

Задача об оптимальном месте для магазина

На рис. 1 помещен протокол решения задачи о поиске места для магазина на дачном участке. Критерий поиска — минимум суммы расстояний от магазина до всех остальных домиков. Их координаты x и y — случайные числа в интервале от 0 до 100 (наша задача — учебная). Магазин может быть либо встроен в один из домиков, либо стоять отдельно.

ORIGIN := 1 n := 70 X := runif(n,0,100) Y := runif(n,0,100)

Встроенный магазин

$$i := 1..n \quad \text{Расстояние}_i := \sum_{j=1}^n \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$$

Мин.расст

i.Mar

Мин_расст := min(Расстояние) i_Mar := match(Мин_расст, Расстояние)_1

X_Mar := X_i_Mar Y_Mar := Y_i_Mar

Отдельно стоящий магазин

$$\text{Расстояние}(x,y) := \sum_{j=1}^n \sqrt{(x - X_j)^2 + (y - Y_j)^2}$$

Первое приближение

x := X_Mar y := Y_Mar $\begin{pmatrix} x \\ y \end{pmatrix} := \text{Minimize}(\text{Расстояние}, x, y)$

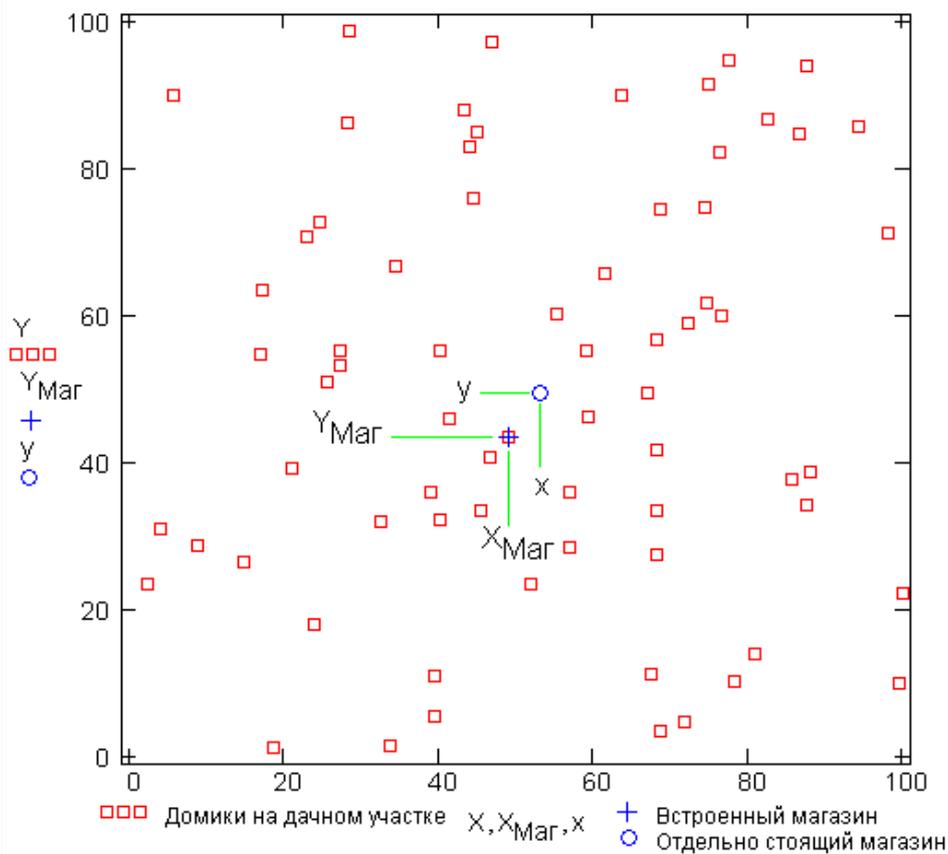
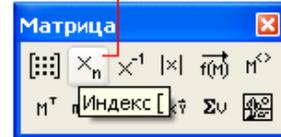


Рис. 1. Задача об оптимальном месте магазина в дачном поселке

На рис. 1 координаты встроенного магазина определяются перебором через функцию `match` (один из переводов с английского — "находить соответствие"). Затем эти координаты становятся первым приближением для уточнения местоположения отдельно стоящего магазина. Заканчивается расчет графической иллюстрацией — X-Y-графиком. Здесь главное — правильно отформатировать точки на графике: квадратики — дачи на участке, крестик — встроенный магазин и кружок — отдельно стоящий магазин.

Продолжением задачи о месте магазина среди дачных домиков является другая оптимизационная задача — *задача коммивояжера* (рис. 2).

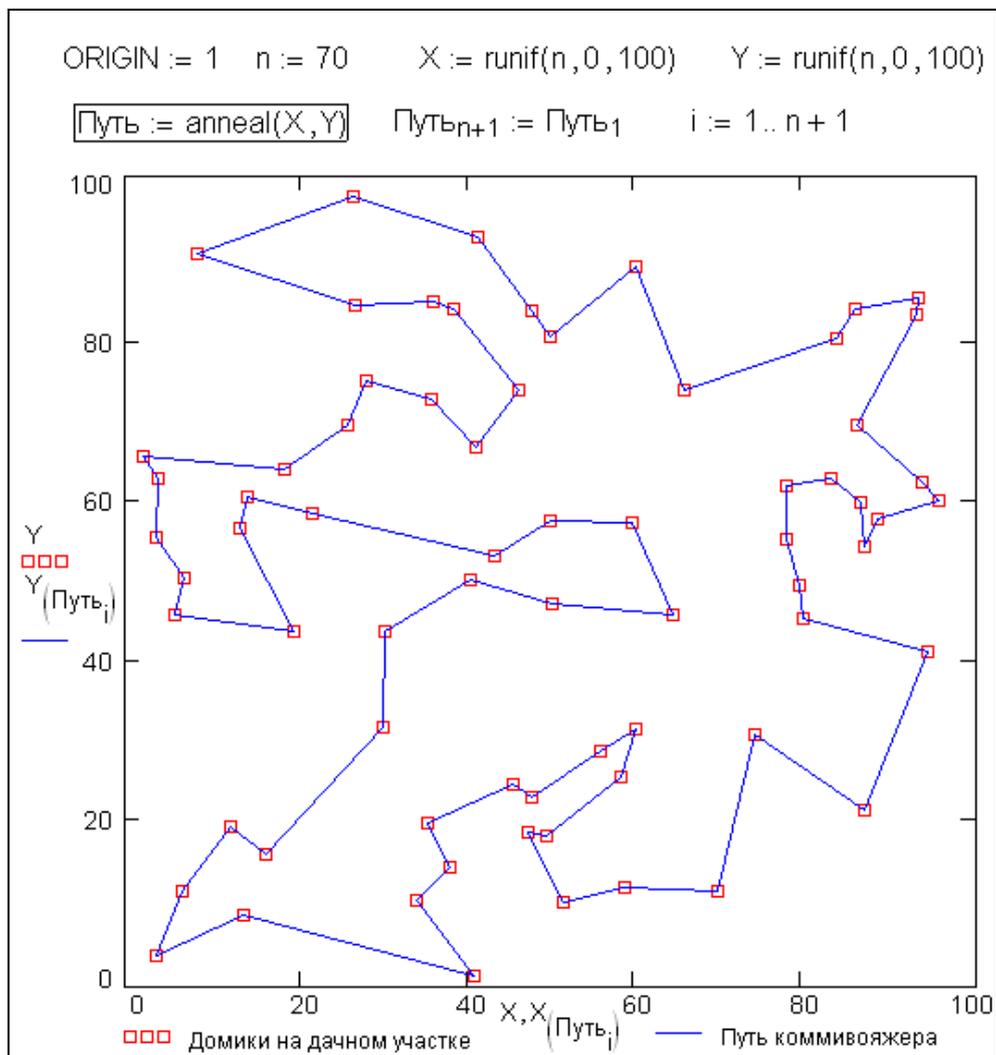


Рис. 2. Задача коммивояжера

Эта задача сводится к нахождению порядка обхода всех дачных домиков по маршруту наименьшей длины. На рис. 2 задача коммивояжера решается с помощью функции `anneal` (отжиг¹), которая становится встроенной (но не описанной в диалоговом окне ввода встроенных функций) после подгрузки к Mathcad расширения пакета Numerical

¹ Алгоритм основан на имитации физического процесса кристаллизации вещества из жидкого состояния в твердое, в том числе при *отжиге* металлов.

Recipes. Парадокс задачи коммивояжера в том, что от многих точек (домиков) путь лежит не к ближайшей точке, а чуть ли не к самой дальней.

Переход к ближайшей точке определяет так называемый "жадный" алгоритм, сущность которого мы уже слегка коснулись, описывая транспортную задачу (см. рис. 4.18): "жадная" полная нагрузка самого дешевого маршрута перевозки угля со второй шахты на первую электростанцию приводит к общим потерям — к максимизации, а не минимизации общих затрат на перевозку топлива. Тем не менее "жадные" алгоритмы имеют право на жизнь, и мы приложим их к задаче коммивояжера.

На рис. 3 показана программа, возвращающая вектор way (путь коммивояжера) в зависимости от векторов X и Y (координаты домиков) и номера первого домика ib , с которого начинается обход.

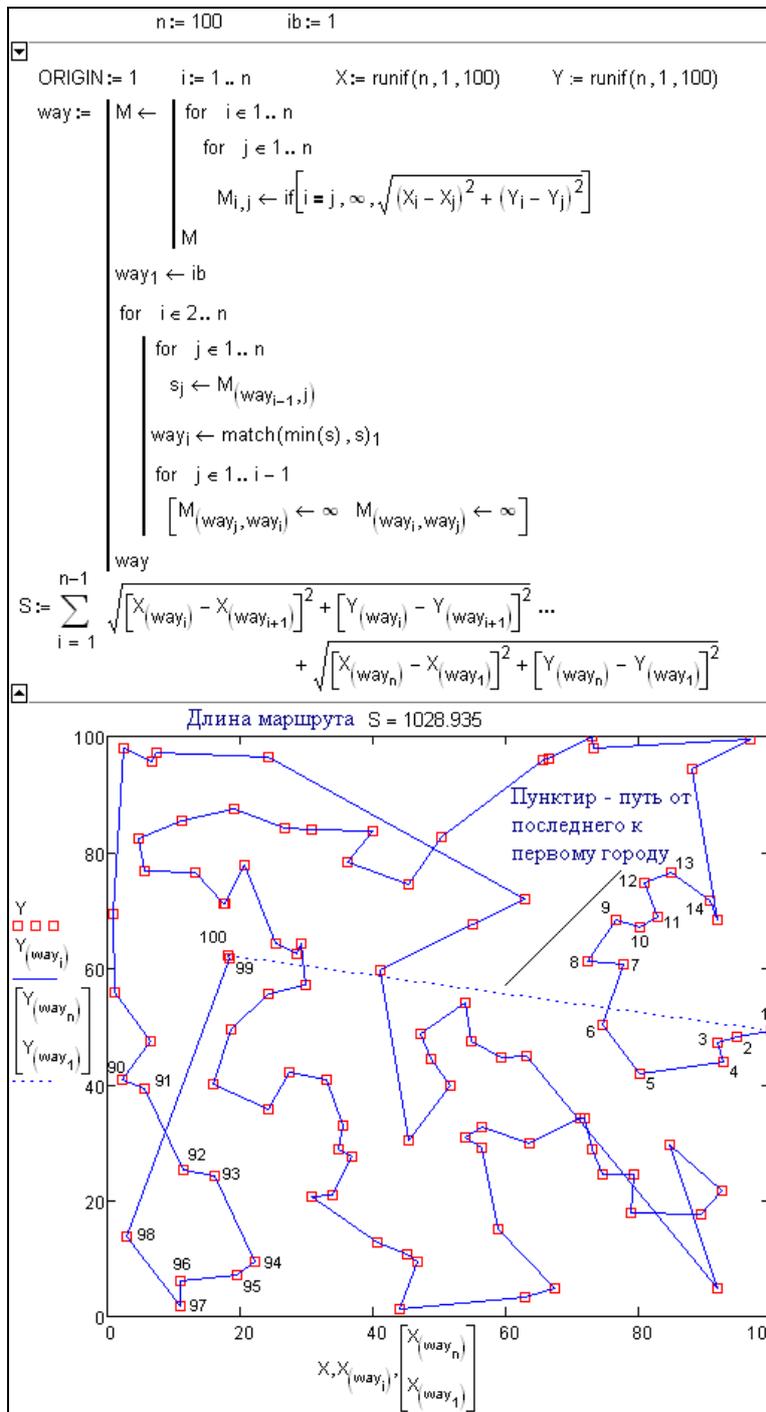


Рис. 3. "Жадный" алгоритм решения задачи коммивояжера

В программе сначала формируется квадратная матрица M , хранящая расстояния от точки i к точке j . Причем диагональ этой матрицы хранит значение числовой бесконечности $\infty = 10^{304}$. Это же значение будут получать элементы матрицы, когда будут выбираться данные пути (от i -й к j -й точке и от j -й к i -й точке) при заполнении вектора way . Вектор s в программе хранит расстояние от очередной точки ко всем остальным точкам. "Жадный" алгоритм сильно зависит от номера первой точки пути коммивояжера ib . Поэтому данный алгоритм можно развить: добавить в него перебор всех точек — дачных домиков и выполнить выбор того, из которого путь коммивояжера будет минимальным. На рис. 4 показана графическая иллюстрация такого перебора.

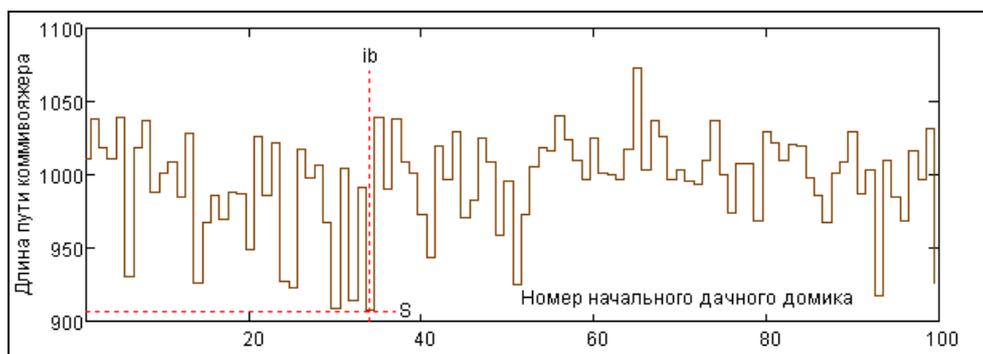


Рис. 4. График длины пути коммивояжера из разных дачных домиков при "жадном" алгоритме

На графике рис. 3 видно, что "жадный" алгоритм приводит к увеличению общего пути коммивояжера, в том числе и за счет *петель*. Так из 91-го домика² нужно было идти не в 92-й, а в 98-й. Предлагаю читателям самим доработать программу так, чтобы программа сама развязывала эти петли и... помогала коммивояжеру быстрее выполнить свою работу.

Программа, показанная на рис. 3, и ее самостоятельное развитие студентами — прекрасная лабораторная работа по информатике. Тем более, некоторые студенты подрабатывают... рассылными и начинают задумываться об оптимизации этой своей нелегкой побочной работы.

² Номера домиков на графике были проставлены вручную при доработке рисунка.