

## ПРОГРАММНЫЙ КОМПЛЕКС MULTIBOOL РЕШЕНИЯ ЗАДАЧИ МНОГОКРИТЕРИАЛЬНОЙ ПСЕВДОБУЛЕВОЙ ОПТИМИЗАЦИИ С НЕПОЛНОЙ НАЧАЛЬНОЙ ИНФОРМАЦИЕЙ О КРИТЕРИЯХ

Козлова М. Г.

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ  
ПР. ВЕРНАДСКОГО, 4, Г. СИМФЕРОПОЛЬ, КРЫМ, УКРАИНА, 95007  
E-MAIL: ART - INF@MAIL . RU

### Abstract

The program implementation description of the multicriterion pseudo-Boolean optimization with incomplete initial information about criteria are given in the article. Basic theoretical results, the problem solving algorithms descriptions for the case of complete and partial information about criteria are taken from [8]. Algorithms of complete search, linear convolution of criteria and local search are implemented in the Visual C++ 6 environment.

*Целью данной работы* является описание программной реализации решения задач многокритериальной псевдодобулевой оптимизации с неполной начальной информацией о критериях. Подходы к решению таких задач обосновываются и исследуются в [8]. Программный комплекс MultiBool предназначен для решения задач безусловной многокритериальной псевдодобулевой линейной оптимизации вида

$$\begin{cases} \max f_i = \sum_{j=1}^n c_{ij}x_j, & i = \overline{1, m} \\ \tilde{x} = (x_1, \dots, x_n) \in B^n \end{cases} \quad (1)$$

где коэффициенты  $c_{ij}$  целевых функций  $f_i$  могут быть неизвестны (имеется лишь информация об их знаках). В этом случае задача (1) сводится к задаче

$$\begin{cases} \max f_i = \sum_{j=1}^n \text{sign}c_{ij}x_j, & i = \overline{1, m} \\ \tilde{x} = (x_1, \dots, x_n) \in B^n \end{cases} \quad (2)$$

для решения которой в [8] предложены и исследованы алгоритмы: а) линейной свертки критериев; б) локальный поиск решения по первому приближению; в) наискорейшего спуска; г) полного перебора решений. Эти алгоритмы легли в основу разработки программного комплекса.

Программный комплекс позволяет решать задачу (1) и в случае полностью заданных коэффициентов  $c_{ij}$ . Он допускает расширение как по методам, так и по классам решаемых задач с неполной информацией. Структура комплекса учитывает необходимость проведения квазиреальных вычислительных экспериментов, что предполагает испытание алгоритмов на широком классе задач, формализуемых случайным

образом, а также фиксирование множества результатов и их статистический анализ. Опишем характерные особенности программного комплекса. Для начала работы программы, необходимо создать текстовый файл (например, input.txt), содержащий информацию о размерности задачи ( $N$ ) количестве критериев ( $M$ ), коэффициентах целевых функций (матрица размерности  $M$  на  $N$ ). В случае, если точные значения коэффициентов не известны, но имеется информация о линейности критериев и знаках коэффициентов, матрица составляется следующим образом:

$$c_{ij} = \begin{cases} -1, & c_{ij} < 0; \\ 0, & c_{ij} = 0; \\ 1, & c_{ij} > 0, i = \overline{1, M}, \quad j = \overline{1, N}. \end{cases} \quad (3)$$

Данные во входном текстовом файле должны быть представлены следующим образом:

$$\begin{array}{l} N \quad \quad \quad // \text{Размерность задачи} \\ M \quad \quad \quad // \text{Количество критериев} \\ c_{11} \quad \dots \quad c_{1j} \quad \dots \quad c_{1N} \\ \vdots \quad \dots \quad \vdots \quad \dots \quad \vdots \quad // \text{Матрица коэффициентов} \\ c_{M1} \quad \dots \quad c_{Mj} \quad \dots \quad c_{MN} \end{array} \quad (4)$$

Таким образом, программная реализация алгоритмов в качестве входных параметров использует: размерность задачи ( $n$ ); число критериев ( $m$ ); матрицу коэффициентов ( $C$ ).

На выходе программа выдает: множество парето-оптимальных решений, множество решений полученных с помощью:

- 1) линейной свертки критериев для всевозможных наборов весовых коэффициентов;
- 2) локального поиска по первому приближению;
- 3) наискорейшего спуска.

В программе используются следующие объекты, классы и функции.

**Класс node.** Объект этого класса представляет элемент списка и содержит информацию о двоичном наборе фиксированной длины и векторе значений целевых функций при реализации алгоритмов локального поиска и полного перебора или вектора весовых коэффициентов при реализации алгоритма линейной свертки критериев. Также показывает, нужно ли в дальнейшем рассматривать данный набор, и указывает на следующий элемент в списке.

**Класс list.** Объект данного класса хранит указатель на начало списка, состоящего из объектов типа node. Может хранить произвольное количество элементов. Существует возможность добавления точки; удаления повторяющихся мажорируемых наборов, содержащихся в другом объекте этого класса; определения числа элементов в

списке, а также числа элементов из другого объекта, имеющих в данном; создания копии данного объекта.

**LDNF**— объект класса list, хранит множество решений соответствующей одно-критериальной задачи для всевозможных наборов весовых коэффициентов;

**DNF**— объект класса list, содержит множество парето-оптимальных решений;

**Second** — объект класса list, хранит множество решений, найденных с помощью алгоритма локального поиска.

Функция **void getCmatr()** формирует массив коэффициентов  $C$ , заполняя их случайными целыми числами в диапазоне от  $-10$  до  $10$ .

Функция **void getNewCmatr()**, формирует массив коэффициентов  $C$ , заполняя их значениями  $-1, 0, 1$  в соответствии со знаком каждого элемента.

Функция **void lambda()** с шагом  $0.1$  генерирует всевозможные наборы весовых коэффициентов. Для каждого набора вычисляется решение соответствующей одно-критериальной задачи. Если это решение не совпадает с найденным на предыдущем шаге, то оно заносится в **LDNF** вместе с набором весовых коэффициентов.

Функция **int generate()** генерирует всевозможные наборы длины  $n$  и заносит в **DNF** вместе с вектором значений целевых функций в данной точке. Затем из списка удаляются элементы, вектор значений целевых функций которых мажорируется соответствующим вектором другого элемента. Возвращается мощность множества парето-оптимальных решений.

Функция **int localall()** позволяет найти  $x_i, i = \overline{1, n}$ , удовлетворяющие необходимому условию

$$\left( \&x_i \right)_{i \in P_0} \quad \left( \&\bar{x}_j \right)_{j \in N_0} = 1$$

и зафиксировать их значения. Здесь  $P_0 = P_1 \cap P_2 \cap \dots \cap P_m$ ,  $N_0 = N_1 \cap N_2 \cap \dots \cap N_m$ , где  $P_i$  - множество номеров переменных, имеющих положительный, а  $N_i$  — множество номеров переменных, имеющих отрицательный коэффициент в линейной функции  $f_i$ ,  $i = \overline{1, m}$ . Если все  $x_i, i = \overline{1, n}$ , определены, полученная точка заносится в **Second**, иначе переменным, подлежащим исследованию присваивается значение  $1$ , а точка заносится в **First** — объект класса list. Пока **First** не пуст, из него извлекается исходный набор, рассматриваются его всевозможные соседние наборы, отличающиеся только не фиксированными компонентами  $x_i, i = \overline{1, n}$ . Вычисляется вектор значений целевых функций, если он мажорирует соответствующий вектор для исходного набора, то соседний набор заносится в **First**. Если же вектора эквивалентны, то соседний набор заносится в **First** и рассматриваются другие соседние наборы исходного вектора. Если векторы значений целевых функций всех соседних наборов мажорируются исходным, равны или эквивалентны ему, то исходный

вектор заносится в Second. Из First удаляются повторяющиеся, мажорируемые и содержащиеся в Second наборы. В конце из списка решений удаляются мажорируемые наборы. Возвращается количество итераций, произведенных данным алгоритмом.

Функция **int localfirst()**. Реализация отличается от функции **int localall()** тем, что в случае, когда вектор значений целевых функций для соседнего набора эквивалентен соответствующему вектору исходного набора, то соседний набор заносится в First, а исходный — в Second.

При запуске программы на выполнение необходимо открыть файл данных. После этого можно воспользоваться одним из четырех методов решения.

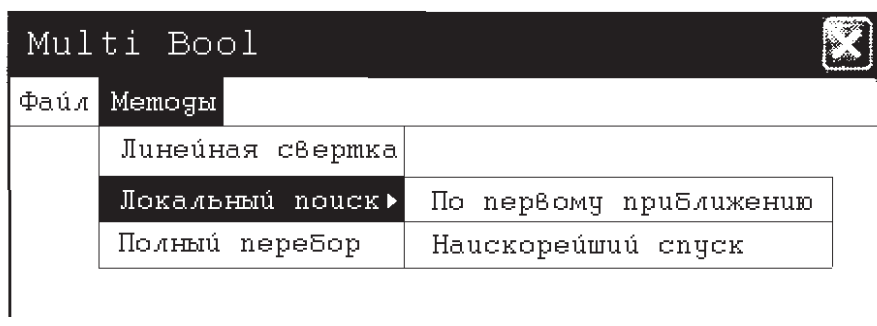


Рис. 1. Выбор метода решения

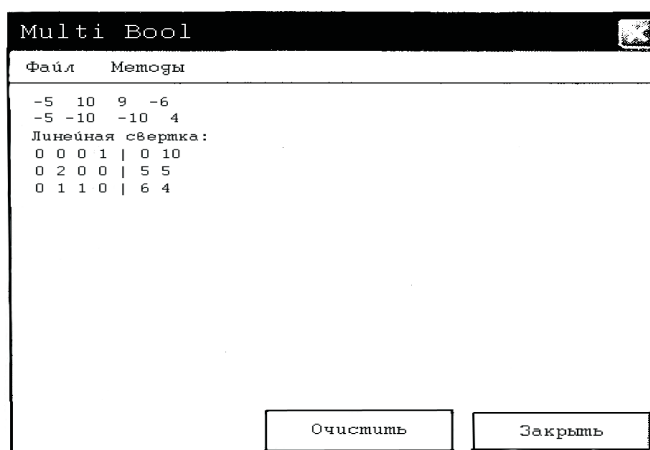


Рис. 2. Результат работы

Результаты работы программы отображаются на экране в следующем порядке:

- выводится матрица коэффициентов;

- указывается название метода, который использовался для решения задачи;
- каждое новое решение выводится с новой строки и имеет вид: вектор  $\tilde{x}$ , разделитель (|), а) вектор весовых коэффициентов (умноженных на 10), для которых был получен данный набор  $\tilde{x}$  - при использовании линейной свертки критериев; б) вектор значений целевых функций, соответствующий данному набору  $\tilde{x}$ , - при использовании локального поиска и полного перебора.

Для локального поиска также отображается число итераций, за которое было найдено решение. Если компонента вектора  $\tilde{x}$  равна 2, это значит, что данная переменная может принимать любое значение из множества 0,1.

Задача может быть решена всеми методами, для этого необходимо последовательно щелчком мыши выбирать соответствующий пункт меню. Результаты будут по очереди отображаться в поле вывода, для их просмотра нужно воспользоваться полосой прокрутки.

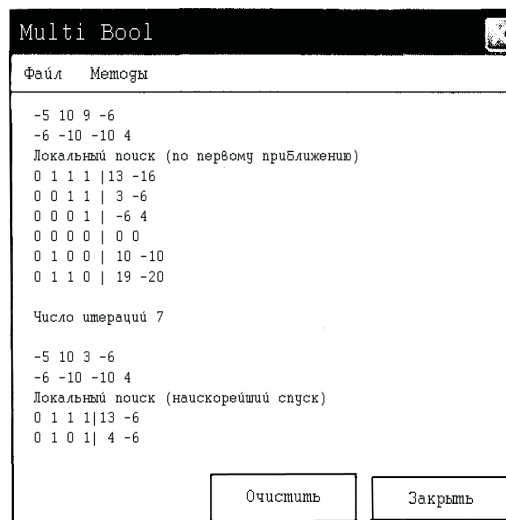


Рис. 3. Результат работы

Результаты работы могут быть сохранены в текстовом файле (например, output.txt). Если программа запускается повторно после сохранения результатов, необходимо заново открыть файл, содержащий начальную информацию.

## ЗАКЛЮЧЕНИЕ.

Реализованы программно в виде допускающего расширение программного комплекса MultiBool следующие алгоритмы решения задач многокритериальной псевдодвулевой оптимизации с неполной начальной информацией о критериях: полного перебора, линейной свертки критериев и локального поиска. Протестированы случаи многокритериальной булевой и псевдодвулевой оптимизации, когда критерии и ограничения заданы как полностью,

так и представленной информацией о линейности и знаках коэффициентов целевых функций.

## СПИСОК ЛИТЕРАТУРЫ

1. Антамошкин А.Н., Масич И.С. Гриды алгоритмы и локальный поиск для условной псевдодобулевой оптимизации Электр. журнал «Исследовано в России». - 2003. - №177 - С.2143- 2149. <http://zhurnal.ape.relarn.ru/articles/2003/177.pdf>
2. Антамошкин А.Н., Масич И.С. Идентификация свойств псевдодобулевых функций Электронный журнал «Исследовано в России». - 2004. - №130. - С.1391-11:\*. <http://zhurnal.ape.relarn.ru/articles/2004/130.pdf>
3. Донской В.И. Задачи псевдодобулевой оптимизации с дизъюнктивным ограничением Журнал вычислительной математики и мат. физики. - 1994. - Т.34, №3. - С.389-398.
4. Донской В.И., Козлова М.Г. Извлечение знаний о свойствах целевой функции в логических системах поддержки принятия решений Искусственный интеллект. - 2000. - ЛИ - С.230-234.
5. Карлин С. Математические методы в теории игр, программировании и экономике. - М Мир, 1964.
6. Козлова М.Г. Синтез сужающих запросов Динамические системы. - 2000. - Вып. 16. - С.208-211.
7. Козлова М.Г. Многокритериальные модели принятия решений с линейными псевдодобулевыми функциями и дизъюнктивным ограничением Искусственный интеллект. - 2000 - №2. - С.67-73.
8. Козлова М.Г., Романчук Л.М. Задача многокритериальной псевдодобулевой оптимизации с неполной начальной информацией о критериях Таврический вестник информатики ж математики. - 2005. - №2. - С. 155-167.
9. Меламед И.И. Сигал И.Х. Исследование линейной свертки критериев в многокритериальном дискретном программировании Журнал вычислительной математики и мат физики. - 1995. - Т.35, №8. - С.1260-1270.
10. Ногин В.Д., Толстых И.В. Использование набора количественной информации об относительной важности критериев в процессе принятия решений Журнал вычислительной математики и мат. физики. - 2000. - Т.40, №11. - С.1593-1601.
11. Ногин В.Д. Принятие решений в многокритериальной среде: количественный подход. - М.: Физматлит, 2002.
12. Ногин В.Д. Принцип Эджворта-Парето и относительная важность критериев в случае нечеткого отношения предпочтения Журнал вычислительной математики и мат. физики. - 2003. - Т.43, №11. - С.1676-1686.
13. Boros E., Hammer P.L. Pseudo-Boolean Optimization Discrete Applied Mathematics. - 2002. - №123(1-3). - P.155-225.