизации занимают комбинаторные методы. В своей сущности эти методы используют конечность множества вариантов. Комбинаторные методы наиболее простые и достойны в практической реализации. Хорошо известный метод ветвей и границ [1] также относится к классу комбинаторных методов, является переборным методом, по своей сложности относится к классу *NP*-полных алгоритмов. В них полный перебор вариантов заменяется частичным перебором. Эффективность алгоритмов предопределяется выделением подмножества перебираемых (перспективных) вариантов. Чем это подмножество содержит меньше вариантов, тем алгоритм работает эффективнее в смысле скорости сходимости. Любой способ увеличения отсеивания вариантов в какой-то степени улучшает скорость сходимости. В данной статье приводится алгоритм метода ветвей и границ с односторонним ветвлением. Увеличение отсеивания неперспективных вариантов, в первую очередь, получается за счет изначального определения исходного варианта. Приведена очередь приоритетов

компонент вектора решений. Это упрощает перебор вариантов и ускоряет получение оптимального.

### Анализ основных исследований и публикаций

В известных алгоритмах метода ветвей и границ для решения задачи целочисленного линейного программирования (ЦЛП) отсеивание неперспективных вариантов ведется различными способами.

В алгоритме Лэнд и Дойч [2], имеющего в основе симплекс метод, поочередно исключаются нецелочисленные значения в районе оптимального значения нецелочисленной компоненты. Пусть эта компонента  $x_k$ . Множество вариантов делится на два подмножества. В одном подмножестве будут рассматриваться варианты, в которых  $x_k \leq [x_k]$ , а во втором –  $x_k \geq [x_k] + 1$  - где  $[x_k]$  - целая часть  $x_k$ . Варианты, в которых компоненты  $x_k$  находится в интервале  $[x_k] < x_k < [x_k] + 1$  не будут рассматриваться.

В алгоритме Литла и др. [3] для решения задачи о коммивояжере, которого можно считать предком метода ветвей и границ в целом, предусмотрено довольно сильное правило отсеивания вариантов. Подмножество вариантов, содержащее неперспективный переезд между парой городов, довольно часто безвозвратно исключается из рассмотрения. Не перспективность переезда оценивается потерей доли значения целевой функции в случае не включения рассматриваемой пары в цикл объезда городов.

В классических алгоритмах метода вервей и границ, использующих правило расширения – суждения множества вариантов [4] для получения оценок и последовательное построение решения [5] как средство разбиения множества вариантов на подмножества отсеивание вариантов осуществляется за счет удачно подобранного способа конкретизации переменных. В работе [6] экспериментально было показано, что на скорость сходимости значительной степени влияет процедура определения компоненты, включаемой в решение и являющейся параметром разбиения множества вариантов на подмножества. На отсеивание вариантов также влияет точность получаемых оценок. Признак оптимальности, практически не подвергавшийся обновлению от момента появления метода, выражается через оценки подмножеств вариантов. Чем ближе значение оценки подмножества к значению целевой функции варианта, тем больше подмножеств исключается из рассмотрения и быстрее выявляется оптимальность варианта.

Алгоритм Балаша [7] с так называемым односторонним ветвлением имеет другой способ отсеивания вариантов. Конкретизируются только единичные компоненты вектора решений. Компоненты принимают нулевые значения, если они не могли стать единичными по следующим причинам. Во-первых, из-за жесткости системы ограничений. Вторая причина состоит в том, что присвоение единичного значения не позволит улучшить значение целевой функции по сравнению с уже имеющимся рекордом. Предложенный автором алгоритма способ выбора компонент-претендентов на значение «1», довольно хороший и оптимальное решение получается быстро. Прием, заменяющий признак оптимальности, не достаточен. Процедура подтверждения того, что найденный вариант оптимальный, часто более вычислительноемкий, чем сама процедура его определения.

Способ отсеивания вариантов, стоящий во главе признака оптимальности в алгоритмах с односторонним ветвлением, полностью зависит от поиска перспективных компонент, которым будет присвоено значение «1». Модификации таких алгоритмов «усиливающих» процедуру отсеивания вариантов, увеличит скорость сходимости.

Нижче буде приведено алгоритм з одностороннім ветвленим. Алгоритм навеян ідеями алгоритма Балаша, однак має деякі особливості, що покращують швидкість збіжності. Спочатку визначається варіант рішення, значення цільової функції якого розглядається як рекорд. Процедура визначення початкового варіанта в обчислювальному сенсі нескладна. В основу цієї процедури закладено спосіб упорядкування за перспективністю компонент вектора рішень. Сама перспективність оцінюється величиною сумарних неувязок в системі обмежень, помноженою на відносну величину відповідуючої компоненти вектора цільової функції. Такого типу упорядкування компонент використовувалося в алгоритмі з двохстороннім ветвленим і значно збільшило швидкість збіжності [6]. Решта частини алгоритму це підтвердження оптимальності отриманого варіанта. Якщо варіант виявиться не оптимальним, то йде його покращення, використовуючи ті ж оцінки перспективності компонент. Не перспективні компоненти відсіваються.

### Основна частина

Алгоритм використовується для рішення задач ЦЛП в постановці

$$Z = \max \sum_{j=1}^n c_j x_j$$

при обмеженнях

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m};$$

$$x_j \in \{0, 1\} \quad j = \overline{1, n},$$

де всі  $a_{ij}$ ,  $b_i$  і  $c_j$  є неотрицательними числами.

Введемо деякі позначення і твердження. Індекси компонент, претендентів на значення «1» складають множину  $V_j^k$ , де  $k$  крок алгоритму

$$V_j^k = \{j / (b_i^k - a_{ij}) \geq 0, \quad \forall_i\};$$

Оцінка перспективності компонент  $x_j$  для  $j \in V_j^k$  визначається за формулою

$$p_j = \sum_{i=1}^m (b_i - a_{ij}) \cdot \sqrt{c_j / \min c_j}, \quad j \in V_j^k.$$

За величиною  $p_j$  визначається черга пріоритетності компонент  $x_j$ , що приймають значення «1». Індекси  $j$  складають множину  $PX$  і визначають пріоритетну чергу компонент  $x_j$ . Множина  $PX$  є домінуючою при побудові вектора  $X = \{x_j\}_{j=1, n}$  на будь-якому рівні роботи алгоритму. Значення цільової функції  $R = \sum_{j=1}^n c_j x_j$  називається рекордом, якщо воно найбільше серед уже розглянутих варіантів.

$B$  – вектор правых частей в системе ограничений,  $B^k$  – рабочий вектор, используемый на  $k$ -ом шаге алгоритма. Во второй части алгоритма используются еще два массива:  $\tilde{P}$  – индексы нулевых компонент, имеющихся в варианте  $X$  и  $\bar{X}$  – рабочий вариант решения задачи.

Ниже приводится описание самого алгоритма, состоящего из двух частей:

- определение исходного варианта и рекорда;
- проверка отсутствия лучшего варианта и формирования нового варианта.

Первая часть

Шаг 1. Формирование массива оценок перспективности компонент вектора решений  $P = \{p_j\} \quad j = \overline{1, n}$ ;

Шаг 2. Упорядочение компонент  $p_j$  массива  $P$  в порядке не возрастания и формирование массива индексов  $PX = \{j_i\}_{i=\overline{1, n}}$ .

Шаг 3. Определение варианта решений  $X = \{x_j\}_{j=\overline{1, n}}$ .

3.1.  $\kappa = 1$ ;  $b_i^k = b_i \quad \forall_i$ ;

3.2.  $j = PX(\kappa)$ ;

3.3. если  $(b_i^k - a_{ij}) \geq 0 \quad \forall_i$ , то  $x_j = 1$ ;  $b_i^k = (b_i^k - a_{ij}) \quad \forall_i$  и к 3.5;

3.4.  $x_j = 0$ ;

3.5.  $k = k + 1$ ; если  $k \leq n$ , то к 3.2;

3.6. вычисление рекорда  $R = \sum_{j=1}^n c_j x_j$ .

Вторая часть алгоритма, связанная с проверкой о возможности улучшения варианта  $X$  и если это возможно, то улучшение.

Шаг 1. Формирование массива  $\tilde{P}$  индексов нулевых компонент варианта  $X$ .

1.1.  $j = 1$ ;  $l = 0$ ;

1.2. если  $x_j > 0$ , то 1.4;

1.3.  $l = l + 1$ ;  $P_l = j$ ;

1.4.  $j = j + 1$ , если  $j \leq n$ , то 1.2;

Шаг 2. Исходные условия.

$k = 1$ ;

2.1.  $Z = 0$ ;  $\bar{x} = \emptyset$ ;

2.2.  $b_i^k = b_i \quad \forall_i$ ;

2.3.  $j = \tilde{P}(k)$ ;  $\bar{x}_j = 1$ ;

2.4.  $b_i^k = (b_i^k - a_{ij}) \quad \forall_i$ ;

Шаг 3. Формирование множества хороших компонент.

3.1.  $V_j^k = \{r = \overline{1, n} / (b_i^k - a_{ir}) \geq 0 \quad \forall_i\}$ ;

3.2. если  $V_j^k = \emptyset$ , то шаг 7;

Шаг 4. Формирование варианта решения.

$s = 1$ ;

4.1.  $r = P_x(s)$ ;

4.2. если  $(r = j)$  или  $(r \in V_j^k)$ , то шаг 5;

4.3.  $\bar{x}(r) = 1$ ;  $Z = Z + C_r$ ;  $b_i^k = (b_i^k - a_{ir}) \quad \forall_i$ ;  $V_j^k = V_j^k - r$ ;

4.4. если  $V_k^k = \emptyset$ , то шаг 5;

4.5. если  $Z + \sum_{j \in V_j^k} c_j \leq R$ , то шаг 7;

Шаг 5.  $s = s + 1$ ; если  $s \leq n$ , то 4.1;

Шаг 6. Если  $Z > R$ , то  $R = Z$ ;  $X = \bar{X}$ ;

Шаг 7.  $k = k + 1$ ; если  $k \leq l$ , то 2.1;

Шаг 8.  $R$  и  $X$  являются результатом.

Пример. Найти оптимальное решение задачи в постановке

$$Z = \max (4x_1 + 7x_2 + 2x_3 + x_4 + 3x_5 + 5x_6)$$

при ограничениях

$$\begin{cases} 2x_1 + 5x_2 + x_3 + 2x_4 + 3x_5 + 4x_6 \leq 14 \\ 4x_1 + 3x_2 + 2x_3 + 4x_4 + 2x_5 + 3x_6 \leq 11 \\ x_j \in \{0,1\} \quad j = \overline{1,6} \end{cases}$$

**Таблица 1.**

Последовательное решение рассматриваемого примера

Переменная и значение	$b_i^k$	$Z$	$V_j^k$	Ситуация
$x_2 = 1$	12 7	7	{1,3,4,5,6}	
$x_6 = 1$	8 4	7+5=12	{1,3,4,5}	
$x_1 = 4$	2 0	12+4=16	$\emptyset$	$x = (110001)$ ; $R = 16$ .
$x_3 = 1$	13 9	2	{1,2,4,5,6}	$\tilde{P} = \{3,4,5\}$
$x_2 = 1$	8 6	2+7=9	{1,4,5,6}	
$x_6 = 1$	4 3	9+5=14	{4,5}	$(14+1+3) > 16$
$x_5 = 1$	1 1	14+3=17	$\emptyset$	$x = (011011)$ ; $R = 17$ .
$x_4 = 1$	12 7	1	{1,2,3,5,6}	$\tilde{P} = \{4,5\}$
$x_2 = 1$	7 4	1+7=8	{1,3,5,6}	
$x_6 = 1$	3 1	8+5=13	$\emptyset$	$13 < 17$
$x_5 = 1$	10 8	3	{1,2,3,4,6}	$\tilde{P} = \{5\}$
$x_2 = 1$	5 5	3+7=10	{1,3,4,6}	
$x_6 = 1$	1 2	10+5=15	{3}	$15+7=17$ улучшений нет $\tilde{P} = \emptyset$

Оценка перспективности каждой искомой величины

$$P_1 = (14 - 2 + 11 - 4) \cdot \sqrt{4} = 19 \sqrt{4} = 38.0;$$

$$P_2 = 17 \sqrt{7} = 45.0;$$

$$P_3 = 22 \sqrt{2} = 31.1;$$

$$P_4 = 19 \sqrt{0,5} = 13.4;$$

$$P_5 = 20 \sqrt{3} = 34.6;$$

$$P_6 = 18 \sqrt{5} = 40.2.$$

$$P_x = \{2, 6, 1, 5, 3, 4\}.$$

Оптимальное решение  $X = (011011)$ ;  $Z = 17$ .

## Заключение

Приведенный алгоритм имеет ряд положительных моментов по сравнению с алгоритмами с односторонним ветвлением. К ним относится определение изначального варианта решения и рекорда – значение целевой функции. Далее, приведенная очередность включения в решении компонент с единичными значениями также значительно улучшает скорость сходимости. Сам перебор вариантов построен на замене нулевых значений вектора решений поочередно единичными, что значительно уменьшает количество перебираемых вариантов. Проведенный пока не большой числовой эксперимент дал утешительные результаты. Конечно, проверить работу необходимо на большом количестве числовых задач и, по возможности, дать теоретическое обоснование в смысле скорости сходимости. Работа над алгоритмом продолжается.

## Список литературы

1. Корбут, А.А. Дискретное программирование / А.А. Корбут, Ю.Ю. Финкельштейн. – М.: Наука, 1969. – 368 с.
2. Land, A.H. An automatic method of solving discrete programming problems / A.H. Land, A. G. Doig // *Econometrica*, 1960. – V. 28. – №3. – С. 497-520.
3. Little, J.D. C. An algorithm for the traveling salesman problem / K.G. Murty, D. W.Sweeney, C. Karel // *Operat. Res.*, 1963. – Т. 11. – №6. – С. 972-989.
4. Шкурба, В.В. Конструктивные подходы к решению задач дискретной оптимизации / В. В. Шкурба // В кн. «IV симпозиум по экстремальным задачам» Тезисы докладов, Каунас, 1969. – С. 15-17.
5. Михалевич, В.С. Последовательные алгоритмы оптимизации и их применение, I, II / В.С. Михалевич // *Кибернетика*. – 1965. – №1. – С. 45-55; №2. – С. 85-89.
6. Юхименко, Б.И. Сравнительная характеристика алгоритмов метода ветвей и границ для решения задач целочисленного линейного программирования / Б.И. Юхименко, Ю.Ю. Козина // *Тр. Одес. политехн. ун-та*. – 2005. – Вып. 2. – С. 199-204.
7. Balas, E. An additive algorithm for solving linear programs with zero-one variables / E. Balas // *Operat. Res.* – 1965. – V.13. – №4. – С. 517-546.

**ПРИСКОРЕНИЙ АЛГОРИТМ ОДНОСТОРОННЬОГО ГАЛУЖЕННЯ ДЛЯ ВИРІШЕННЯ  
ЗАВДАНЬ ЛІНІЙНОГО ПРОГРАМУВАННЯ З БУЛЕВИМИ ЗМІННИМИ**

Б.І. Юхіменко

Одеський національний політехнічний університет,  
проспект Шевченка, 1, Одеса, 65044, Україна; e-mail: pm1987pm@gmail.com

В роботі даний невеликий огляд комбінаторних методів, які представлені в початкових виданнях. Наведені передумови до необхідності удосконалення математичного апарату розв'язання задач цілочисельного лінійного програмування. Основною розробкою є алгоритм з одностороннім гілкуванням для розв'язання задачі лінійного програмування з булевими змінними. Дана покрокова структура алгоритму, вирішений числовий приклад.

**Ключові слова:** відсіювання, пріоритетна черга, одностороннє розгалуження.

**SINGLE ACCELERATE THE ALGORITHM BRANCHES FOR SOLVING LINEAR  
PROGRAMMING PROBLEMS WITH BOOLEAN VARIABLES**

B.I. Ukhimenko

Odessa National Polytechnic University,  
1, Shevchenko Ave., Odessa, 65044, Ukraine; e-mail: pm1987pm@gmail.com

The paper presents a short review of combinatorial methods presented in the primary publications. Preconditions are listed for the need to improve conditions of the mathematical tool of solving problems of integer linear programming. A one-way branching algorithm is the main development is for solving problems of linear programming with Boolean variables. The paper presents turn-based structure of the algorithm, was solved a numerical example.

**Keywords:** screening, priority queue, unilateral branching.