**Аннотация**

Транспортная задача с правильным балансом состоит в том, чтобы найти оптимальный план по заданной таблице перевозок, при котором стоимость перевозок будет минимальна.

Такая задача актуальна в областях связанных с транспортировкой грузов.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| САРОВСКИЙ ГОСУДАРСТВЕННЫЙФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ |
|  |  |  |
| ЭКОНОМИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ |
|  |  |  |
| КАФЕДРА МАТЕМАТИЧЕСКИХ МЕТОДОВ ИИССЛЕДОВАНИЙ ОПЕРАЦИЙ В ЭКОНОМИКЕ |
|  |  |  |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА |
| К КУРСОВОЙ РАБОТЕ |
|  |
| на тему: |
|  | **Решение транспортной задачи с правильным балансом** |  |
|  |  |  |
| Студента |  |  |
| руководитель работы |  |  |
|  |  |  |
| консультанты работы |  |  |
|  |  |  |
|  |  |  |
| Зав. кафедрой |  |  |
|  |  |  |
|  | г. Саров2005 г |  |

**Оглавление**

Введение 3

Постановка задачи 4

Метод решения 5

Язык программирования 7

Описание алгоритма 8

Описание основных структур данных 12

Описание интерфейса с пользователем 14

Заключение 16

Литература 17

Текст программы 18

**Введение**

Задача состоит в том, чтобы наитии такой способ перевозки при котором затраты связанные с перевозкой были бы минимальны

Эта задача заинтересовала меня потому, что её решение напрямую связанно с экономией средств. Важно нахождение такого алгоритма, который позволит наиболее оптимально решить задачу.

# Постановка задачи

Транспортная задача ставится следующим образом: имеется m пунктов отправления А1, А2 , ..., Аm , в которых сосредоточены запасы каких-то однородных грузов в количестве соответственно а1, а2, ... , аm. Имеется n пунктов назначения В1 , В2 , ... , Вn подавшие заявки соответственно на b1 , b2 , ... , bn груза. Известны стоимости Сi,j перевозки от каждого пункта отправления Аi до каждого пункта назначения Вj . Все числа Сi,j, образующие прямоугольную таблицу заданы. Требуется составить такой план перевозок (откуда, куда и сколько поставить), чтобы все заявки были выполнены, а общая стоимость всех перевозок была минимальна.

# Метод решения

**1.Составление опорного плана.**

Решение транспортной задачи начинается с нахождения опорного плана. Для этого существуют различные способы. Например, способ “северо-западного угла” Рассмотрим конкретный примере:

Условия транспортной задачи заданы транспортной таблицей.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  ПН ПО  | В1  | В2 | В3 | В4 | В5 | Запасыаi |
|  А1 | 10 | 8 | 5 | 6 | 9 | 48 |
|  А2 | 6 | 7 | 8 | 6 | 5 | 30 |
|  А3 | 8 | 7 | 10 | 8 | 7 | 27 |
|  А4 | 7 | 5 | 4 | 6 | 8 | 20 |
|  Заявки  bj | 18 | 27 | 42 | 12 | 26 | 125 |

Будем заполнять таблицу перевозками постепенно начиная с левой верхней ячейки (“северо-западного угла“ таблицы). Будем рассуждать при этом следующим образом. Пункт В1 подал заявку на 18 единиц груза. Удовлетворим эту заявку за счёт запаса 48, имеющегося в пункте А1 , и запишем перевозку 18 в клетке (1,1). После этого заявка пункта В1 удовлетворена , а в пункте А1 осталось ещё 30 единиц груза. Удовлетворим за счёт них заявку пункта В2 (27 единиц), запишем 27 в клетке (1,2); оставшиеся 3 единицы пункта А1 назначим пункту В3. В составе заявки пункта В3 остались неудовлетворёнными 39 единиц. Из них 30 покроем за счёт пункта А2, чем его запас будет исчерпан, и ещё 9 возьмём из пункта А3. Из оставшихся 18 единиц пункта А3 12 выделим пункту В4; оставшиеся 6 единиц назначим пункту В5, что вместе со всеми 20 единицами пункта А4 покроет его заявку. На этом распределение запасов закончено; каждый пункт назначения получил груз согласно своей заявки. Это выражается в том, что сумма перевозок в каждой строке равна соответствующему запасу, а в столбце — заявке. Таким образом, нами сразу же составлен план перевозок, удовлетворяющий балансовым условиям. Полученное решение является опорным решением транспортной задачи:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  ПН ПО  | В1  | В2 | В3 | В4 | В5 | Запасыаi |
|  А1 | 10 18 | 8 27 | 5 3 | 6 | 9 | 48 |
|  А2 | 6 | 7 | 8 30 | 6 | 5 | 30 |
|  А3 | 8 | 7 | 10 9 | 8 12 | 7 6 | 27 |
|  А4 | 7 | 5 | 4 | 6 | 8 20 | 20 |
|  Заявки  bj | 18 | 27 | 42 | 12 | 26 | 125 |

Составленный нами план перевозок, не является оптимальным по стоимости, так как при его построении мы совсем не учитывали стоимость перевозок Сi,j .

**2.Распределительный метод достижения оптимального плана**

Теперь попробуем улучшить план, составленный способом “северо-западного угла”. Перенесем, например, 18 единиц из клетки (1,1) в клетку (2,1) и чтобы не нарушить баланса перенесём те же 18 единиц из клетки (2,3) в клетку (1,3). Получим новый план. Подсчитав стоимость опорного плана (она ровняется 1039) и стоимость нового плана (она ровняется 913) нетрудно убедиться что стоимость нового плана на 126 единиц меньше. Таким образом за счёт циклической перестановки 18 единиц груза из одних клеток в другие нам удалось понизить стоимость плана:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  ПН ПО  | В1  | В2 | В3 | В4 | В5 | Запасыаi |
|  А1 | 10  | 8 27 | 5 21 | 6 | 9 | 48 |
|  А2 | 6  18 | 7 | 8 12 | 6 | 5 | 30 |
|  А3 | 8 | 7 | 10 9 | 8 12 | 7 6 | 27 |
|  А4 | 7 | 5 | 4 | 6 | 8 20 | 20 |
|  Заявки  bj | 18 | 27 | 42 | 12 | 26 | 125 |

На этом способе уменьшения стоимости в дальнейшем и будет основан алгоритм оптимизации плана перевозок. Циклом в транспортной задаче мы будем называть несколько занятых клеток, соединённых замкнутой ломанной линией, которая в каждой клетке совершает поворот на 90°.

Существует несколько вариантов цикла :

1.) 2.) 3.)

Нетрудно убедиться, что каждый цикл имеет чётное число вершин и значит, чётное число звеньев (стрелок). Условимся отмечать знаком “+” те вершины цикла, в которых перевозки необходимо увеличить, а знаком “-“ те вершины, в которых перевозки необходимо уменьшить. Цикл с отмеченными вершинами будем называть “означенным”. Перенести какое-то количество единиц груза по означенному циклу — это значит увеличить перевозки, стоящие в положительных вершинах цикла, на это количество единиц, а перевозки, стоящие в отрицательных вершинах уменьшить на то же количество. Очевидно, при переносе любого числа единиц по циклу равновесие между запасами и заявками не меняется: по прежнему сумма перевозок в каждой строке равна запасам этой строки, а сумма перевозок в каждом столбце — заявке этого столбца. Таким образом при любом циклическом переносе, оставляющем перевозки неотрицательными допустимый план остаётся допустимым. Стоимость же плана при этом может меняться: увеличиваться или уменьшатся. Назовём ценой цикла увеличение стоимости перевозок при перемещении одной единицы груза по означенному циклу. Очевидно цена цикла ровна алгебраической сумме стоимостей, стоящих в вершинах цикла, причём стоящие в положительных вершинах берутся со знаком “+”, а в отрицательных со знаком “-“. Вершины чередуются начиная с “+”. Обозначим цену цикла через γ. При перемещении одной единицы груза по циклу стоимость перевозок увеличивается на величину γ. При перемещении по нему k единиц груза стоимость перевозок увеличиться на kγ. Очевидно, для улучшения плана имеет смысл перемещать перевозки только по тем циклам, цена которых отрицательна. Каждый раз, когда нам удаётся совершить такое перемещение стоимость плана уменьшается на соответствующую величину kγ. Так как перевозки не могут быть отрицательными, мы будем пользоваться только такими циклами, отрицательные вершины которых лежат в базисных клетках таблицы, где стоят положительные перевозки. Если циклов с отрицательной ценой в таблице больше не осталось, это означает, что дальнейшее улучшение плана невозможно, то есть оптимальный план достигнут.

Метод последовательного улучшения плана перевозок и состоит в том, что в таблице отыскиваются циклы с отрицательной ценой, по ним перемещаются перевозки, и план улучшается до тех пор пока циклов с отрицательной ценой уже не останется. Можно доказать, что для любой свободной клетке транспортной таблице всегда существует цикл и притом единственный, одна из вершин которого лежит в этой свободной клетке, а все остальные в базисных клетках. Если цена такого цикла, с плюсом в свободной клетке, отрицательна, то план можно улучшить перемещением перевозок по данному циклу. Количество единиц груза k, которое можно переместить, определяется минимальным значением перевозок, стоящих в отрицательных вершинах цикла (если переместить большее число единиц груза, возникнут отрицательные перевозки).

Применённый выше метод отыскания оптимального решения транспортной задачи называется распределённым; он состоит в непосредственном отыскании свободных клеток с отрицательной ценой цикла и в перемещении перевозок по этому циклу.

# Язык программирования

Для написания программы был выбран язык VBA по следующим причинам:

1. Visual Basic for Applications был написан специально для Office приложений то есть потенциальному заказчику достаточно установит Office и запустить Ms Excel чтобы программы работала

2. Так как Транспортная задача заданна в виде таблицы то удобнее решать ее использую возможности табличного редактора

3 . Это один из языков, в котором я неплохо разбираюсь. Поэтому мне удобнее писать программу с помощью VBA.

# Описание алгоритма

1.При запуске программы пользователю предлагается ввести количество запасов и запросов

А) Выполняется проверка на правильность ввода. Если введены числа то

Б) Рисуется таблица

2. После прорисовки таблицы пользователю предлагается заполнить таблицу стоимости перевозок

А) После подтверждения заполнения пользователем происходи проверка на правильность ввода

Б) Суммируется строка Запросы и столбец Запасы выполняется проверка на равенство. Если равны то

3. Выполняем построение опорного плана методом Северо-западного угла

A) Cтроим нулевую матрицу заданного размера на “листе2”

Б) Затем методом северо-западного угла исходя из данных на листе “Данные”

В) Переносим получившуюся таблицу на лист “Циклы”

4.Используем метод распределения следующим образом

Из приведенного примера уже можно выделить, как алгоритм перебирает пути. Он действует по следующей схеме:

1. Начальное значение j = 1 (первое место в маршруте).
2. Мы находимся в городе k.
3. Для каждого города (i = от 1 до n)
4. Рассматриваем город i.
5. Если этот город еще не посещен,
	1. тогда переходим в город i; j увеличиваем на единицу. Добавляем номер города в маршрут на место j. Помечаем город как посещенный. Переходим к пункту 1 (k = i).
	2. иначе идти некуда, т.е. все города мы посетили.
		1. если j = количеству городов (n), т.е мы добрались до последнего пункта в маршруте и наш путь сформирован,
		тогда сравниваем длину пути с длиной минимального маршрута.
		2. Помечаем город как не посещенный и выходим из него. Уменьшаем j на единицу.
6. Берем следующий город (i=i+1).

**Описание основных структур данных**

 Теперь рассмотрим структуру приложения, опишем классы и процедуры, которые были изменены и наполнены кодом.

Программа состоит из 4 классов:

1. CAboutDlg связан со встроенным диалоговым окном «О программе».
2. CKurs\_LipinApp управляет запуском приложения и не связан с каким-либо диалоговым окном. [1]
3. CKurs\_LipinDlg связан с окном IDD\_KURS\_LIPIN\_DIALOG. Этот класс организует постановку и решение задачи.
4. CSetting связан с окном IDD\_DIALOG1. В окне вводятся параметры к задаче – расстояния между городами.

Класс CKurs\_LipinDlg.

В начале при вызове функции OnInitDialog() переменные заполняются начальными данными. Затем из файла «table.ini» считывается таблица расстояний между городами. И теперь диалоговое окно готово к работе с пользователем.

Функция OnPaint() выводит на экран карту, позволяет выделять города, выбранные пользователем, а также соединяет города линиями-путями, когда задача решена. Кроме того, обеспечивается вывод информации для пользователя: пояснения, длина минимального пути и список расстояний между городами, составляющие минимальный путь.

При нажатии левой кнопкой мыши вызывается функция
OnLButtonDown(). Она определяет по какому городу щелкнул пользователь и ставит/снимает выделение с него. Также здесь осуществляется проверка на количество выделенных городов, т.к. время ожидания решения задачи для количества более 13 городов станет не удовлетворительным (от 1,5 минут и более). Поэтому программа выдаст сообщение, если мы попытаемся выйти за допустимые пределы.

Кнопка «Выбрать стандартные города» выделяет города, которые требуется соединить в условии задачи, а именно Москва, Ярославль, Нижний Новгород, Самара, Саратов, Волгоград, Воронеж, Пенза, Липецк, Тамбов, Орел, Курск, Тула.

Кнопка «Очистить» снимает выделение со всех городов и задает начальные значения ряду необходимых переменных.

Функция OnButton3() связана с кнопкой «Задать пункт отправления». После нажатия кнопки пользователь может щелчком мыши выбрать пункт отправления коммивояжера.

Кнопка «Параметры» вызывает диалоговое окно «Параметры», где пользователь может редактировать таблицу расстояний между городами.

Функция OnOK() обрабатывает нажатие кнопки «Рассчитать путь». Она подготавливает начальные значения для вызова рекурсивной функции: создается таблица расстояний только для выделенных городов, заполняется первоначальный минимальный путь, выводится поясняющая информация. Затем вызывается функция recursiv(), которая перебирает варианты маршрутов, отсекает лишние ветви путей и находит минимальный.

Класс CSetting.

Функция OnInitDialog() программным путем создает и выводит поля ввода, имитируя таблицу расстояний между городами. Затем считывается файл «table.ini», и заполняются поля ввода полученными значениями. Вызывается функция Proverka(), которая сканирует полученные данные на ошибки, т.е. определяет введены ли только цифры, и если нет, тогда удаляет все лишние символы.

По нажатию кнопки «Сохранить» программа передает данные в родительское окно, записывает данные из полей ввода в файл и закрывает диалоговое окно.

# Описание интерфейса с пользователем

При загрузке приложения появляется основное диалоговое окно (рис. 3).

Здесь пользователь может выбрать несколько городов и рассчитать для них минимальный путь.

Чтобы отменить выделение городов нужно щелкнуть по кнопке «Очистить». Нажав кнопку «Рассчитать путь», мы получим результат: города соединены минимальным путем, его длина дана в окне информации, в списке показаны расстояния между городами, входящими в полученный путь (рис. 4).

Кнопка «Выбрать стандартные города» выделяет города, требуемые в задании.

Чтобы выделить пункт отправления коммивояжера нужно выбрать «Задать пункт отправления».

Кнопка «Параметры» вызывает диалоговое окно для ввода расстояний между городами (рис. 5). Это окно является модальным и его особенностью является то, что нет возможности перехода к родительскому окну.

Здесь пользователь может отредактировать расстояния между городами. Для этого нужно щелкнуть в поле ввода, и ввести другое значение. Перемещаться по этой «таблице» можно по строкам при помощи клавиш Tab или Shift+Tab.

По завершении ввода нужно нажать кнопку «Сохранить», чтобы программа записала измененные данные в файл. При этом автоматически проверится правильность введенный информации и все ошибки будут исправлены.

Кнопка «Отмена» позволяет не сохранять введенные изменения, если пользователь ошибся во введенной информации.

По нажатии любой из кнопок диалоговое окно «Параметры» закрывается и мы возвращаемся к главному окну .

Если в строке заголовка главного окна щелкнуть правой кнопкой мыши и выбрать пункт «О программе», то появится диалоговое окно, содержащее сведения о программе и об авторе (рис. 6). Нажав кнопку «OK» возвращаемся к главному окну.

**Заключение**

Был разработан алгоритм, позволяющий быстро найти кратчайший путь между городами. По алгоритму разработано приложение, решающее задачу о коммивояжере.

# Литература

1. Круглински Д., Программирование на Microsoft Visual C++ 6.0 для профессионалов/Пер.с англ. –СПб:Питер; 2004г. – 861 с.: ил.
2. Беляев С.П. Курс лекций по «Исследованию операций».