

Д. В. Крупа, Н. Б. Осипенко
(ГГУ им. Ф. Скорины, Гомель)
**РАЗРАБОТКА СРЕДСТВ АВТОМАТИЗАЦИИ
РЕШЕНИЯ ЗАДАЧИ О КОММИВОВАЖЕРЕ
НА ОСНОВЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ**

Генетические алгоритмы представляют собой алгоритмы поиска, построенные на принципах, сходных с принципами естественного отбора и генетики: выживания наиболее перспективных особей – решений и структурированный обмен информацией, в котором присутствует элемент случайности, который моделирует природные процессы наследования и мутации. Опосредованное вмешательство человека в развивающийся процесс поиска осуществляется через задание исходных параметров. Будучи разновидностью методов поиска с элементами случайности, генетические алгоритмы позволяют найти одно из лучших, но не оптимальное решение задачи.

Настоящая статья описывает разработанное в среде «Borland C++ Builder» программное приложение автоматизации решения задачи о коммивояжере на основе генетических алгоритмов.

Для решения задачи был применён следующий генетический алгоритм. Решение представлено в виде перестановки чисел от 1 до n (n – общее количество городов), отображающей последовательность посещения городов. Значение целевой функции будет равно сумме расстояний, вычисленной в соответствии с матрицей расстояний.

В основной части функционала реализовано: получение исходных данных, основные операторы генетического алгоритма (кроссовер и мутация), составление исходной популяции, проверка корректности ввода данных пользователем, расчёт кратчайшего пути. Также дополнительно были реализованы классы: Chromosome, хранящий в себе маршрут, методы получения маршрута и его длины, и ListChromosome, хранящий список маршрутов, методы добавления в список и удаления из него, получения количества маршрутов в списке, получения маршрута по номеру, сортировки списка по возрастанию длительности маршрута.

Исходными данными для программы являются: матрица расстояний между городами, количество поколений, уровень сходимости, вероятность мутации. Все эти данные предоставлены для ввода пользователю. «Количество поколений» – количество итераций пересчёта наилучшего пути в алгоритме. «Вероятность мутации» – вероятность, с которой будет происходить смена позиций городов в маршруте. «Уровень сходимости» определяет расстояние, минимум на которое должно улучшиться значение целевой функции. Если улучшения нет, то высокая похожесть особей привела к чрезвычайно медленному улучшению и процесс нужно остановить. «Матрица расстояний между городами» представляет собой двумерный массив. Этот массив заполняется автоматически, но не без участия пользователя. Пользователю предоставляется карта необходимой ему местности с населёнными пунктами. Кликая мышью по карте, пользователь заносит тем самым данную точку местности в массив TPoint tPoints[100], хранящий координаты всех точек местности, которые выбрал пользователь. В данном массиве хранятся расстояния между точками в пикселях. Но для того, чтобы получить реальное расстояние между населёнными пунктами, нужно перевести эти расстояния в стандартные метрические единицы. Проблема состоит в том, что у различных мониторов различный DPI – количество пикселей на дюйм. Для того, чтобы перевести расстояние в пикселях в одну из стандартных метрических единиц, можно воспользоваться функцией WIN API GetDeviceCaps(DC: HDC; Index: Integer): Integer. Она считывает с дисплея характерную для устройства информацию. Параметры: DC – идентификатор контекста устройства, Index – возвращаемый элемент, Возвращаемое значение – нужное значение элемента. При помощи этой функции можно получить размеры дисплея в миллиметрах и в пикселях. Затем, разделив размер в миллиметрах на размер в пикселях, узнаётся расстояние между пикселями в миллиметрах. На рисунке 1 приведен пример работы программы.

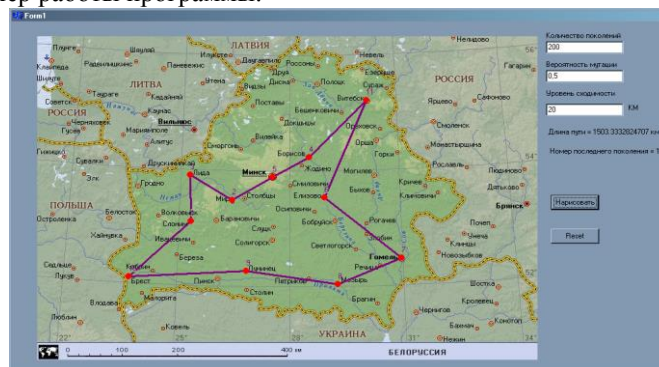


Рисунок 1 – Пример работы программы

РЕПОЗИТОРИЙ ГГУ ИМЕНИ Ф. СКОРИНЫ