УДК 519.85

И.В. Лысенко, В.О. Бутенко

Национальный аэрокосмический университет имени Н.Е. Жуковского «ХАИ», Харьков

АНАЛИЗ ВОЗМОЖНОСТЕЙ РЕШЕНИЯ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ СРЕДСТВАМИ СИСТЕМ КОМПЬЮТЕРНОЙ МАТЕМАТИКИ

В сравнительном отношении рассматриваются наиболее популярные системы компьютерной математики на предмет решения задач дискретной оптимизации. Описываются разработанные в среде Matlab встроенные функции для решения задач комбинаторной оптимизации, допускающие интерпретацию в виде задач булевого линейного программирования.

Ключевые слова: *дискретная оптимизация, система компьютерной математики.*

Введение

В практике принятия решений в самых разных областях человеческой деятельности приходится сталкиваться с задачами, относящимися к задачам дискретной оптимизации, многие из которых, как известно, являются задачами, принадлежащими классу NP. Существует множество методов, алгоритмов и программных средств решения этих задач. В этой связи справедливо ожидать возможности решения данных задач системами компьютерной математики (СКМ). Они представляют собой специализированные программные пакеты решения математических задач самого разного характера. К числу наиболее популярных СКМ относятся пакеты Mathematica, Maple, Matlab, Mathcad. Кроме того, широкими возможностями для решения математических задач, в том числе оптимизационных, обладает табличный процессор EXCEL.

Очевидно, что необходимым условием систематического изложения какого-либо вопроса является чёткое обозначение его места в системе понятий данной предметной области. В этой связи представляется целесообразным попытаться представить видение авторов взаимосвязи понятий в области дискретной оптимизации (рис. 1).

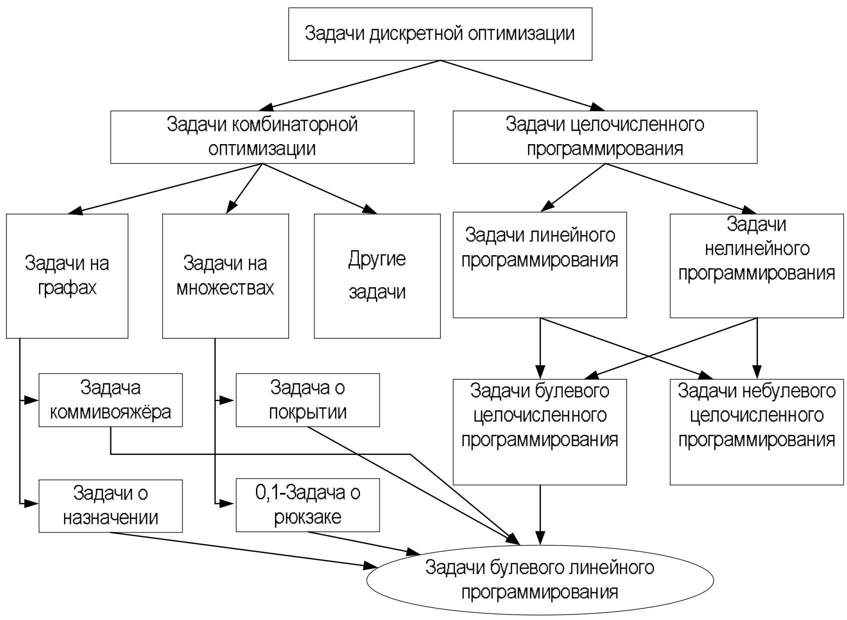

Рис. 1. Схема взаимосвязи понятий в области дискретной оптимизации

Как видно из рис. 1, некоторые задачи комбинаторной оптимизации, представленные в виде задач, интерпретируемых в рамках теории графов (задача коммивояжёра, задача о назначении), а также теории множеств (задача о покрытии множества, 0,1-задача о рюкзаке) могут быть сформулированы как задачи булевого линейного программирования. В этой связи **цель статьи** состоит в том, чтобы проанализировать наиболее популярные системы компьютерной математики на предмет возможности решения в их рамках задач дискретной оптимизации, а также предложить программные реализации в виде функций среды Matlab для решения некоторых задач комбинаторной оптимизации, которые могут быть сведены к задачам булевого линейного программирования.

1. Задачи дискретной оптимизации, решаемые системами компьютерной математики

А. Matlab. Известно, что система *Matlab*, имея мощный набор средств для решения разнообразных задач непрерывной оптимизации в виде пакетов Optimization Toolbox и Global Optimization Toolbox, не содержит встроенных функций для решения задач комбинаторной оптимизации [1,2]. Некоторые исследователи стремятся восполнить этот пробел путём разработки соответствующих функций. Так, например, система Matlog [3], являющаяся расширением Matlab, содержит встроенные функции для решения некоторых задач дискретной оптимизации, интерпретируемых как задачи теории графов, а именно: задача коммивояжёра, задача нахождения кратчайшего пути графа, задача отыскания потока сети минимальной стоимости, задача нахождения минимального остовного дерева.

В [2] дано описание пакета Graph Theory Toolbox, разработанного харьковчанином профессором С.П. Иглиным. В данном пакете представлены функции для решения таких задач, как несимметричная задача коммивояжёра, задача нахождения максимального потока в сети, задача о максимальном паросочетании, задача отыскания минимального остовного дерева, задача нахождения минимального вершинного покрытия и др. Функции данного пакета используют в свою очередь функции, в частности, расширения TOMLAB [4] для решения задач целочисленного и смешанного целочисленного квадратичного программирования в виду того, что, как замечается в [1], «многие задачи на графах могут быть сформулированы в терминах целочисленного линейного программирования (ЦЛП)» и «фактически проблема сводится к тому, чтобы информацию о графе сформулировать как задачу ЦЛП, решить её и вернуть результаты в терминах теории графов».

Б. Mathematica. Данная СКМ является одним из ведущих программных продуктов для обработки численных, символьных, а также графических данных. В ней реализовано базовое множество функций (*LinearProgramming, Minimize, NMinimize, FindMinimum, Maximize, NMaximize*, и *FindMaximum*), комбинации которых позволят пользователю решать основные типы задач линейной и нелинейной оптимизации с учётом ограничениями, а также без ограничений.

Функции *LinearProgramming*, *Minimize*, *Maximize* позволяют решать задачи линейной оптимизации используя симплекс-метод либо же метод внутренней точки *(interior point method)*, в случае решения линейной целочисленной задачи оптимизации – метод ветвей и границ. Примером использования функции *Maximize* может служить решение задачи о рюкзаке. Функции *NMinimize*, *NMaximize* используют различные итеративные методы *(derivative-free iterative methods)* для решения нелинейных задач численной оптимизации, например метод Нелдера (метод деформируемого многогранника). *FindMinimum* и *FindMaximum* позволяют решать задачи нелинейной локальной численной оптимизации путем использования итерационных методов на основе производных, например метода Левенберга-Марквардта либо градиентного метода [7].

Использование встроенного расширения *Combinatorica* СКМ *Mathematica* позволяет использовать около 450 функций для построения и исследования графов, и как следствие, представлены функции решения задач дискретной оптимизации, интерпретируемых как задачи теории графов, среди которых *Dijkstra, ShortestPath, MinimumSpanningTree, NetworkFlow, TravelingSalesman* [8, 9].

Следует отметить, что на платформе *Mathematica* разработано множество коммерческих продуктов, сконцентрированных непосредственно на решении задач оптимизации. Одним из наиболее ярких примеров являются пакеты *Global Optimization 9, MathOptimizer 2* и *MathOptimizer Professional 3*, совместимые с версиями *Mathematica 8 и Mathematica 9* [10].

В. Maple. СКМ *Maple* (от компании *MapleSoft*) является мощным инструментом, содержащим в себе более двух тысяч команд, позволяющих пользователю решать базовое множество математических задач.

Для решения задач оптимизации в данной СКМ реализованы следующие пакеты – *Global Optimization, Optimization* [11], *Simplex* [12], а также в случае решения задач на графах – *Network* [5], *Graph Theory* [13]. С помощью команд *LPSolve, QPSolve, NLPSolve*, содержащихся в пакете *Optimization*, можно получить решение задач линейного, квадратичного и нелинейного программирования, соответственно. Однако существуют некоторые ограничения. К примеру, спецификация метода, в случае использования команд *LPSolve* и *QPSolve*, возможна только в случае решения задач непрерывного программирования. В данном случае предложено использование двух методов: *activeset* и *interiorpoint*, первый из которых реализует метод активных множеств, а второй – метод внутренней точки. В случае, если метод не определен, пакет по умолчанию будет использовать второй метод. Задачи целочисленного программирования с использованием команды *LPSolve* решаются методом ветвей и границ. В данном случае также возможно использование дополнительных опций *assume*, *binaryvariables* либо *integervariables*, позволяющих наложить ограничения неотрицательности искомых значений (*assume-*

nonneg), определить, что переменные имеют бинарные (*binaryvariables*) или целочисленные (*integervariables*) значения [14].

В случае решения задач нелинейной оптимизации (*NLPSolve*) пользователю доступны следующие методы: *quadratic*, *branchandbound*, *modifiednewton*, *nonlinearsimplex*, *pcg* и *sqp*. В основе команды *quadratic* лежит использование метода квадратичной интерполяции, принимающий допущение о том, что целевая функция имеет непрерывную производную первого порядка; *branchandbound* – метод ветвей и границ, использующий условие Липшица; *modifiednewton* – модифицированный метод Ньютона, для использования которого необходимо явное определение, в матричном виде, градиента целевой функции; *nonlinearsimplex* – нелинейный симплекс-метод (метод Нельдера-Мида), рекомендован к использованию в случае повышенных условий точности решения; *pcg* – квазиньютоновский метод сопряженных градиентов с предопределенными условиями ограничения памяти, доступный в случае отсутствия общих ограничений; *sqp* – метод последовательного квадратичного программирования.

Пакет *Simplex* содержит в себе команды, позволяющие решать задачи линейной оптимизации при помощи симплекс-метода, а именно *minimize* (определение минимума функции), *maximize* (максимум функции), *feasible* (проверка на существование решения для данной системы ограничений) и другими командами позволяющими выполнять операции, реализующие отдельные шаги симплекс-метода [12].

Для работы с графами в *Maple* ранее был представлен пакет *Networks*, однако разработчиками предложена также его более новая альтернатива - пакет *Graph Theory*. Данные пакеты представляют собой наборы команд для построения, преобразования, проверки различных свойств графов, поддерживающие работу как с ориентированными, так и неориентированными графами. Базовыми алгоритмами для решения задач оптимизации, реализованными в пакете *Graph Theory* являются – *DijkstrasAlgorithm, BellmanFordAlgorithm* и *ShortestPath* (для определения кратчайшего пути в графе), *MinimalSpanningTree*, *KruskalsAlgorithm* и *PrimsAlgorithm* (определение остова с минимальным весом в неориентированном графе), *TravelingSalesman* (решение задачи коммивояжера) [13].

Г. Mathcad является СКМ, ориентированной на построение интерактивных документов для проведения расчётов с визуализированным сопровождением. Для численного решения задач поиска локального минимума или максимума в *Mathcad* представлены встроенные функции – *Minner*, *Minimize* и *Maximize*. В случае решения задачи линейной оптимизации *Mathcad* по умолчанию использует опцию *Linear*, реализующую метод ветвей и границ. В случае решения задач нелинейной оптимизации предусмотрена возможность выбора алгоритма: метод сопряженных градиентов, Левенберга-Марквардта либо же квазиньютоновский метод. В отличие от рассмотренных выше СКМ, в *Mathcad* не предусмотрено специальное расширение для работы с графами, однако пользователь может достаточно гибко использовать встроенный мощный графический редактор [15]. Однако, как было отмечено ранее, фактически пользователь может сформулировать задачи на графах в терминах ЦЛП [1] и использовать рассмотренные выше функции. Функциональные возможности *Mathcad* могут быть существенно расширены за счёт использования динамически подключаемых библиотек, разработанных с использованием высокоуровневых языков программирования.

Следует также заметить, что возможностями для решения некоторых задач дискретной оптимизации, в частности, задач на графах, обладает табличный процессор EXCEL. Так, с помощью данного средства могут быть решены: задача определения кратчайшего пути сети, задача определения максимального потока сети, задача коммивояжёра, задача о назначении [16].

2. Решение задач комбинаторной оптимизации в системе Matlab

Как было отмечено выше, программная система Matlab не имеет специальных встроенных функций для решения задач комбинаторной оптимизации, а лишь встроенную функцию для решения задачи булевого линейного программирования методом ветвей и границ (*bintprog*). Это даёт возможность реализовать функции в виде m-файлов для решения некоторых задач комбинаторной оптимизации, сводимых к задачам булевого линейного программирования, в частности, задачи коммивояжёра, задачи о назначении, задачи о покрытии множества, 0,1-задачи о рюкзаке (рис. 1).

Что касается *задачи о назначении*, то, как известно, её суть состоит в отыскании наилучшего – в смысле минимальной стоимости – распределения *n* работников по *n* работам при известных затратах c_{ij}, связанных с назначением работника с номером i на работу с номером j (i,j = 1,…,n). Формальная постановка данной задачи имеет вид:

$$f = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \longrightarrow \min , \qquad (1)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j = \overline{1,n}\ ; \ \sum_{j=1}^{n} x_{ij} = 1 \ \forall i = \overline{1,n}\ . \qquad (2)$$

При этом $x_{ij} \in \{0,1\}$ и $x_{ij} = 1$, если i-й работник назначается на j-ю работу и $x_{ij} = 0$ в противном случае.

М-файл-функция с именем *assignprob* для решения задачи о назначении имеет вид, представленный на листинге 1.

```
function [] = assignprob(f)
 n = sqrt(length(f));
 A1 = genMatrixA1(n);
 A2 = repmat(eye(n),1,n);
 A = [A1;A2];
 b = ones(2*n,1);
 [x,fval] = bintprog(f, [], [], A, b);
x
disp('Значение целевой функции')
fval
end
 function A = genMatrixA1(n)
k = 1;
for i = 1:n
 for ii = 1:n * n
 if(k>(i-1)*n && k<((i-1)*n)+n+1)
 A(i,ii) = 1;
 else
 A(i,ii) = 0;
 end;
 k = k + 1;
 end;
 k = 1;
 end;
end.
```

Листинг 1. М-файл-функция *assignprob*

Входным параметром f данной функции является вектор (столбец) значений c_{ij}. В теле данной функции используется функция *genMatrixA1* для формирования части системы ограничений (2).

Следует заметить, что задача о назначении, которая в терминах теории графов определяется как задача поиска наименьшего паросочетания в двудольном графе, может быть решена путём несложной модификации Matlab-функции *grMaxMatch* пакета Graph Theory Toolbox [2], позволяющей решать задачу отыскания максимального взвешенного паросочетания.

При этом последняя функция в качестве входных данных требует, помимо перечисления весов рёбер, указания начальной и конечной вершины каждого из них, в то время как функция *assignprob* в качестве входных данных требует упорядоченный столбец весов рёбер (коэффициентов линейной целевой функции). В свою очередь, путём несложной модификации функции *assignprob* может быть решена задача нахождения максимального паросочетания двудольного графа с чётным числом вершин и равными долями, в которой каждая вершина каждой доли соединена с каждой вершиной другой доли. В этом отношении преимущество функции *grMaxMatch* очевидно, так как она позволяет находить максимальное паросочетание для графа любого вида (а не только двудольного).

Что касается *0,1-задачи о рюкзаке*, то её суть, как известно, состоит в отыскании такого набора предметов из заданного их множества с размерностью a_i и стоимостью c_i каждого, которые, будучи помещёнными в рюкзак размерности A, обеспечивали бы максимальную стоимость. Формальная постановка данной задачи имеет вид:

$$f = \sum_{i=1}^{n} c_i x_i \to \max, \tag{3}$$

$$\sum_{i=1}^{n} a_i x_i \le A\,. \tag{4}$$

При этом $x_i \in \{0,1\}$ и $x_i = 1$, если i-й предмет кладётся в рюкзак и $x_i = 0$ в противном случае.

М-файл-функция с именем *knapsack01* для решения 0,1-задачи о рюкзаке имеет вид, представленный на листинге 2.

```
function func = knapsack01(f, A, b)
 [x,fval]= bintprog(-f, A, b);
 x
 fval = -fval
 end.
```

Листинг 2. М-файл-функция *knapsack01*

Входными параметрами данной функции являются: f – вектор (столбец) значений c_i; A – вектор значений a_i; b – размерность рюкзака.

Сущность *задачи о покрытии множества*, как известно, заключается в следующем. Имеется некоторое множество $M = \{h_1, \ldots, h_m\}$ объектов h_i ($i = 1,\ldots,m$), а также $S = \{S_1, \ldots, S_n\}$ – семейство подмножеств S_j ($j = 1,\ldots,n$), содержащих элементы множества M, и каждому из этих подмножеств поставлено в соответствие некоторое число (вес) c_j. Требуется найти такой набор подмножеств $S^* \subset S$, при котором достигается покрытие множества M с минимальным суммарным весом.

Формально данная задача может быть представлена в виде:

$$f = \sum_{j=1}^{n} c_j x_j \to \min, \tag{5}$$

$$.\sum_{j=1}^{n} a_{ij} x_j \ge 1\,.,\ i = 1,\ldots,m. \tag{6}$$

Здесь $x_j \in \{0,1\}$ и $x_j = 1$, если подмножество S_j входит в покрытие множества M, и $x_j = 0$ в противном случае; $a_{ij} \in \{0,1\}$ и $a_{ij} = 1$, если $h_i \in S_j$ и $a_{ij} = 0$ в противном случае.

М-файл-функция с именем *covering* для решения задачи о покрытии множества имеет вид, представленный на листинге 3.

```
function [x, fval] = covering(f,s)
n = size (f,1); z = 0;
for i=1:n
 b=max(s{i});
 z=max(z,b);
end
A = zeros(z, n);
for i = 1:n
 for j = 1:size(s{i},2)
 A(s{i}(j),i)=1;
 end
end
b = ones(z,1);
[x, fval] = bintprog(f, -A, -b).
```

Листинг 3. М-файл-функция *covering*

Входными параметрами данной функции являются: f – вектор (столбец) значений c_i; s – набор векторов s{j} (j = 1,…,n), содержащих элементы соответствующих подмножеств S_j.

Что касается *задачи коммивояжёра*, то, как известно, она заключается в отыскании минимального по длине гамильтонова цикла в графе.

Формальная постановка данной задачи имеет вид:

$$f = \sum_{i=1}^{n}\sum_{\substack{j=1\\ i\neq j}}^{n} c_{ij}x_{ij} \longrightarrow \min, \qquad (7)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j = \overline{1,n}\,; \quad \sum_{j=1}^{n} x_{ij} = 1 \quad \forall i = \overline{1,n}\,. \qquad (8)$$

Здесь c_{ij} – расстояние между i-й и j-й вершиной в графе; $x_{ij} \in \{0,1\}$ и $x_{ij} = 1$, если путь проходит из i-й вершины в j-ю и $x_{ij} = 0$ в противном случае.

М-файл-функция с именем *tsp* для решения несимметричной задачи коммивояжёра имеет вид, представленный на листинге 4.

Входным параметром f данной функции является вектор (столбец) значений c_{ij}. В теле данной функции используются функции *matrixA2* и *constraintsmatrixA* для формирования ограничений (8), причём функция *matrixA2* является частью функции *constraintsmatrixA*.

В связи с разработанной функцией для решения задачи коммивояжёра следует отметить, что её применение не всегда даёт требуемый результат – гамильтонов цикл; результатом может быть псевдоцикл.

Причина этого, очевидно, в том, что функция *bintprog*, корректно решая задачу булевого ЛП, не ориентируется на ожидаемую интерпретацию получаемого результата, т.е. что полученный результат в виде вектора значений 0 и 1 должен обязательно соответствовать гамильтоновому циклу.

```
function [] = tsp(f)
[s1,s2] = size(f);
i = 1; sq = 1;
while (sq < s1)
 sq = i * i;
 if (sq >= s1)
 sq = i; break;
 end;
 i = i + 1;
end;
n = sq; A = constraintsmatrixA(n);
b = ones(2*n,1);
[x,fval] = bintprog(f, [], [], A, b);
x
disp('Значение целевой функции')
fval
end
function [m] = constraintsmatrixA(n)
k = 1; z = 1;
for i = 1:n
 for ii = 1:n
 if(k == i)
 else
 tmp(i,ii) = z;
 z = z + 1;
 end;
 k = k + 1;
 end;
 k = 1;
end;
k = 1;
for i = 1:n
 for ii = 1:n
 if(k == i)
 else
 c(i,tmp(ii, i)) = 1;
 end;
 k = k + 1;
 end;
 k = 1;
end;
m = vertcat(matrixA2(n), c);
end
function A = matrixA2(n)
k = 1;
for i = 1:n
 for ii = 1:n * (n - 1)
 if(k > (n-1)*(i -1) && k < (n-1)*(i-1)+n)
 A(i,ii) = 1;
 else
 A(i,ii) = 0;
 end;
 k = k + 1;
 end;
 k = 1;
 end;
end.
```

Листинг 4. М-файл-функция *tsp*

В этой связи преимущество функции *grTravSale* пакета Graph Theory Toolbox для решения несимметричной задачи коммивояжёра очевидно, поскольку она всегда даёт минимальный гамильтонов цикл.

В то же время, следует заметить, что в тех случаях, когда обе функции (*tsp* и *grTravSale*) дают один и тот же корректный результат, его нахождение с помощью функции *tsp* осуществляется значительно быстрее, чем с помощью функции *grTravSale* для графа с числом вершин, большим 6. Так, например, для графа с числом вершин, равным 8 и матрицей расстояний, элементы которой есть числа в диапазоне от 6 до 11, время получения результата на основе функции *grTravSale* равно 15,05 сек., а на основе функции *tsp* – 0,046 сек.

Заключение

Что касается возможностей решения оптимизационных задач, то существующие СКМ имеют в своём составе встроенные функции, преимущественно ориентированные на решение задач непрерывной оптимизации. Однако в рамках СКМ созданы и создаются расширения и отдельные программы (функции) для решения задач дискретной оптимизации, прежде всего допускающие трактовку в терминах теории графов.

В рамках данной статьи продемонстрированы возможности создания таких расширений.

Интересным направлений дальнейших исследований является тестирование точности решения задач при использовании различных СКМ, а также временных затрат для получения решений однотипних задач.

Список литературы

1. Иглин С.П. Решение некоторых задач теории графов в MATLAB [Текст] / С.П. Иглин // Exponenta Pro. Математика в приложениях. – 2004. – № 4(4).– С. 28-33.

2. Иглин С.П. Математические расчёты на базе MATLAB [Текст] / С.П. Иглин. – СПб.: БХВ-Петербург, 2005. – 640 с.

3. Matlog: Logistics Engineering MATLAB Toolbox [Электронный ресурс]. – Режим доступа: http://www.ise.ncsu.edu/kay/matlog.

4.TOMLAB Optimization Inc. [Электронный ресурс]. – Режим доступа: http://tomopt.com/tomlab/optimization/miqp.php.

5. Кирсанов М.Н. Графы в Maple. Задачи, алгоритмы, программы [Текст] / М.Н. Кирсанов. – М.: Физматлит, 2007. – 163 с.

6. Охорзин В.А. Оптимизация экономических систем. Примеры и алгоритмы в среде Mathcad [Текст] / В.А. Охорзин. – М.: Финансы и статистика, 2005. – 144с.

7. Ruskeepaa H. Mathematica navigator: Mathematics, Statistics and Graphics. Third Edition [Текст] / H. Ruskeepaa. – Academic Press, 2009. – 1132 p.

8. Maeder R.E. Computer Science with Mathematics:Theory and Practice for Science, Mathematics and Engineering. [Текст] / R.E. Maerder. – Cambridge University Press, 2000. – 389 p.

9. Wolfram Mathematica 9 Documentation center [Электронный ресурс]. – Режим доступа: http://reference.wolfram.com/mathematica/Combinatorica/tutorial/ Combinatorica.html

10. Wolfram Product. All Mathematica Applications [Электронный ресурс]. – Режим доступа: http://www.wolfram.com/products/fields.

11. Maple. Online Help [Электронный ресурс]. – Режим доступа: http://www.maplesoft.com/support/help/Maple/view.aspx?path=Optimization/General/Methods.

12. Говорухин В.Н. Введение в Maple. Математический пакет для всех [Текст] / В.Н. Говорухин, В.Г. Цибулин. – М.:Мир, 1997. – 208 с.

13. Morgan M. Introduction to Maple's GraphTheory Package [Текст] / M. Morgan. – MapleSoft. Maple Conference 2013 Proceedings, 2013. – P.. 1-22.

14. Maple. Online Help [Электронный ресурс]. – Режим доступа: http://www.maplesoft.com/support/help/Maple/view. aspx?path=Optimization/LPSolve.

15. PTC User's Guide. Mathcad 14.0 [Электронный ресурс]. – Режим доступа: http://ru.scribd.com/doc/3239532/Mathcad-14-Users-Guide.

16. Кузьмичов А.І. Математичне програмування в Excel: Навч. посіб. [Текст] / А.І. Кузьмичов, М.Г. Медведев – К.: Вид-во Європ. Ун-ту, 2005. – 320 с.

Поступила в редколлегию 22.05.2013

Рецензент: д-р техн. наук, проф., В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского Н.Е. «ХАИ», Харьков.

АНАЛІЗ МОЖЛИВОСТЕЙ РІШЕННЯ ЗАДАЧ ДИСКРЕТНОЇ ОПТИМІЗАЦІЇ ЗАСОБАМИ СИСТЕМ КОМП'ЮТЕРНОЇ МАТЕМАТИКИ

І.В. Лисенко, В.О. Бутенко

У порівняльному відношенні розглядаються найбільш популярні системи комп'ютерної математики на предмет рішення задач дискретної оптимізації. Надається опис розроблених в середовищі Matlab вбудованих функцій для рішення задач комбінаторної оптимізації, що припускають інтерпретацію у вигляді задач булєва лінійного програмування

Ключові слова: *дискретна оптимізація, система комп'ютерної математики.*

ANALYSIS OF THE POSSIBILITIES OF THE SOLVING OF THE DISCRETE OPTIMIZATION PROBLEMS BY MEANS OF SYSTEMS OF COMPUTER MATHEMATICS

I.V. Lysenko, V.O. Butenko

The most popular systems of computer mathematics for solving of discrete optimization problems are considered comparatively. Developed in the Matlab built-in functions for solving combinatorial optimization problems, that can be interpreted as a problem of Boolean linear programming, are described.

Keywords: *discrete optimization, system of computer mathematics.*