Министерство науки и образования Республики Казахстан

Талдыкорганский политехнический колледж

**Курсовая работа**

**По предмету:**

«Моделирование производственных и экономических процессов»

**На тему:**

«Решение задач нелинейного программирования»

г. Талдыкорган 2007 г.

**Введение**

Математическое программирование занимается изучение экстремальных задач и поиском методов их решения. Задачи математического программирования формулируются следующим образом: найти экстремум некоторой функции многих переменных f (x1, x2,…, xn) при ограничениях gi (x1, x2,…, xn) bi, где gi – функция, описывающая ограничения, а bi – действительное число, i = 1,…, m. Функция f называется функцией цели (целевой функцией).

В общем, виде задача нелинейного программирования состоит в определении максимального (минимального) значения функции f(x1, x2, …, xn) при условии, что ее переменные удовлетворяют соотношениям:



где f и g – некоторые известные функции n переменных, а bi – заданные числа.

В результате решения задачи будет определена точка Х\*= (x1\*, x2\*, …, xn\*), координаты которой удовлетворяют соотношениям и такая, что для всякой другой точки Х= (x1, x2, …, xn), удовлетворяющей условиям, выполняется неравенство f (x1\*, x2\*, …, xn\*) ≥ f (x1, x2, …, xn) [f (x1\*, x2\*, …, xn\*) ≥ f (x1, x2, …, xn)].

Если f и gi – линейные функции, то задача является задачей линейного программирования.

Соотношения образуют систему ограничений и включают в себя условия не отрицательности переменных, если такие условия имеются. Условия неотрицательности переменных могут быть заданы и непосредственно.

В евклидовом пространстве Еn система ограничений определяет область решений задачи. В отличие от задачи линейного программирования она не всегда является выпуклой.

Если определена область допустимых решений, то нахождение решения задачи сводится к определению такой точки этой области, через которую проходит гиперповерхность наивысшего (наименьшего) уровня: f (x1, x2, …, xn) = h. Указанная точка может находиться как на границе области допустимых решений, так и внутри неё.

Процесс нахождения решения задачи нелинейного программирования с использованием ее геометрической интерпретации включает следующие этапы:

1. Находят область допустимых решений задачи, определяемую соотношениями (если она пуста, то задача не имеет решения).
2. Строят гиперповерхность f (x1, x2, …, xn) = h.
3. Определяют гиперповерхность наивысшего (наинизшего) уровня или устанавливают неразрешимость задачи из-за неограниченности функций сверху (внизу) на множестве допустимых решений.
4. Находят точку области допустимых решений, через которую проходит гиперповерхности наивысшего (наинизшего) уровня, и определяют в ней значение функции.

Или приводят задачу нелинейного программирования к задаче линейного программирования и решают нижеизложенными способами.

Задача является задачей линейного программирования, а следовательно, ее решение можно найти известными методами: 1) графический; 2) табличный (прямой, простой) симплекс – метод; 3) метод искусственного базиса; 4) модифицированный симплекс – метод; 5) двойственный симплекс – метод.

**1. Табличный симплекс-метод**

Для его применения необходимо, чтобы знаки в ограничениях были вида «меньше либо равно», а компоненты вектора b – положительны.

Алгоритм решения сводится к следующему:

1. Приведение системы ограничений к каноническому виду путём введения дополнительных переменных для приведения неравенств к равенствам.

2. Если в исходной системе ограничений присутствовали знаки» равно "или" больше либо равно», то в указанные ограничения добавляются искусственные переменные, которые так же вводятся и в целевую функцию со знаками, определяемыми типом оптимума.

3. Формируется симплекс – таблица.

4. Рассчитываются симплекс – разности.

5. Принимается решение об окончании либо продолжении счёта.

6. При необходимости выполняются итерации.

7. На каждой итерации определяется вектор, вводимый в базис, и вектор, выводимый из базиса. Таблица пересчитывается по методу Жордана – Гаусса или каким-нибудь другим способом.

**2.** **Метод искусственного базиса**

Данный метод решения применяется при наличии в ограничении знаков «равно» больше либо равно» меньше либо равно и является модификацией табличного метода. Решение системы производится путём ввода искусственных переменных со знаком, зависящим от типа оптимума, т.е. для исключения из базиса этих переменных последние вводятся в целевую функцию с большими отрицательными коэффициентами, а в задачи минимизации – с положительными. Таким образом, из исходной задачи получается новая задача.

Если в оптимальном решении – задачи нет искусственных переменных, это решение есть оптимальное решение исходной задачи. Если же в оптимальном решении – задачи хоть одна из искусственных переменных будет отлична от нуля, то система ограничений исходной задачи несовместна и исходная задача неразрешима.

**3. Модифицированный симплекс-метод**

В основу данной разновидности симплекс-метода положены такие особенности линейной алгебры, которые позволяют в ходе решения задачи работать с частью матрицы ограничений. Иногда метод называют методом обратной матрицы.

В процессе работы алгоритма происходит спонтанное обращение матрицы ограничений по частям, соответствующим текущим базисным векторам. Указанная способность делает весьма привлекательной машинную реализацию вычислений вследствие экономии памяти под промежуточные переменные и значительного сокращения времени счёта. Способность хороша для ситуаций, когда число переменных n значительно превышает число ограничений m.

В целом, метод отражает традиционные черты общего подхода к решению задач линейного программирования, включающего в себя канонизацию условий задачи, расчёт симплекс – разностей, проверку условий оптимальности, принятие решений о коррекции базиса и исключение Жордана – Гаусса. Особенности заключаются в наличии двух таблиц – основной и вспомогательной, порядке их заполнения и некоторой специфичности расчётных формул.

Зная оптимальный план этой задачи, на основе соотношений получаем оптимальный план исходной задачи.

Таким образом, процесс нахождения решения задачи нелинейного программирования включает следующие этапы:

1. Первоначальную задачу сводят к задаче линейного программирования.
2. Находят решение линейной задачи

Используя соотношения, определяют оптимальный план исходной задачи и находят максимальное значение целевой функции нелинейной задачи.

**Первый этап: Получение задания к курсовой работе**

1. Все числовые данные, касающиеся предполагаемых производственных и экономических процессов, берутся на основе шестизначного шифра:

**9 5 5 8 7 2**

Под каждую цифру записываются буквы a, b, c, d, e, f в следующем виде:

**9 5 5 8 7 2**

**а b c d e f**

из последней строки таблицы индивидуальных заданий находим столбцы соответствующие буквам a, b, c, d, e, f. Тогда числовыми данными, необходимыми для выполнения данной курсовой работы, будут данные находящиеся в **а** – том столбце в строке 9, **b** – том столбце в строке 5, **c** – том столбце в строке 5, **d** – том столбце в строке **8**, **e** – том столбце в строке **7**и **f** – том столбце в строке 2.

По таблице исходных заданий для любого варианта заданий по столбцу **а** исполнитель получает вариант выполняемого задания. В моем случае для цифры 9 соответствует вариант 9.

На некотором заводе производится три вида продукта и при этом расходуется два вида ресурсов. Производственная функция каждого вида продукта на предприятии опишется равенствами:



где Сi и - постоянные величины, i = 1, 2, 3;



X1 – трудовые ресурсы в человеко-днях;

Х2 – денежно-материальные средства, в тенге;

Уi – получаемый продукт

Х1 = а1х1 + b1x2 + c1x3

Х2 = а2х1 + b2x2 + c2x3

Найти все неотрицательные базисные решения и определить оптимальный план F = y1 + y2 + y3.

Известно, что продукт для производства j – того вида затрачивается aij единиц i – того ресурса. Эти затраты даются в таблицах 3.9.1. – 3.9.10

Последующие числовые данные берутся только из таблицы исходных данных выбранного варианта задания т.е. из таблицы №3.9.11.

2. По столбцу таблицы №3.9.11 для строки 8 исходной таблицей затрат единиц ресурса, будет таблица №3.9.4 т.е. следующая таблица:

|  |  |  |  |
| --- | --- | --- | --- |
| **Продукты ресурсы** | **1** | **2** | **3** |
| I | 8 | 4 | 6 |
| II | 160 | 240 | 200 |

3. По столбцу **c –** на 3 строке находим с1=6, α1=0,6

4. По столбцу **d** – на 5 строке определяем с2=5, α2=0,5

5. По столбцу **e** – по 4 строке установим, что с3=8, α3=0,4.

6. И наконец по столбцу **f** – в 1 строке найдем Тчел.дней =1000, Птенге = 280000

Для производства имеются трудовые ресурсы Тчел.дней и денежно-материальные средства Птенге.

Требуется найти оптимальный план выпуска продукции, при котором выпускаемый продукт будет наибольшим.

**Второй этап – составление математической модели задачи**

1. На основании полученных в первом этапе исходных данных и описания заданного производственного процесса составляется следующая таблица:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Продукты ресурсы** | **1** | **2** | **3** |  |
| I | 8 | 4 | 6 | 1000 |
| II | 160 | 240 | 200 | 280000 |

Через **Х1** обозначим ресурсы I вида.

Через Х2 обозначим ресурсы II вида.

2. Обращаясь к условиям задачи, определяем все возможные ограничения, объединяя их в систему ограничений.

8Х1 + 4Х2 + 6Х3 ≤ 1000

240Х1+ 200Х2 + 160Х3 ≤ 280000

Таким образом, получили задачу нелинейного программирования. Такие задачи называются задачами нелинейного программирования.

Решение задач нелинейного программирования осуществляется приведением их к задачам линейного программирования.

Для решения задачи линейного программирования применяется симплекс – метод.

**Третий этап – выбор метода решения полученной математической задачи**

**Решение**

1. Для решения задач линейного программирования симплекс – методом задача приводиться к каноническому виду:

8Х1 + 4Х2 + 6Х3 + Х4= 1000

240Х1+ 200Х2 + 160Х3 + Х5= 280000

2. Составляем таблицу и определяем все неотрицательные базисные решения системы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Базисные переменные** | **Х1** | Х2 | Х3 | Х4 | Х5 | **Свободный член** |
| Х4 | 8 | 4 | 6 | 1 | 0 | 1000 |
| Х5 | 240 | 200 | 160 | 0 | 1 | 280000 |

А) Нашли некоторое неотрицательное базисное решение: Х4 =1300, Х5 = 190000. По заданию продолжаем искать базисные решения. Разрешающим элементом выбираем в 1 строке – **Х2**. Соответственно вся строка делится на 8, а все остальные элементы находятся по правилу прямоугольника.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Базисные переменные** | **Х1** | Х2 | Х3 | Х4 | Х5 | **Свободный член** |
| Х4 | 8 | 4 | 6 | 1 | 0 | 1000 |
| Х5 | 240 | 200 | 160 | 0 | 1 | 280000 |
| **Базисные переменные** | **Х1** | **Х2** | **Х3** | **Х4** | **Х5** | **Свободный член** |
| Х2 | ¾ | 1 | ½ | 1/8 | 0 | 325/2 |
| Х5 | 90 | 0 | 60 | -25 | 1 | 157500 |

Б) Нашли некоторое неотрицательное базисное решение: Х2 =325/2, Х5 =157500. По заданию продолжаем искать базисные решения. Разрешающим элементом выбираем в 1 строке – **Х1.** Соответственно вся строка делится на 3/4, а все остальные элементы находятся по правилу прямоугольника.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Базисные переменные** | **Х1** | Х2 | Х3 | Х4 | Х5 | **Свободный член** |
| Х2 | ¾ | 1 | ½ | 1/8 | 0 | 325/2 |
| Х5 | 90 | 0 | 60 | -25 | 1 | 157500 |
| **Базисные переменные** | **Х1** | Х2 | Х3 | Х4 | Х5 | **Свободный член** |
| Х1 | 1 | 4/3 | 2/3 | 1/6 | 0 | 650/3 |
| Х5 | 0 | -120 | 0 | -40 | 1 | 138000 |

В) Нашли некоторое неотрицательное базисное решение: Х1 =650/3, Х5 =138000. По заданию продолжаем искать базисные решения. Разрешающим элементом выбираем в 1 строке – **Х3.** Соответственно вся строка делится на 2/3, а все остальные элементы находятся по правилу прямоугольника.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Базисные переменные** | **Х1** | Х2 | Х3 | Х4 | Х5 | **Свободный член** |
| Х1 | 1 | 4/3 | 2/3 | 1/6 | 0 | 650/3 |
| Х5 | 0 | -120 | 0 | -40 | 1 | 138000 |
| **Базисные переменные** | **Х1** | Х2 | Х3 | Х4 | Х5 | **Свободный член** |
| Х3 | 3/2 | 2 | 1 | 1/4 | 0 | 325 |
| Х5 | 0 | -120 | 0 | -40 | 1 | 138000 |

Г) Нашли некоторое неотрицательное базисное решение: Х5 =138000, Х3 =325. Найдены все неотрицательные базисные решения.

2. Находим получаемый продукт.

Х1= 6\*0+8\*0+4\*0=0

Х2=240\*0+200\*0+160\*0=0

У1=3\*00,4\*00,6=0

У2=5\*00,5\*00,5=0

У3=8\*00,6\*00,4=0

F1=0+0+0=0

Х1= 6\*0+8\*325/2+4\*0=1300

Х2=240\*0+200\*325/2+160\*0=32500

У1=3\*13000,4\*325000,6=26904,728

У2=5\*13000,5\*325000,5=32500

У3=8\*13000,6\*325000,4=37688,542

F2=26904,728 +32500 +37688,542 = 97093,27

Х1= 6\*650/3+8\*0+4\*0=1300

Х2=240\*650/3+200\*0+160\*0=52000

У1=3\*13000,4\*520000,6=35699,794

У2=5\*13000,5\*520000,5=41109,610

У3=8\*13000,6\*520000,4=45483,862

F3= 35699,794+ 41109,610+ 45483,862= 122263,266

Х1= 6\*0+8\*0+4\*325=1300

Х2=240\*0+200\*0+160\*325=52000

У1=3\*13000,4\*520000,6=35699,794

У2=5\*13000,5\*520000,5=41109,610

У3=8\*13000,6\*520000,4=45483,862

F3= 35699,794+ 41109,610+ 45483,862= 122263,266

F1 < F2

F2 < F3

F3 = F4

Ответ: Fmax= 122263,266

# Четвертый этап – подготовка словесного алгоритма решения задачи

1. Вводим данные в таблицу
2. Выбираем разрешающий элемент:

2.1. Берем каждый неотрицательный элемент первой строки и делим на свободный член первой строки.

2.2. Находим среди всех деленных элементов минимальный.

2.3. Берем каждый неотрицательный элемент второй строки и делим на свободный член второй строки.

2.4. Находим среди всех деленных элементов минимальный.

2.5. Берем каждый неотрицательный элемент n-ой строки и делим на свободный член n-ой строки.

2.6. Находим среди всех деленных элементов минимальный.

2.7. Берем минимальные элементы первой, второй и n-ой строки и среди них находим минимальный (это и будет разрешающий элемент). При условии если минимальные элементы строк совпадают, берется элемент первой строки.

3. Вычисляем всю таблицу методом прямоугольника относительно разрешающего элемента:

1. Умножаем разрешающий элемент на элемент решаемой строки.
2. Отнимаем произведение соответствующего элемента решаемой строки на элемент разрешающего столбца решаемой строки
3. И делим ответ на разрешающий элемент.
4. Делим разрешающую строку на разрешающий элемент.
5. Берем каждый элемент разрешающей строки и делим на разрешающий элемент.
6. Всем элементам, кроме разрешающего элемента, разрешающего столбца присвоим (0)
7. Разрешающему элементу присвоим (1).

В индексе разрешающей строки присвоить индекс

4. Повторяем процедуру вычисления с 2 пункта.

5. В конечном результате находим все неотрицательные базисные решения. Подставляем значения и находим получаемый продукт.

6. Находим все F.

7. Выбираем наибольшую из них, которая будет являться оптимальным планом выпуска продукции.

# Пятый этап – разработка программы для решения задачи

Private Sub Form\_Load()

Left = (Screen. Width – Width) \ 2

Top = (Screen. Height – Height) \ 2

End Sub

Private Sub Timer1\_Timer()

Unload Form1

Load Form2

Form2. Show

End Sub

‘Объявление переменных

Public a As Integer

Public b As Integer

Public c As Integer

Public d As Integer

Public e As Integer

Public f As Integer

Public aa As Integer

Public ab As Integer

Public ac As Integer

Public ad As Integer

Public ae As Integer

Public af As Integer

Public ba As Integer

Public bb As Integer

Public bc As Integer

Public bd As Integer

Public be As Integer

be = Text17. Text

bf = Text18. Text

ca = Text19. Text

cb = Text20. Text

cc = Text21. Text

cd = Text22. Text

ce = Text23. Text

cf = Text24. Text

X1 = Text25. Text

X2 = Text26. Text

X3 = Text27. Text

‘Проверка выполнения равенств

If a\*x1+aa\*x2+ba\*x3=ca Then «Равенство выполняется» Else «Равенство не выполняется»

If b\*x1+ab\*x2+bb\*x3=cb Then «Равенство выполняется» Else «Равенство не выполняется»

If c\*x1+ac\*x2+bc\*x3=cc Then «Равенство выполняется» Else «Равенство не выполняется»

If d\*x1+ad\*x2+bd\*x3=cd Then «Равенство выполняется» Else «Равенство не выполняется»

If e\*x1+ae\*x2+be\*x3=ce Then «Равенство выполняется» Else «Равенство не выполняется»

F= f\*x1+af\*x2+bf\*x3

If F<fmin Then «Решение не выполняется» Else «Решение выполняется, план является оптимальным»

Text28. Visible = True

Text29. Visible = True

End Sub

Private Sub Command2\_Click()

‘очистка текстовых окон для следующего ввода данных

Text1. Text = «»

Text2. Text = «»

Text3. Text = «»

Text4. Text = «»

Text5. Text = «»

Text6. Text = «»

Text7. Text = «»

Text8. Text = «»

Text9. Text = «»

Text10. Text = «»

Text11. Text = «»

Text12. Text = «»

Text13. Text = «»

Text14. Text = «»

Text15. Text = «»

Text16. Text = «»

Text17. Text = «»

Text18. Text = «»

Text19. Text = «»

Text20. Text = «»

## Text21. Text = «»

Text22. Text = «»

Text23. Text = «»

Text24. Text = «»

Text25. Text = «»

Text26. Text = «»

Text27. Text = «»

Text28. Visible = False

Text29. Visible = False

End Sub

Private Sub Command3\_Click()

‘показать справку

Unload Form2

Load Form3

Form3. Show

End Sub

Private Sub Command4\_Click()

Unload Form2

End Sub

Private Sub Form\_Load()

Left = (Screen. Width – Width) \ 2

Top = (Screen. Height – Height) \ 2

‘подготовка текстовых окон к вводу данных при запуске рабочего окна

Text1. Text = «»

Text2. Text = «»

Text3. Text = «»

Text4. Text = «»

Text5. Text = «»

Text6. Text = «»

Text7. Text = «»

Text8. Text = «»

Text9. Text = «»

Text10. Text = «»

Text11. Text = «»

Text12. Text = «»

Text13. Text = «»

Text14. Text = «»

Text15. Text = «»

Text16. Text = «»

Text17. Text = «»

Text18. Text = «»

Text19. Text = «»

Text20. Text = «»

Text21. Text = «»

Text22. Text = «»

Text23. Text = «»

Text24. Text = «»

Text25. Text = «»

Text26. Text = «»

Text27. Text = «»

End Sub

Private Sub Form\_Load()

Left = (Screen. Width – Width) \ 2

Top = (Screen. Height – Height) \ 2

End Sub

Private Sub Timer1\_Timer()

Unload Form3

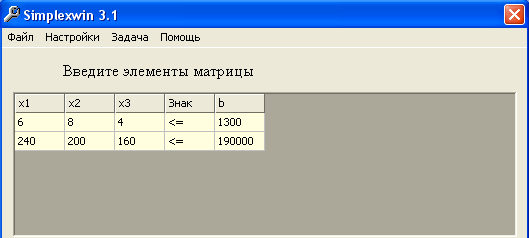
Load Form2

Form2. Show

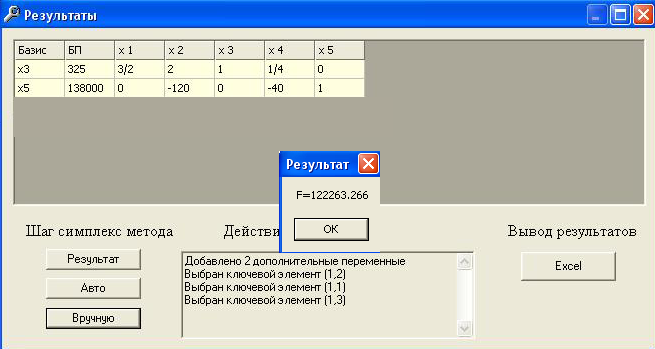
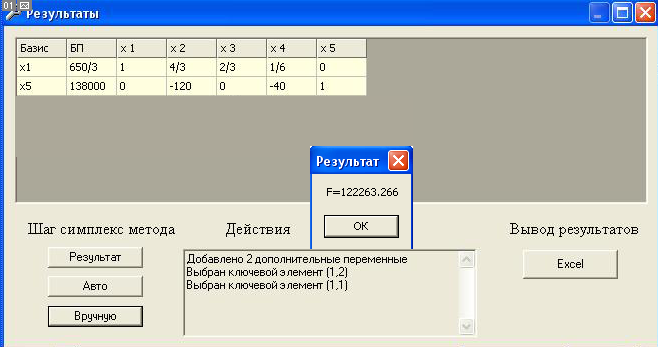
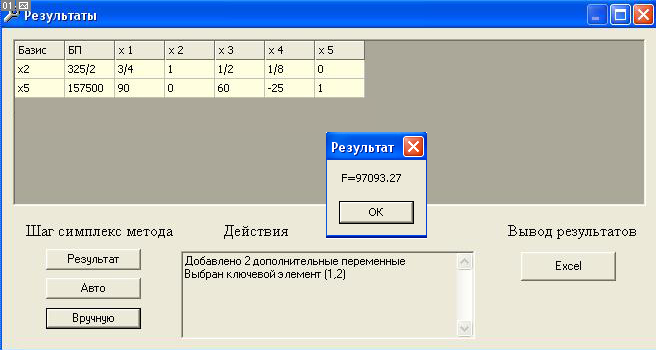
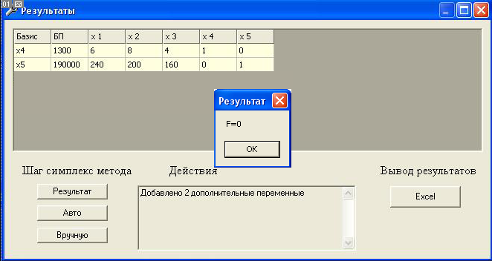
End Sub

**Результат использования программы**

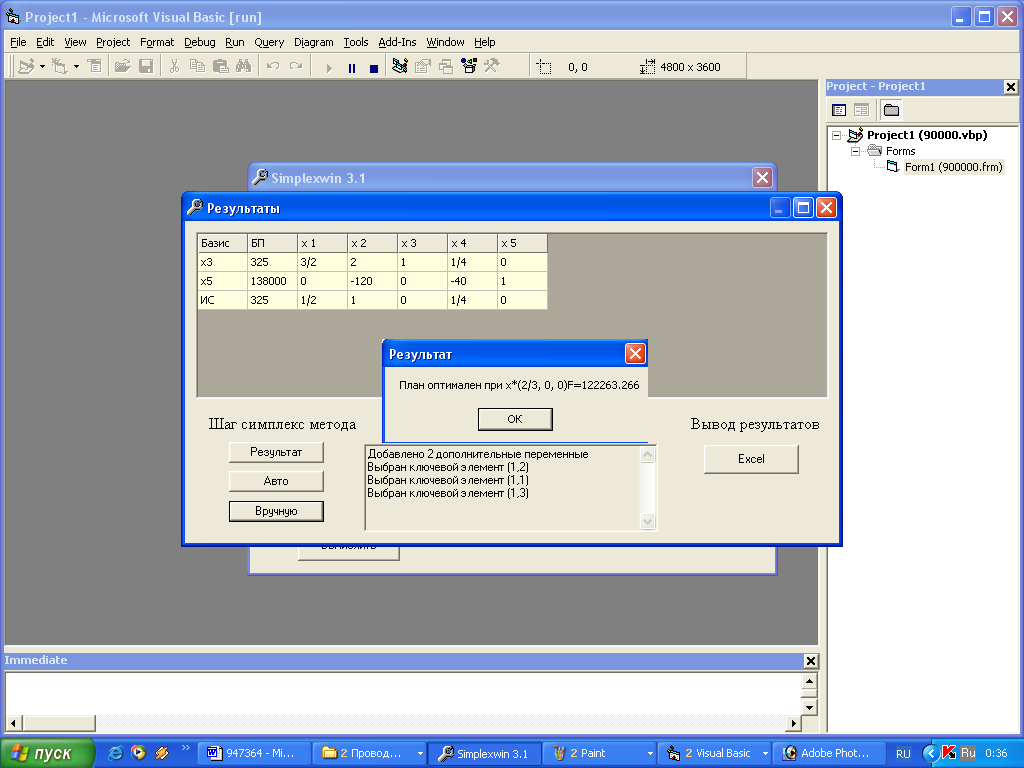
Ввод начальных коэффициентов



**Полученное решение**



**Конечный результат**



**Список используемой литературы**

1. Методические рекомендации «Курсовая работа по моделированию производственных и экономических процессов» Талдыкорган. 1999 г.
2. Уолш Б. «Программирование на Бейсике» Пер. с анг. – Москва: Радио и связь, 1998 г.
3. Фиакко А., Маккормик Г. «Нелинейное программирование» Пер. С анг. – Москва: Мир, 1988 г.
4. Солодовников А.С. «Введение в линейную алгебру и линейное программирование» Москва, «Просвещение», 1996 г.
5. Кузнецов Ю.Н. и др. «Математическое программирование» Москва, «Высшая школа», 1980 г.

ЛИСТ

7

КР– КФЗ– 031302– 955872

ЛИСТ

8

ЛИСТ

9

КР– КФЗ– 031302– 955872

ЛИСТ

11

КР– КФЗ– 031302– 955872

ЛИСТ

12

КР– КФЗ– 031302– 955872

ЛИСТ

13

КР– КФЗ– 031302– 434– 09 - 923926

ЛИСТ

13

КР– КФЗ– 031302– 955872

ЛИСТ

14

КР– КФЗ– 031302– 955872

ЛИСТ

15

КР– КФЗ– 031302– 955872

ЛИСТ

16

КР– КФЗ– 031302– 955872

ЛИСТ

17

КР– КФЗ– 031302– 955872

ЛИСТ

18

КР– КФЗ– 031302– 955872

ЛИСТ

19

КР– КФЗ– 031302– 955872

ЛИСТ

20

КР– КФЗ– 031302– 955872

ЛИСТ

21

КР– КФЗ– 031302– 955872

лист

22

КР– КФЗ– 031302– 955872

лист

23

КР– КФЗ– 031302– 943541

лист

КР– КФЗ– 031302– 434– 08 - 849472

лист

24

КР– КФЗ– 031302– 943541

лист

25

лист

26

КР– КФЗ– 031302– 434– 09 - 923926

ЛИСТ

28