

Министерство образования и науки Украины
Севастопольский национальный технический университет



ЗАДАЧИ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ И ИССЛЕДОВАНИЯ ОПЕРАЦИЙ

Методические указания
к выполнению лабораторных работ №№ 1 – 5
по дисциплине "Математическое программирование
и исследование операций"
для студентов дневной и заочной форм обучения
направления подготовки 6.050201
"Системная инженерия"

Севастополь

2014



Математическое программирование и исследование операций: Методические указания к выполнению цикла лабораторных работ по дисциплине "Математическое программирование и исследование операций" для студентов дневной и заочной форм обучения направления подготовки 6.050201 «Системная инженерия» / Разраб. Е.А.Шушляпин – Севастополь: Изд-во СевНТУ, 2014. – 24с.

В методических указаниях приводится описание лабораторных работ, целью которых является овладение практическими навыками постановок задач математического программирования и исследования операций, умениями работы с компьютерными пакетами EUREKA, Matlab, Mathcad, Simplex. Методические указания предназначены для студентов дневной и заочной форм обучения по направлению подготовки 6.050201 «Системная инженерия».

Методические указания рассмотрены и утверждены на заседании кафедры

Технической кибернетики,

протокол № 9 от "20" июня 2013 г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент

Карапетьян В.А. канд. техн. наук, доцент кафедры ТК.

СОДЕРЖАНИЕ

1. Лабораторная работа № 1 Решение задач математического программирования с помощью пакета MathCad	4
2. Лабораторная работа № 2 Решение задач математического программирования с помощью программного пакета MATLAB.....	7
3 Лабораторная работа № 3. Решение задач линейного программирования с помощью программного пакета SIMPLEX.....	15
4. Лабораторная работа № 4 Решение задач выбора и коммивояжера с помощью пакета SIMPLEX.....	18
5. Лабораторная работа № 5 Решение транспортной задачи.....	21
Приложение. Постановки задач и методические указания к их решению	28
Библиографический список.....	31

1. ЛАБОРАТОРНАЯ РАБОТА № 1

РЕШЕНИЕ ЗАДАЧ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ ПАКЕТА MATHCAD

1.1 ЦЕЛЬ РАБОТЫ

Цель работы - научиться решать задачи математического программирования с помощью компьютерного пакета MathCad.

1.2 ОБЪЕКТЫ И ИНСТРУМЕНТЫ ИССЛЕДОВАНИЯ

Объектами исследования являются задачи линейного и нелинейного программирования. Инструментом исследования являются компьютерная программа MathCad.

1.3 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.3.1 Решение оптимизационных задач с помощью компьютерной программы MathCad

Для решения задач указанного класса с помощью MathCad используются функции Minimize и Maximize. Данные функции введены в MathCad, начиная с версии 8. Использование функций сходно с использованием функций Find или Minerr, используемых при решении систем алгебраических уравнений.

Рассмотрим пример рабочего листа MathCad для следующей задачи:

$$J = 5x_1^2 + 10x_2^2 \rightarrow \min,$$

$$x_1 + x_2 = 4,$$

$$x_1 \leq 2,3$$

В MathCad решение этой задачи имеет вид:

$$f(x1,x2) := 5 \cdot x1^2 + 10 \cdot x2^2$$

$$x1 := 1 \quad x2 := 1$$

Given

$$x1 + x2 = 4$$

$$x1 \leq 2.3$$

$$z := \text{Minimize}(f, x1, x2) \quad z = \begin{pmatrix} 2.3 \\ 1.7 \end{pmatrix} \quad f(z_0, z_1) = 55.35$$

Как видим, вначале следует задать целевую функцию критерия и начальную точку. Начальная точка нужна для работы использованного в MathCad метода оптимизации. В вариантах таблицы П.1 для нелинейных задач начальные точки заданы, обозначены вектором x^0 , но, как правило, их значения выбираются произвольно, сам процесс решения задачи производится многократно, исходя из разных начал

ных точек. При этом могут получиться разные решения и оптимальные значения критерия, из которых следует выбрать наилучшие. Для линейных задач, где решение не зависит от выбора начальной точки, ее можно задавать в виде нулевого вектора. После задания начальной точки обязательно следует слово Given, после чего набирается система ограничений (в данном примере – одно равенство и одно неравенство), а далее записывается функция minimize (для задачи минимизации) или maximize (для задачи максимизации). Заметим, что жирный знак равенства, указываемый в уравнениях, набирается только из панели со знаками операций отношения (в русифицированном MathCad эта панель называется «Булево»). Первым аргументом функций minimize, maximize является имя целевой функции, а остальными аргументами (их число не оговаривается) – оптимизируемые переменные. Для того, чтобы на рабочем листе появилось решение и значение целевой функции, необходимо отобразить результат посредством обычного знака равенства из клавиатуры или панели «Арифметика». Заметим, что по умолчанию векторы в MathCad нумеруются с нуля (в данном примере вектором или массивом является z). Индексы массива набираются посредством выбора x_n , а степени – выбором x^n из панели «Арифметика».

1.3.2 Подготовка к решению задач и счет

Для решения задачи математического программирования необходимо записать ее математическую постановку в виде критерия и ограничений. Заметим, что ограничения при этом должны иметь вид системы уравнений и неравенств (ограничения типа включений здесь учесть нельзя), после чего набрать задачу аналогично приведенному выше примеру. В случае отсутствия решения на рабочем листе появится выделенный красным цветом оператор, в процессе которого произошла аварийная ситуация (например, выделенная красным цветом функция minimize вместе с аргументами). При надвижении крест-курсора на выделенную область появится всплывающая подсказка о причине аварийной ситуации.

1.4 ПРОГРАММА РАБОТЫ

При выполнении **части 1** необходимо изучить пункты 1.3.1 – 1.3.2 настоящих указаний и получить разрешение на дальнейшее выполнение работы.

При выполнении **части 2** необходимо:

- Загрузить MathCad.
- Выбрать из таблицы П.1 и справочной системы пакета SIMPLEX (см. раздел 1 настоящих указаний) данные, соответствующие назначенному преподавателем варианту
- Набрать рабочий лист для задачи линейного программирования, выполнить отладку.
- Сделать то же, что в предыдущем пункте, для задачи нелинейного программирования.

При выполнении **части 3** необходимо:

- Получить решение и сохранить рабочий лист с решением в файле для текстового процессора WORD; это можно сделать, посредством выделения («Пра

ка»-«Выделить все» или мышью) всех элементов рабочего листа и копирования их в буфер обмена.

- Дополнить полученный документ фамилиями членов бригады, номером варианта, выводами; в выводах привести итоги сравнения решений задач, полученных с помощью пакета Mathcad, а также сравнительную оценку удобства решения задач.
-

1.5 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для каждой из математических постановок задач показать а) критерий; б) ограничения; в) целевую функцию.
2. Показать начальную точку и рассказать о ее назначении.
3. Что такое «нелинейное» и «линейное» программирование?
4. Показать нелинейные члены в постановке задачи нелинейного программирования.
5. Сколько переменных и ограничений в линейной и нелинейной задачах?
6. Какие ограничения называются параметрическими и есть ли они в постановках линейной и нелинейной задач?

2. ЛАБОРАТОРНАЯ РАБОТА № 2

РЕШЕНИЕ ЗАДАЧ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ ПРОГРАММНОГО ПАКЕТА MATLAB

2.1 ЦЕЛЬ РАБОТЫ

Цель работы - научиться использовать программный пакет MatLab для решения задач математического программирования.

2.2 ОБЪЕКТЫ И ИНСТРУМЕНТЫ ИССЛЕДОВАНИЯ

Объектами исследования являются оптимизационные задачи линейного и нелинейного программирования. Инструментом исследования является программный пакет MatLab.

2.3 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.3.1 Общие сведения о средствах оптимизации пакета MatLab

Пакет имеет набор функций для решения оптимизационных задач математического программирования, размещенных в каталоге `...\toolbox\optim`. Здесь имеется множество **m**-файлов (т.е. файлов с исходными текстами программ и функций), предназначенных для указанных целей. Имеются также демонстрационные программы, иллюстрирующие использование средств оптимизации совместно с организацией диалогов, вывода информации и пр.

В пакете реализованы различные методы нелинейного программирования, относящиеся к методам нулевого (поисковые), первого (градиентные) и второго (ньютоновские) порядков, а также квазиреализации. Квазиреализациями называются методы, в которых производные высоких порядков определяются через производные более низких порядков, вычисленных в нескольких точках (имеются в виду производные целевой функции по оптимизируемым параметрам). Методы минимизации для задач без ограничений реализованы в процедурах **fminu**, **fminunc**, **fmins**, **fminsearch**, **leastsq**, **lsqnonlin**. В процедурах **fminu**, **fminunc** (**fminunc** рекомендована вместо устаревшей **fminu**, которая в последующих версиях будет упразднена) реализованы градиентный метод наискорейшего спуска (Steepest Descent), квазиньютоновские методы DFP (Davidon-Fletcher-Pawell), BFGS (Broyden-Fletcher-Goldfarb-Shanno), GM (Gill, Murrey). В **fminsearch**, (**fminsearch** рекомендована вместо **fmins**) - поисковый метод Нелдера-Мида (Nelder-Mead). В процедурах **leastsq**, **lsqnonlin** (**lsqnonlin** рекомендована вместо **leastsq**) – методы Гаусса-Ньютона (Gauss-Newton) и Левенберга-Марквардта (Levenberg-Marquardt) для задач с критериями вида $\sum f(x)^2$. Для задач с ограничениями предназначены процедуры **fminbnd** (для минимизации функции одной переменной с ограничениями), **fmincon**

(для минимизации функции многих переменных со смешанными ограничениями из неравенств «не больше» и равенств).

Для задач линейного программирования предназначены процедуры **lp** и **linprog** (**linprog** рекомендована вместо **lp**), где использован прямо-двойственный симплекс-метод..

Имеются также процедуры для решения задач квадратического программирования (**quadprog**), минимаксных задач и др. Полный перечень задач и соответствующих процедур приведен в справочной системе MatLab. Просмотреть этот перечень можно, набрав в командном окне пакета команду

>> help demo optim или **>>help optim**

Все перечисленные процедуры реализованы в файлах с такими же именами и расширениями **.m** (например, процедура **fminunc** - в файле **fminunc.m**).

2.3.2 Правила работы с пакетом

Для помощи при работе с пакетом следует использовать справочную систему, доступ к которой производится через команду **help** либо **help <имя>**. В первом случае (**help**) в командное окно выводится перечень подключенных к Matlab каталогов, во втором случае (**help <имя>**) дается информация о содержимом каталога либо комментарии к процедуре. Например, после исполнения **help optim** в командное окно выводится перечень процедур каталога **optim**, а после исполнения **help fmincon** – о процедуре **fmincon**. Полезны также демонстрационные примеры, в частности те, которые приводятся в командном окне после исполнения команды **help optim – optdemo, tutdemo** и другие.

Для лабораторной работы достаточно просмотреть **tutdemo**, где приведены примеры использования процедуры **fmincon** для задач нелинейного программирования с ограничениями. Ниже приведены основные правила использования данной процедуры.

2.3.3 Решение задач нелинейного программирования с ограничениями

Пакет предназначен для решения задачи нелинейного программирования следующего вида:

$$J = f(x) \rightarrow \min,$$

$$A_1 x \leq b_1,$$

$$A_2 x = b_2,$$

$$\Phi_1(x) \leq 0,$$

$$\Phi_2(x) = 0,$$

$$lb \leq x \leq ub.$$

Здесь (в порядке записей) представлены: критерий с целевой функцией $f(x)$; линейные ограничения типа «не больше»; линейные ограничения типа равенств; нелинейные ограничения типа «не больше»; нелинейные ограничения типа равенств; параметрические ограничения. Матрицы A_1, A_2 , а также векторы $b_1, b_2, \Phi_1, \Phi_2, lb, ub$ (

«less boundary» и «up boundary», т.е. «нижняя граница» и «верхняя граница») задаются по правилам, понятным из текстов Matlab-программ для задач 1 и 2 (см. ниже).

Можно использовать два способа решения. Первый способ ориентирован на решение в пошаговом режиме из командного окна и рекомендуется для относительно простых задач, не требующих ввода многих данных. Кроме того, первый способ не может применяться к задачам с нелинейными ограничениями (когда заданы Φ_1 и/или Φ_2). Второй способ - программный, когда предварительно набираются **m**-файлы (главная и подчиненные процедуры), а затем главная программа запускается на счет посредством опции **run** либо из командного окна. Второй способ применяется для задач с нелинейными ограничениями, а также более сложных задач.

Главное отличие первого и второго способов - в определении целевой функции и левых частей ограничений. Указанные способы рассмотрим на примере двух задач.

Задача 1.

$$J = 10x_1^2 + 5x_1x_2 + 4x_2^2 \rightarrow \min,$$

$$x_1 + x_2 \leq 10,$$

$$x_1 + 4x_2 \geq 1.$$

Решим ее первым способом. Для этого в командном окне последовательно набираем и исполняем команды («» – это символы приглашения к набору команды, появляющиеся в строке автоматически):

```
» fun=inline('10*x(1)^2+5*x(1)*x(2)+4*x(2)^2');
```

```
» A1=[1,1;-1,-4]
```

```
A1 =
```

```
1    1
-1   -4
```

```
» b1=[10;-1]
```

```
b1 =
```

```
10
-1
```

```
x0 =
```

```
5
5
```

```
» [x,f]=fmincon(fun,x0,A1,b1)
```

Warning: Trust region method does not currently solve this type of problem, switching to line search.

> In D:\MATLAB53\toolbox\optim\fmincon.m at line 190

Optimization terminated successfully:

Search direction less than 2*options.TolX and

maximum constraint violation is less than options.TolCon

Active Constraints:

```
2
```

```
x =
```

```
-0.0417
```

0.2604

f =

0.2344

»

Комментарии:

- Обратите внимание на то, что перед набором матрицы A1 и вектора b1 задача приведена к неравенствам типа «не больше».
- Результаты всего, что набирается в строке без точки с запятой в конце, выводятся на экран.
- Предупреждающее сообщение «Warning...» говорит о том, что использовались два метода «Trust region» и «Line research» и автоматическое переключение с первого на второй после неточного решения первым методом.
- Дальнейшие сообщения свидетельствуют об успешном завершении задачи и удовлетворении дополнительных критериев останова по аргументу (**TolX**) и максимальному нарушению ограничений (**TolCon**), которые в данном случае не заданы.
- Указано, что активны все два ограничения, т.е. точка минимума находится на границе как по первому, так и по второму неравенствам.
- На экран выведены координаты точки минимума и соответствующее значение критерия, заданные в командной строке **[x,f]=fmincon....** Результаты выведены, так как в конце данной командной строки нет точки с запятой.

Задача 2.

$$J = 10x_1^2 + 2x_1x_2 + 4x_2^2 \rightarrow \max,$$

$$10x_1 + 8x_2 \leq 3000,$$

$$2x_1 + 4x_2 \leq 800,$$

$$x_1^2 + x_2^2 \leq 10000,$$

$$x_1 \geq -10, x_2 \geq 10.$$

Поскольку в задаче есть нелинейные ограничения, воспользуемся вторым способом, подготовив и сохранив в файлах следующие программные модули

%Файл Main1LR4.m

x0=[0;0]

options=optimset('LargeScale','off','Display','iter');

A=[10 8;2 4]

b=[3000;800]

lb=[-10;10]

ub=[]

```
[x,fval,exitflag,output]=fmincon('exfunLR4',x0,A,b,[],[],lb,ub,...
'exogr1LR4',options)
```

%Файл exfunLR4.m

function f=exfunLR4(x)

```

f=-10*x(1)^2-2*x(1)*x(2)-4*x(2)^2;
%end

%Файл exogrLR4.m
function [c,ce]=exogr1LR4(x)
c=[x(1)^2+x(2)^2-10000];
ce=[];
%end

```

Комментарии.

- В главной программе вначале задается начальная точка, затем параметры **options**, матрица и вектор линейных ограничений типа «не больше», вектор нижних и верхних границ (последний пустой ввиду их отсутствия), обращение к процедуре **fmincon** с выходными параметрами в квадратных скобках слева.
- Параметры из **optimset** задаются парами имя-значение; имя **'LargeScale'** указывает на альтернативу выбора методов для больших (**Large**) или средних (**medium**) размерностей. Если указано **'off'**, подключаются методы **medium**, если **'on'** - метод класса **large** (он в пакете один - «**trust region**»). Вторая пара задает выдачу в командное окно итераций метода. Подробнее смотри через **help optimset**.
- Выходные параметры **fmincon** имеют смысл: **x** - вектор оптимального решения; **fval** - оптимальное значение критерия; **exitflag** - флаг выхода (>0 при нормальном исходе, =0 при превышении заданного числа вычислений целевой функции, <0 при отсутствии решения); **output** - структура с полями **iterations** – количество итераций, **funcCount** – количество вычислений функции, **stepsize** – размер шага, **algorithm** – класс и перечень подключенных алгоритмов, **firstorderopt** и **cgiterations** – количества итераций алгоритмов **CG** и **first-order** (если они используются).
- Входные параметры **fmincon** по порядку следования такие: 1) строка с именем процедуры вычисления целевой функции; 2) начальная точка; 3) матрица линейных ограничений типа «не больше»; 4) вектор правых частей неравенств; 5) матрица линейных ограничений типа «равно»; 6) вектор правых частей равенств; 7) вектор нижних границ; 8) вектор верхних границ; 9) строка с именем процедуры для ограничений; 10) выборка параметров из **optimset**. Могут быть еще дополнительные параметры (см. через **help fmincon**).
- Процедура для целевой функции критерия имеет один выходной скалярный параметр, а для ограничений – два векторных (для векторов невязок неравенств и равенств), даже если ограничения имеют только неравенства или только равенства. Так как в **exogrLR4** параметры **c** и **ce** – векторы (правда, каждый только из одной компоненты), то определяющие их выражения заключены в квадратные скобки.

Ниже приведены результаты расчета, взятые из командного окна после исполнения команды

```

>> Main1LR4
x0 =

```

12

0

0

A =

10 8

2 4

b =

3000

800

lb =

-10

10

ub =

[]

Iter	F-count	f(x)	max		Directional		Procedure
			constraint		Step-size	derivative	
1	3	-400.008	-0.0001		1	-6.8e+003	
2	7	-40000.8	-30		1	-1.07e+005	Hessian modified twice
3	11	-732516	6.208e+004		1	6.31e+005	Hessian modified
4	25	-731899	6.202e+004		0.000977	6.31e+005	Hessian modified
5	36	-726981	6.154e+004		0.00781	6.26e+005	Hessian modified
6	54	-727019	6.154e+004		-6.1e-005	6.25e+005	Hessian modified twice
7	58	-237138	1.347e+004		1	1.1e+005	
8	63	-205284	1.235e+004		0.5	1.19e+005	
9	67	-103942	1768		1	1.64e+004	
10	71	-88154.9	68.61		1	-1.14e+004	Hessian modified
11	75	-102278	520.1		1	-4.8e+003	
12	79	-110704	897.3		1	9.05e+003	
13	83	-101863	23.61		1	240	
14	87	-101623	0.01517		1	0.154	
15	91	-101623	9.071e-007		1	9.02e-006	Hessian modified
16	95	-101623	1.579e-008		1	1.6e-007	Hessian modified

Optimization terminated successfully:

Magnitude of directional derivative in search direction

less than 2*options.TolFun and maximum constraint violation

is less than options.TolCon

Active Constraints:

5

x =

98.7087

16.0182

fval =

-1.0162e+005

exitflag =

1

output =



iterations: 16
 funcCount: 97
 stepsize: 1
 algorithm: 'medium-scale: SQP, Quasi-Newton, line-search'
 firstorderopt: []
 citerations: []

2.3.4 Решение задач линейного программирования

Пакет предназначен для решения задач линейного программирования вида

$$\begin{aligned}
 z &= c^T x \rightarrow \min, \\
 A_1 x &\leq b_1, \quad A_2 x = b_2, \quad lb \leq x \leq ub.
 \end{aligned}
 \tag{3.1}$$

Здесь (в порядке записей) представлены: критерий с линейной целевой функцией $c^T x = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$; линейные ограничения типа «не больше»; линейные ограничения типа равенств; параметрические ограничения. Матрицы A_1, A_2 , а также векторы b_1, b_2, lb, ub (от «less boundary» и «up boundary», т.е. «нижняя граница» и «верхняя граница») задаются по правилам, понятным из следующего ниже текста Matlab-программы для задачи линейного программирования:

$$\begin{aligned}
 z &= 10x_1 + 5x_2 \rightarrow \min, \\
 10x_1 + 8x_2 &\geq 3000, \\
 0.02x_1 + 0.04x_2 &\leq 8, \\
 x_1 &\geq 0, \quad x_2 \geq 0.
 \end{aligned}$$

Как видим, в (3.1) присутствуют три группы ограничений – неравенства типа «не больше», равенства и двустороннее параметрическое ограничение.

Для решения задачи используется процедура **LinProg** с параметрами:

LinProg(c, A₁, b₁, A₂, b₂, lb, ub).

Текст главной программы, хранящейся в m-файле и обращающейся после ввода исходных данных к **LinProg**, может быть таким

```

A=[-10 -8;0.02 0.04;-1 0;0 -1]
b=[-3000;8;0;0]
c=[10;5]
x=linprog(c,A,b)
  
```

Здесь использован один выходной параметр - оптимальное решение. Для более детальной информации можно получить вектор выходных параметров, как в **Main1LR4** (смотри подраздел 3.3).

Ниже приведены результаты решения

```

A =
-10.0000  -8.0000
  0.0200   0.0400
 -1.0000    0
  0  -1.0000
  
```

14

b =

-3000

8

0

0

c =

10

5

Optimization terminated successfully.

x =

233.3333

83.3333

2.4 ПРОГРАММА РАБОТЫ

В работе требуется решить две задачи – линейного и нелинейного программирования. Варианты задач линейного и нелинейного программирования берутся из задания к лабораторной работе №1 (см. таблицу 1.1).

При выполнении **части 1** необходимо изучить пункты 2.3.1 – 2.3.4 настоящих указаний и получить разрешение на дальнейшее выполнение работы.

При выполнении **части 2** необходимо:

- Подготовить файлы главных программ для задач нелинейного и линейного программирования.
- Подготовить файлы с процедурами для задачи нелинейного программирования.

При выполнении **части 3** необходимо:

- Решить обе задачи, скопировав полученные результаты в буфер обмена.
- Вызвать WORD и извлечь содержимое буфера обмена поочередно – после счета задачи НП и после счета задачи ЛП.
- Дополнить полученный документ фамилиями членов бригады, номером варианта, выводами; в выводах привести итоги сравнения решений задач, полученных с помощью пакета Mathcad, а также сравнительную оценку удобства решения задач.

2.5 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. К какому виду должна быть приведена математическая постановка задачи *нелинейного* программирования для ее решения в MATLAB?
2. К какому виду должна быть приведена математическая постановка задачи *линейного* программирования для ее решения в MATLAB?
3. Каким образом задаются *нелинейные* ограничения?
4. Каким образом задаются *линейные* ограничения?
5. Каким образом задаются *параметрические* ограничения?
6. В каких случаях используется функция **inline**?

3 ЛАБОРАТОРНАЯ РАБОТА № 3. РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ ПРОГРАММНОГО ПАКЕТА SIMPLEX

3.1 ЦЕЛЬ РАБОТЫ

Цель работы - научиться использовать программный пакет SIMPLEX для решения задач линейного программирования.

3.2 ОБЪЕКТЫ И ИНСТРУМЕНТЫ ИССЛЕДОВАНИЯ

Объектами исследования являются оптимизационные задачи линейного программирования. Инструментом исследования является программный пакет SIMPLEX.

3.3 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

3.3.1 Общие сведения о программном пакете SIMPLEX

Пакет SIMPLEX разработан в 1996 году на кафедре технической кибернетики СевНТУ, предназначен для эксплуатации на IBM - совместимых компьютерах с операционными средами Windows. Пакет предназначен для решения задач линейного программирования, содержащих до 150 переменных и 50 ограничений в виде равенств и нестрогих неравенств. Условия неотрицательности при этом подразумеваются и в число ограничений не входят. Пакет снабжен развитой справочной системой с контекстными ссылками, содержащей как правила работы с пакетом, так и разнообразные сведения по задачам линейного программирования - постановки задач (23 постановки в словесном и формализованном виде и 23 постановки только в словесном виде), указания по формализации, методы решения. Постановки задач в словесном виде предназначены для выдачи в качестве заданий студентам. В пакете имеется режим пошагового решения, когда в удобном виде выдаются результаты решения после каждого шага метода.

3.3.2 Правила работы с пакетом

Правила работы с интерактивным интерфейсом пакета изложены в справочной системе, вызов которой производится нажатием кнопки "?" горизонтального меню либо клавиши F1 клавиатуры компьютера. Справочную систему можно запустить отдельно, щелкнув дважды по ее наименованию Lpr1.hlp в директории, где размещается пакет SIMPLEX.

Перед счетом задачи ее необходимо подготовить к решению. Эта подготовка включает формализацию задачи, т.е. запись на математическом языке. При этом зада

формулируется в общей форме с условиями неотрицательности. О формах задач линейного программирования смотри подразделы «Эквивалентные формы» и «Преобразования форм» справочной системы. Необходимые для подготовки задачи сведения имеются в подразделе «Правила формализации» и в примерах постановок задач из подраздела «Постановки задач – Примеры задач – Примеры содержательных и формализованных постановок задач».

3.4 ПРОГРАММА РАБОТЫ

В работе требуется формализовать и решить одну задачу линейного программирования, заданную в словесном виде. Вариант задачи линейного программирования содержится в справочной системе пакета SIMPLEX (пункт «Постановки задач – Примеры задач – Примеры содержательных постановок задач»). Литература, откуда взяты постановки задач, приведена в разделе «Библиография» справочной системы. Номер варианта дается преподавателем.

При подготовке задачи следует загрузить пакет SIMPLEX, вызвать справочную систему и изучить раздел «Постановки задач».

Лабораторная работа включает три части, состоящие из отдельных этапов.

ЧАСТЬ 1. ИЗУЧЕНИЕ ИНТЕРАКТИВНЫХ СРЕДСТВ (2 ЧАСА)

1. Изучение раздела справочной системы «Работа с пакетом» (1 час).
2. Решение заданного в таблице П.1 варианта задачи линейного программирования (1 час).

ЧАСТЬ 2. ПОДГОТОВКА ЗАДАЧИ К РЕШЕНИЮ (2 ЧАСА)

1. Изучение раздела справочной системы «Постановки задач» (1 час).
2. Математическая постановка задачи. Вариант задачи берется из справочной системы пакета SIMPLEX (пункт «Постановки задач – Примеры задач – Примеры содержательных постановок задач»). При этом условия целочисленности, если они по смыслу задачи должны быть, в математическую постановку не включаются, а целочисленное решение образуется путем округления полученного нецелочисленного решения с помощью опции пакета «Счет – Проверка решения». Полученная в итоге постановка задачи в общей форме используется в качестве исходных данных для решения задачи на компьютере (1 час).

ЧАСТЬ 3. РЕШЕНИЕ ЗАДАЧИ (2 ЧАСА)

1. Изучение раздела справочной системы «Методы решения задачи» на примере заданного варианта задачи: приведение к канонической форме; если каноническая форма имеет единичный базис, записать начальную симплекс-таблицу; в противном случае сформулировать вспомогательную задачу, преобразовать ее к канонической форме с единичным базисом и представить последнюю в виде симплек

таблицы; по полученной начальной симплекс-таблице выполнить одну итерацию симплекс-метода, записать базисное решение для новой симплекс-таблицы, указать, оптимальное оно или нет и почему (1 час).

2. Ввод полученных на этапе 2 части 2 исходных данных задачи в компьютер (15 мин).
3. Получение решения в режимах «Итерации» (пошаговом) и «Счет» со сравнительным анализом «ручного» (2, 3, 4 этапы части 3) и компьютерного решения на каждом шаге симплекс-метода. (15 мин). После первых тестовых прогонок задачи и ее верификации сгенерировать отчет (см. «Вывод результатов» в справочной системе), который включить в отчет о работе.
4. Подготовка и защита отчета по работе (1 час). Отчет должен включать: титульный лист установленной формы; полное описание задания к работе; подробное, с промежуточными выкладками, отображение этапов 2, 3, 4 части 2; листинг отчета; вывод о результатах сравнения «ручных» и компьютерных результатов.

4. ЛАБОРАТОРНАЯ РАБОТА № 4 РЕШЕНИЕ ЗАДАЧ ВЫБОРА И КОММИВОЯЖЕРА С ПОМОЩЬЮ ПАКЕТА SIMPLEX

4.1. ЦЕЛЬ РАБОТЫ

Цель работы - научиться решать задачи выбора и коммивояжера с помощью компьютерного пакета SIMPLEX.

4.2. ОБЪЕКТЫ И ИНСТРУМЕНТЫ ИССЛЕДОВАНИЯ

Объектами исследования являются задачи исследования операций (выбора и коммивояжера). Инструментом исследования являются компьютерная программа SIMPLEX.

4.3. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

4.3.1. Общие сведения о компьютерной программе SIMPLEX

Описание программы см. в методических указаниях к лабораторной работе №2.

4.3.2. Подготовка к решению задач

Для решения перечисленных задач исследования операций необходимо записать их как задачи линейного программирования (математические постановки см. в [1]), а также в справочной системе пакета. Следует иметь в виду, что задача коммивояжера, совпадающая во многом по постановке с задачей выбора, содержит дополнительное требование в виде замкнутости (нерасщепленности) маршрута, которое не может быть формализовано в виде линейных равенств или неравенств. По этой причине решение задачи коммивояжера как задачи линейного программирования может получиться расщепленным. Кроме того, в этой задаче для запрета петель необходимо диагональные элементы матрицы стоимостей перевозок задавать большими (теоретически - бесконечными). В этом состоят особенности решения задачи коммивояжера симплекс-методом.

Кроме того, для всех трех задач, имеющих двухиндексное обозначение переменной, необходимо формально выполнить переобозначение одноиндексной переменной. Однако при использовании пакета SIMPLEX этого делать не нужно, так как двухиндексные обозначения можно задать в полях для альтернативных имен, после чего ввод коэффициентов критерия и матрицы ограничений не представляет трудностей. Следует иметь в виду, что матрица стоимостей для данных задач вводится построчно (или по столбцам) в вектор коэффициентов критерия в соответствии тем, в каком порядке в поле альтернативных имен задавалась переменная с двумя индексами (если, например, в порядке $x_{11}, x_{12}, \dots, x_{21}, x_{22}, \dots$ – то по строкам; если

же в порядке $x_{11}, x_{21}, \dots, x_{12}, x_{22}, \dots$ – то по столбцам). С учетом этого задаются и коэффициенты матрицы ограничений.

4.4. ПРОГРАММА РАБОТЫ

Часть 1

- Загрузить SIMPLEX.EXE.
- Выбрать вариант задач из приложения к настоящим указаниям (номер варианта задает преподаватель).
- Ввести исходные данные для задачи выбора, открыть отчет, запустить задачу на счет в обычном (не итерационном режиме);
- Сохранить отчет, задав индивидуальное имя, имеющее признак задачи выбора.

Часть 2

- Загрузить SIMPLEX.EXE. Ввести исходные данные, которые имеют следующие отличия от исходных данных задачи выбора: флажок «max» переключить на «min»; диагональные элементы матрицы критерия $C_{11}, C_{22}, C_{33}, C_{44}$ заменить большими числами (например, 1000); открыть отчет; запустить задачу на счет в обычном (не итерационном режиме).
- Сохранить отчет, задав индивидуальное имя, имеющее признак задачи коммивояжера.

При подготовке итогового отчета:

- Решить задачи выбора и коммивояжера «вручную», сравнить полученные результаты с компьютерными решениями.
- Подготовить отчет, включающий: титульный лист установленной формы; цель работы; математические постановки задач выбора и коммивояжера; два полученных на компьютере отчета по каждой из задач; выводы. В выводах привести результаты «ручного» и компьютерных решений, которые должны совпадать для задачи выбора по меньшей мере в части оптимального значения критерия. Для задачи коммивояжера решения могут не совпадать и по значению критерия по той причине, что компьютерный маршрут может оказаться расщепленным. Для того чтобы это увидеть, следует перейти от одноиндексных обозначений переменных к двухиндексным, по которым можно определить, является ли полученный маршрут замкнутым или расщепленным. Например, решение в одноиндексном обозначении $x_1=0, x_2=1, x_3=0, x_4=0, x_5=1, x_6=0, x_7=0, x_8=0, x_9=0, x_{10}=0, x_{11}=0, x_{12}=1, x_{13}=0, x_{14}=0, x_{15}=1, x_{16}=0$ или двухиндексном обозначении $x_{11}=0, x_{12}=1, x_{13}=0, x_{14}=0, x_{21}=1, x_{22}=0, x_{23}=0, x_{24}=0, x_{31}=0, x_{32}=0, x_{33}=0, x_{34}=1, x_{41}=0, x_{42}=0, x_{43}=1, x_{44}=0$ имеет переходы из пункта 1 в пункт 2 (т.к. $x_{12}=1$), из пункта 2 в пункт 1 ($x_{21}=1$), из пункта 3 в пункт 4 ($x_{34}=1$) и из пункта 4 в пункт 3 ($x_{43}=1$). Поскольку эти переходы не образуют замкнутый цикл, соответствующий данному решению маршрут является расщепленным (из пункта 1 в пункт 2 и обратно; из пункта 3 в пункт 4 и обратно).

4.5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каков смысл критерия и ограничений в задачах выбора и коммивояжера?
2. Каким образом задаются исходные данные задач в пакете?
3. Почему в двухиндексной (матричной) форме допустимого решения ровно по одной единице в каждой строке и каждом столбце?
4. По каким причинам «ручное» и компьютерное решения задачи коммивояжера могут не совпадать друг с другом?
5. Прокомментировать этапы «ручных» решений задач выбора (венгерским методом) и коммивояжера (методом Литтла).

5. ЛАБОРАТОРНАЯ РАБОТА № 5 РЕШЕНИЕ ТРАНСПОРТНОЙ ЗАДАЧИ

5.1. ЦЕЛЬ РАБОТЫ

Цель работы - научиться решать транспортные задачи с помощью компьютерных пакетов SIMPLEX и TRANSP.

5.2. ОБЪЕКТЫ И ИНСТРУМЕНТЫ ИССЛЕДОВАНИЯ

Объектом исследования является одна из задач исследования операций (закрытая транспортная задача). Инструментами исследования являются компьютерные программы SIMPLEX и TRANSP.

5.3. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

5.3.1. Общие сведения о компьютерных программах

Описание программы SIMPLEX см. в методических указаниях к лабораторной работе № 3.

Ниже приведено описание компьютерной программы TRANSP.

Программа разработана на кафедре технической кибернетики СевНТУ и предназначена для решения закрытой и открытой транспортных задач с выводом на экран промежуточных результатов. Последняя опция удобна для использования программы в учебном процессе. В программе реализованы поиск опорного плана методом минимального элемента строки и улучшения плана методом потенциалов. Постановка закрытой транспортной задачи, методы и примеры ее решения приведены в [1]. В отличие от [1], где для поиска опорного плана описаны и показаны на примере методы северо-западного угла и минимального элемента, в данной программе использована упрощенная версия метода минимального элемента – метод минимальных элементов строк. Его отличие от метода минимального элемента состоит в том, что минимальный элемент ищется не во всей таблице C , а в первой по счету строке, в которой остались невычеркнутые элементы. Так, для задачи с матрицей стоимости « C » и векторами « a » и « b », показанных в таблице 5.1

Таблица 5.1 – Исходные данные для примера транспортной задачи (матрица стоимостей « C », вектора объема поставок « a » и объема закупок « b »)

	$b_1=40$	$b_2=50$	$b_3=70$	$b_4=90$	$b_5=90$
$a_1=100$	4	1	2	6	9
$a_2=120$	6	4	3	5	7
$a_3=120$	5	2	6	4	8

опорный план, полученный данным методом и представленный таблицей, сходной по структуре с таблицей 5.1, но содержащей вместо матрицы стоимостей матрицу плана X , имеет вид:

Таблица 5.2 – Опорный план, построенный методом минимальных элементов строк

	$b_1=40$	$b_2=50$	$b_3=70$	$b_4=90$	$b_5=90$
$a_1=100$		50	50		
$a_2=120$	10		20	90	
$a_3=120$	30				90

Ниже приведена интерфейсная панель программы с исходными данными из таблицы 5.1.

Д:\TK\TranspColor\3x5.trp

Файл Счет Справка

Транспортная задача (открытая/закрытая)
min

Количество поставщиков $n = 3$ $2 \leq n \leq 50$

Количество потребителей $m = 5$ $2 \leq m \leq 50$

Протокол работы

Объем закупок

B	40	50	70	90	90
---	----	----	----	----	----

Объем поставок

A	100	120	120
---	-----	-----	-----

Матрица стоимостей перевозок

c(i,j)	1	2	3	4	5
1	4	1	2	6	9
2	6	4	3	5	7
3	5	2	6	4	8

Стоимость $F =$

alpha

beta

После ввода данных задачи посредством клавиатуры или из ранее сохраненного файла производится ее счет путем выбора опции «Счет-Запуск», после чего появляется матрица опорного плана перевозок, полученная методом минимальных элементов строк, а также две командные всплывающие кнопки «Пок. контур» и «Итерации». Кроме того, поля векторов Alpha и Beta для потенциалов заполняются значениями. Значения потенциалов (иногда в литературе их обозначают буквами «u» и «v» соответственно) определяются из системы уравнений

$$\alpha_s + \beta_j = C_{ij}, \quad \text{для всех } i \in \{1, 2, \dots, n\}, \quad j \in \{1, 2, \dots, m\}, \quad \text{где } X_{ij} \geq 0 \text{ являются базисными.} \quad (5.1)$$

Так, для опорного плана из таблицы 6.2 система уравнений для потенциалов следующая: $\alpha_1 + \beta_2 = 1$, $\alpha_1 + \beta_3 = 2$, $\alpha_2 + \beta_1 = 6$, $\alpha_2 + \beta_3 = 3$, $\alpha_2 + \beta_4 = 5$, $\alpha_3 + \beta_1 = 5$, $\alpha_3 + \beta_5 = 8$, а ее решение следующее: $\alpha_1 = 0$ (задано, т.к. число переменных больше числа уравнений на единицу), $\alpha_2 = 1$, $\alpha_3 = 0$, $\beta_1 = 5$, $\beta_2 = 1$, $\beta_3 = 2$, $\beta_4 = 4$, $\beta_5 = 8$.

Панель задачи после запуска на счет выглядит следующим образом

Как видим, вместо матрицы стоимостей теперь отображается план перевозок, а поля для потенциалов получили значения, соответствующие данному плану. Для удобства восприятия в плане перевозок цветом выделены клетки матрицы плана, соответствующие свободным компонентам базисного решения. Напомним, что базисное решение транспортной задачи должно содержать ровно $n+m-1$ базисных компонент, а остальные компоненты, называясь «свободными», в базисном решении равны нулю.

Следующий этап решения транспортной задачи – проверка плана на оптимальность по неравенствам

$$\alpha_s + \beta_j \leq C_{ij}, \quad \text{для всех } i \in \{1, 2, \dots, n\}, \quad j \in \{1, 2, \dots, m\}, \quad \text{где } X_{ij} = 0 \text{ являются свободными,} \quad (5.2)$$

которые для оптимального плана все должны выполняться.

Результат указанной проверки можно увидеть, нажав командную кнопку «Пок. контур», в результате чего появится окно:

	1	2	3	4	5	6
1	(-1)			(2)	(1)	
2	(-)	(2)			(+)(-2)	
3	(+)	(1)	(4)	(0)	(-)	
4						
5						
6						

В этом окне показан как результат проверки неравенств (6.2), так и следующий этап метода потенциалов – контур перераспределения перевозок. Результаты проверки неравенств показаны в виде чисел, находящихся в круглых скобках. Контур же показан в виде значков «+» и «-», также находящихся в круглых скобках. Неравенства (5.2) для опорного плана из таблицы (6.2) следующие: $\alpha_1 + \beta_1 \leq 4$, $\alpha_1 + \beta_4 \leq 6$, $\alpha_1 + \beta_1 \leq 4$, $\alpha_1 + \beta_5 \leq 9$, $\alpha_1 + \beta_1 \leq 4$, $\alpha_2 + \beta_2 \leq 4$, $\alpha_2 + \beta_5 \leq 7$, $\alpha_3 + \beta_2 \leq 2$, $\alpha_3 + \beta_3 \leq 6$, $\alpha_3 + \beta_4 \leq 4$. Подставляя решения $\alpha_1 = 0$, $\alpha_2 = 1$, $\alpha_3 = 0$, $\beta_1 = 5$, $\beta_2 = 1$, $\beta_3 = 2$, $\beta_4 = 4$, $\beta_5 = 8$, получаем значения $\delta_{ij} = C_{ij} - \alpha_i - \beta_j$ из таблицы. При этом положительные значения δ_{ij} означают, что неравенство выполняется, а отрицательные свидетельствуют о невыполнении неравенства. Так, в левой верхней клетке стоит число «-1», полученное в результате $\delta_{11} = C_{11} - \alpha_1 - \beta_1 = 4 - 0 - 5 = -1$.

Построение контура начинается от клетки с наибольшим по модулю отрицательным значением δ_{ij} , т.е. в данном случае от клетки с $\delta_{25} = -2$, которая помечена (+). Далее построен контур циклического переноса перевозок, проходящий через начальную клетку (2-5) и клетки (2-1), (3-1), (3.5). Новый план образуется путем мысленного наложения полученного контура на матрицу плана перевозки и переноса перевозки в матрице плана перевозки от помеченной (-) клетки с наименьшим значением перевозки $x_{\min}$ (в данном случае клетки (2-1) с $x_{\min} = 10$) по контуру. При этом в клетках, помеченных плюсами, $x_{\min}$ добавляется, а в минусовых клетках – вычитается.

Для каждого из планов в поле вывода F выводится значение критерия, соответствующее текущему плану.

Новый план можно посмотреть, нажав командную кнопку «Итерации». Далее выполняются остальные итерации до тех пор, пока кнопка «Итерации» не исчезнет.

В опции меню «Файл» предусмотрены пункты «Новая задача», «Открыть», «Сохранить исходные данные», «Сохранить результат».

Для ввода данных с клавиатуры необходимо выбрать пункт «Новая задача».

Для сохранения введенных данных, т.е. матрицы C и векторов A и B, необходимо выбрать пункт «Сохранить исходные данные» и тип сохраняемого файла «Файл для транспортной задачи»

Для открытия ранее сохраненной задачи необходимо выбрать пункт «Открыть» и выбрать файл с расширением «trp».

Полученное решение посредством пункта «Сохранить результат» выводится в текстовый файл типа `prt`, который далее можно либо просмотреть, либо распечатать. В этом файле, распечатка которого приведена ниже, свободные компоненты плана помечены «-1».

Дата/время: 18.02.2014 15:50:25

Размерность задачи:

$n=3$ $m=5$

Оптимальное значение стоимости $F=1400$

Матрица оптимального плана

10	50	40	-1	-1
-1	-1	30	-1	90
30	-1	-1	90	-1

В левом нижнем углу панели имеется командная кнопка с переменным именем, предназначенная для просмотра либо матрицы стоимостей перевозок («Исх.данные») либо матрицы плана перевозок («Результат»).

В правом верхнем углу помещено табло, отображающее результаты работы в виде сообщений о номере итерации либо оптимальности решения.

5.3.2. Подготовка к решению задач

Для решения перечисленных задач исследования операций необходимо записать их как задачи линейного программирования (математические постановки см. в [1]), а также в справочной системе пакета. Следует иметь в виду, что задача коммивояжера, совпадающая во многом по постановке с задачей выбора, содержит дополнительное требование в виде замкнутости (нерасщепленности) маршрута, которое не может быть формализовано в виде линейных равенств или неравенств. По этой причине решение задачи коммивояжера как задачи линейного программирования может получиться расщепленным. Кроме того, в этой задаче для запрета петель необходимо диагональные элементы матрицы стоимостей перевозок задавать большими (теоретически - бесконечными). В этом состоят особенности решения задачи коммивояжера симплекс-методом.

Кроме того, для всех трех задач, имеющих двухиндексное обозначение переменной, необходимо формально выполнить переобозначение одноиндексной переменной. Однако при использовании пакета `SIMPLEX` этого делать не нужно, так как двухиндексные обозначения можно задать в полях для альтернативных имен, после чего ввод коэффициентов критерия и матрицы ограничений не представляет трудностей. Следует иметь в виду, что матрица стоимостей для данных задач вводит построчно (или по столбцам) в вектор коэффициентов критерия в соответствии

тем, в каком порядке в поле альтернативных имен задавалась переменная с двумя индексами (если, например, в порядке $x_{11}, x_{12}, \dots, x_{21}, x_{22}, \dots$ – то по строкам; если же в порядке $x_{11}, x_{21}, \dots, x_{12}, x_{22}, \dots$ – то по столбцам). С учетом этого задаются и коэффициенты матрицы ограничений.

5.4. ПРОГРАММА РАБОТЫ

Часть 1

- Загрузить SIMPLEX.EXE.
- Выбрать вариант задач из приложений к настоящим указаниям (номер варианта задает преподаватель).
- Ввести исходные данные для транспортной задачи, открыть отчет, запустить задачу на счет в обычном (не итерационном режиме); вводимые данные при этом такие же, как для задачи выбора в лабораторной работе № 4, за исключением флажка типа задачи, который здесь имеет значение «min», а также значений вектора b , который теперь должен состоять из чисел таблицы П.3 приложения (в задаче выбора вектор b состоял из единиц).
- Сохранить отчет, задав индивидуальное имя, имеющее признак задачи выбора.

Часть 2

- Запустить программу TRANSPcolor и ввести данные транспортной задачи в соответствии с их назначением (см. пример выше); выписать и поместить в отчет все промежуточные результаты счета: опорный план и планы итераций; для каждой из итераций уравнения типа (5.1) для определения потенциалов и неравенства для проверки плана на оптимальность типа (5.2), а также их решения (из полей α и β интерфейсной панели задачи; значения критерия для опорного, текущих и оптимального планов).
- Сохранить отчет, задав индивидуальное имя, имеющее признак задачи коммивояжера.

Подготовить отчет, включающий: титульный лист установленной формы; цель работы; математические постановки задач выбора и коммивояжера; два полученных на компьютере отчета по каждой из задач; выводы. В выводах привести результаты «ручного» и компьютерных решений, которые должны совпадать для задачи выбора по меньшей мере в части оптимального значения критерия. Для задачи коммивояжера решения могут не совпадать и по значению критерия по той причине, что компьютерный маршрут может оказаться расщепленным. Для того чтобы это увидеть, следует перейти от одноиндексных обозначений переменных к двухиндексным, по которым можно определить, замкнутый маршрут или расщепленный. Например, решение в одноиндексном обозначении $x_1=0, x_2=1, x_3=0, x_4=0, x_5=1, x_6=0, x_7=0, x_8=0, x_9=0, x_{10}=0, x_{11}=0, x_{12}=1, x_{13}=0, x_{14}=0, x_{15}=1, x_{16}=0$ или двухиндексном обозначении $x_{11}=0, x_{12}=1, x_{13}=0, x_{14}=0, x_{21}=1, x_{22}=0, x_{23}=0, x_{24}=0, x_{31}=0, x_{32}=0, x_{33}=0, x_{34}=1, x_{41}=0, x_{42}=0, x_{43}=1, x_{44}=0$ имеет переходы из пункта 1 в пункт 2 (т.к. $x_{12}=1$), из пункта 2 в пункт 1 ($x_{21}=1$), из пункта 3 в пункт 4 ($x_{34}=1$) и из пункта 4 в пункт 3 ($x_{43}=1$). Поскольку эти переходы не образуют замкнутой

цикл, соответствующий данному решению маршрут является расщепленным (из пункта 1 в пункт 2 и обратно; из пункта 3 в пункт 4 и обратно).

5.5. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каков смысл критерия и ограничений в транспортной задаче?
2. Что такое закрытая и открытая транспортные задачи?
3. Каким образом задаются исходные данные транспортной задачи в пакетах SIMPLEX и TRANSP?
4. Почему количество базисных компонент оптимального плана равно $n+m-1$?
5. По каким причинам «ручное» и компьютерное решения задачи коммивояжера могут не совпадать друг с другом?
6. Показать, каким образом получен опорный план.
7. Прокомментировать этапы решения транспортной задачи методом потенциалов.

ПОСТАНОВКИ ЗАДАЧ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ИХ РЕШЕНИЮ

В лабораторных работах №№ 1 – 3 требуется решить задачи двух классов – линейного и нелинейного программирования. В лабораторных работах №№ 4, 5 решаются методами линейного программирования три задачи исследования операций – задачи выбора и коммивояжера (работа № 4) и закрытая транспортная задача (работа № 5). Варианты задач линейного программирования для работ №№ 1, 2 берутся в справочной системе пакета SIMPLEX из пункта «Постановки задач – Примеры задач – Примеры содержательных и формализованных постановок задач», для работы № 4 – из пункта «Постановки задач – Примеры задач – Примеры содержательных постановок задач» по указанному в таблице П.1 ее наименованию, вариант задачи нелинейного программирования – непосредственно из таблицы П.1.

Варианты заданий к лабораторной работе № 4 берутся из таблицы П.2, где для задачи выбора данные используются в неизменном виде, а для задачи коммивояжера – с заменой диагональных элементов С11, С22, С33, С44 на большие числа (теоретически – бесконечности, практически – достаточно 1000). Данные для лабораторной работы № 5 берутся в неизменном виде из таблицы П.2 (значения элементов матрицы стоимостей) и таблицы П.3 (объемы производства и потребления транспортной задачи)

Таблица П.1 – Варианты задания для работ №№ 1, 2

№ вар.	Задача №1 (линейного программирования)	Задача №2 (нелинейного программирования)
1	Об оптимальной смеси сортов твердого топлива	$100(x_2 - x_1^2)^2 + (1 - x_1^2) \rightarrow \min$, $x_1 + 1 \geq 0, 1 - x_2 \geq 0$, $4x_2 - x_1 - 1 \geq 0, 1 - 0,5x_1 - x_2 \geq 0, \quad x^0 = (-1; 1)$
2	Максимизации выпуска комплектов изделий	$(x_1 - 2)^2 + (x_2 - 2)^2 \rightarrow \min$, $3 - x_1 - x_2 \geq 0, x_2 \geq 0$, $10x_1 - x_2 - 2 \geq 0, \quad x^0 = (0, 2; 0)$
3	Максимизации прибыли	$(x_1 - 2)^2 + (x_2 - 2)^2 \rightarrow \min$, $4,5 - x_1^2 - x_2^2 \geq 0, x_2 \geq 0$, $10x_1 - x_2 - 2 \geq 0, \quad x^0 = (0, 2; 0)$
4	Оптимизации землеройных работ	$(x_1 - 2)^2 + (x_2 - 2)^2 \rightarrow \min$, $2,25 - x_1x_2 \geq 0, x_2 \geq 0$, $10x_1 - x_2 - 2 \geq 0, \quad x^0 = (0, 2; 0)$
5	Минимизации времени выполнения плана	$(x_1 - x_2)^2 + x_2 \rightarrow \min$, $x_1 + x_2 - 1 \geq 0, x_1, x_2 \geq 0$, $3 - x_1 + 2x_2 \geq 0, \quad x^0 = (3; 0)$

6	Оптимального раскроя	$x_1 + x_2 \rightarrow \min,$ $x_1^2 + x_2^2 - 1 = 0, \quad x^0 = (0;1)$
7	Оперативного календарного планирования	$2x_1 - 3(x_2 - 4)^2 \rightarrow \max,$ $10 - x_1^2 - x_2^2 \geq 0,$ $9 - x_1^2 - (x_2 - 4)^2 \geq 0,$ $x_1 \geq 0, \quad x^0 = (2;2)$
8	Минимизации неравномерности графика выпуска продукции	$-x_1 x_2^2 x_3^3 / 81 \rightarrow \min,$ $x_1^3 + x_2^2 + x_3 - 13 = 0,$ $x_2^2 x_3^{-1/2} - 1 = 0, \quad x^0 = (1; 1,7; 9)$
9	Планирования ассортимента	$-x_1 - 2x_2 - 4x_1 x_2 \rightarrow \min,$ $9 - (x_1 - 2)^2 - (x_2 - 1)^2 \geq 0,$ $x_1 \geq 0, x_2 \geq 0, \quad x^0 = (5;1)$
10	О назначении	$(x_1 - 1)^2 + (x_2 - 1)^2 + (x_3 - 1)^2 \rightarrow \min,$ $x_1 + x_2 + x_3 - 2 \leq 0,$ $1,5x_1^2 + x_2^2 + x_3 - 2 = 0, \quad x^0 = (1; 0,5; 0,5)$
11	Оптимального планирования отраслей	$4x_1 - x_2^2 + x_3^2 - 12 \rightarrow \min,$ $20 - x_1 - x_2 = 0,$ $x_1 + x_3 - 7 = 0,$ $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, \quad x^0 = (2,4; 3,6; 4,4)$
12	Использования кормов	$4x_1 - x_2^2 + x_3^2 - 12 \rightarrow \min,$ $20 - x_1 - x_2 = 0,$ $x_1 + x_3 - 7 = 0,$ $x_3 / x_1 - 2 = 0,$ $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, \quad x^0 = (2,4; 3,6; 4,4)$
13	Производства кормов	$x_1 + x_2^2 + x_3^2 \rightarrow \min,$ $1 - 64x_1^{-2} x_2^{-4} x_3^{-6} \geq 0,$ $x_1, x_2, x_3 \geq 0, \quad x^0 = (0,5; 0,5; 0,5)$
14	Оптимизации размещения и структуры посевов	$1/x_1 + 1/x_2 \rightarrow \min,$ $0,5x_1^2 + x_2^2 - 1 = 0,$ $x_1, x_2 \geq 0, \quad x^0 = (0,75; 0,75)$
15	Оптимизация многоэтапных перевалок груза	$((x_1 + 3)^3 / 3) + x_2^2 \rightarrow \min,$ $x_2^3 - x_1 + 1 = 0, \quad x_1 \geq 1, \quad x^0 = (2; 1)$
16	Закрытая транспортная задача	$3x_1^2 - 4x_2 \rightarrow \min,$ $2x_1 + x_2 - 4 = 0, \quad 37 - x_1^2 - x_2^2 \geq 0, \quad x^0 = (-1; 6)$
17	Транспортная задача с ограниченными пропускными способностями	$\lg(1 + x_1^2) - x_2 \rightarrow \min,$ $(1 + x_1^2)^2 + x_2^2 - 4 = 0, \quad x^0 = (1; 1)$
18	О кратчайшем пути	$0,0064x_1 \cdot [\exp(-0,184x_1^{0,3} x_2) - 1] \rightarrow \min,$ $1,2 \cdot 10^{13} - (3000 + x_1)x_1^2 x_2 \geq 0,$ $4,1 - \exp(0,184x_1^{0,3} x_2) \geq 0, \quad x^0 = (30000; 0,25)$

19	Планирования выпуска продукции при неопределенном спросе	$x_1 - x_2 + x_1^2 + x_2^2 - x_1 x_2 \rightarrow \min,$ $x_1 + x_2 = 1, x_1, x_2 \geq 0, x^0 = (0; 1)$
20	Планирования выпуска скоропортящейся продукции	$1/(x_1 x_2) + \sqrt{x_1 + x_2^{3/4}},$ $x_1, x_2 \geq 0,0001, x^0 = (1; 1)$
21	О планировании закупок топлива	$1/(x_1 x_2 x_3) \rightarrow \min,$ $2x_1 + x_2 + 3x_3 \leq 1,$ $x_1, x_2, x_3 \geq 0,0001, x^0 = (0,1; 0,1; 0,1)$
22	О народнохозяйственном плане	$5x_1^2 x_2^{-1} x_3 + 10x_1^{-3} x_2^2 x_3^{-2} \rightarrow \min,$ $0,357x_1^{-1} x_3 + 0,625x_1^{-1} x_3^{-1} \leq 1,$ $x_1, x_2, x_3 \geq 0,0001, x^0 = (0,1; 0,1; 0,1)$
23	Регулирования товарных запасов	$5x_1^2 - x_2^2 x_3^4 \rightarrow \max,$ $-5x_1^2 x_2^{-2} + 3x_2^{-1} x_3 \geq 2, x_1, x_2, x_3 \geq 0,0001, x^0 = (0,1; 0,1; 0,1)$

Таблица П.2 – Варианты матрицы стоимостей (для работ №№ 4, 5) 20

№	C11	C12	C13	C14	C21	C22	C23	C24	C31	C32	C33	C34	C41	C42	C43	C44
1	20	8	5	2	3	3	1	7	2	1	2	10	2	10	9	1
2	43	67	19	52	14	13	53	65	64	91	49	14	8	71	40	47
3	6	96	42	25	24	66	6	5	45	28	77	16	99	20	27	5
4	26	8	45	16	33	19	72	43	5	11	11	60	4	13	44	68
5	3	5	7	2	10	6	4	1	12	7	5	11	8	20	4	6
6	23	9	17	45	50	1	40	73	13	3	86	62	55	60	62	55
7	2	13	44	80	19	10	33	50	40	30	25	7	13	63	44	12
8	7	3	88	66	24	14	78	87	64	37	37	82	13	12	7	57
9	15	7	41	8	42	99	78	39	64	84	9	75	34	14	1	52
10	4	15	44	55	40	7	19	95	18	31	64	7	15	34	61	28
11	10	5	10	40	41	17	50	33	10	44	92	99	77	34	2	81
12	11	80	14	12	17	14	20	50	5	61	92	9	8	9	19	60
13	3	32	20	80	45	10	96	68	81	86	93	52	64	67	2	31
14	4	41	30	10	35	5	36	18	56	77	48	7	14	37	57	3
15	2	68	60	66	63	13	44	24	42	4	56	76	37	14	16	42

Таблица П.3 – Варианты объемов производства (a_i) и потребления (b_j) для транспортной задачи (работа № 5)

№	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
a1	10	34	15	10	35	14	12	28	56	34	5	21	18	55	52
a2	30	16	25	9	45	47	45	33	78	32	7	11	28	65	32
a3	20	23	13	35	27	38	13	27	29	67	37	31	38	56	33
a4	40	27	47	15	38	13	34	45	56	13	16	41	48	12	46
b1	15	10	23	23	34	56	11	10	70	28	23	12	84	37	72
b2	35	5	13	6	6	11	34	65	17	44	11	42	11	38	8
b3	25	50	16	13	25	43	23	9	45	11	9	12	30	59	12
b4	25	35	48	27	80	2	36	49	87	63	22	38	7	54	71

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Зайченко Ю.П. Исследование операций: Учеб. пособие для вузов.- Киев: Вища школа, 1975 (320с.), 1979 (392 с.), 1984(224 с.), 1988 (552 с.), 1990 (240 с.), 1991 (192 с.), рус.
2. Химмельблау Д. Прикладное нелинейное программирование / Д. Химмельблау. – М.: Мир, 1975. – 536 с. (75 экз. в электронной библиотеке каф. ТК СевНТУ)
3. Банди Б. Основы линейного программирования / Б. Банди. – М.: Радио и связь, 1989. – 176 с.
4. Курс лекций (файл LisslOp.doc в директории f:\polsov\common\isslop).
5. Справочная система пакета SIMPLEX (файл POLSOV/COMMON/Isslop/lpr1.hlp)

Заказ № 118 от « 5 » 03 2014 г. Тираж 25 экз

Изд-во СевНТУ