Решение задачи коммивояжера Муравьиным алгоритмом.

Задача коммивояжера является одной из самых интересных, практически значимых и одновременно сложных задач теории графов. Задача, берущая свое начало из работ Гамильтона, состоит в определении кратчайшего гамильтонова цикла в графе. Ее решение связано с решением задачи о назначениях и с задачей об остове наименьшего веса. Задача заключается в отыскании самого выгодного маршрута, проходящего через указанные города (вершины) хотя бы по одному разу с последующим возвратом в исходный город (вершину).

Одним из методов искусственного интеллекта является муравьиный алгоритм Марко Дориго. Основная идея алгоритма подсмотрена в природе и имитирует движение колонии муравьев. По форме этот алгоритм похож на жадный Nva (Nearest vertex add) и в некоторой степени является его обобщением. Если в алгоритме ближайшего соседа выбор дальнейшего пути производится, исходя из минимального расстояния до очередной вершины, то здесь выбором управляет случайная функция, направляющая движение от текущего положения с большей вероятностью в вершину *j*, в которой наибольшее значение некоторой функции $P_{ij,k}$ (где i - номер вершины, в которой производится выбор, k - номер муравья, движущегося по дугам графа). Как и в Nva, во время движения создается список пройденных вершин, что позволяет избежать преждевременного зацикливания. Приведем вид функции, управляющей переходом из данной вершины *i* в вершину *j*:

$$P_{ij,k} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_m \tau_{im}^{\alpha}\eta_{im}^{\beta}} \ ,$$

где τ_{ij} количество феромона (pheromon), оставленного муравьями на дуге [*i*,*j*]; n_{ij} - величина, обратная весу (длине) дуги [*i*,*j*]; α, β - эмпирические коэффициенты. Функция $P_{ij,k}$ подсказывает муравью номер вершины *j*, в которую он должен направиться. В знаменателе стоит нормирующий коэффициент, такой, что $0 \leq P_{ij,k} \leq 1$. Индекс m в сумме пробегает по всем непройденным вершинам, смежным с *i*. В реальности муравей оставляет след (феромон) во время прохождения пути, и чем чаще он возвращается в исходную точку (а это возможно, если он выбирает оптимальные пути), тем четче след. В математической же модели функция τ_{ij} увеличивается только по завершении маршрута на величину, обратно пропорциональную длине маршрута. При $\alpha = 0$ алгоритм совпадает с Nva - муравей руководствуется только длиной пути. При $\beta = 0$ основой для выбора пути является только опыт (количество феромона, или «глубина следа») предыдущих муравьев-исследователей. Важно отметить еще одно отличие от алгоритма Nva. Выбор пути производится не по максимуму функции $P_{ij,k}$, а случайным образом, но на случай, конечно, влияет значение $P_{ij,k}$. Поясним это на примере. Пусть муравей *k* подошел к некоторой вершине 8 и обнаружил, что перед ним 7 возможных путей к семи вершинам (на уже пройденные он внимания не обращает). Куда идти? Муравей доверяется случаю. Он «пускает рулетку» (рис. 1). В какой сектор «шарик»

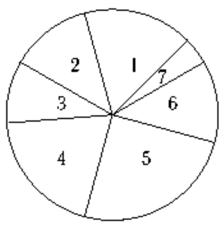


Рис. 1

закатится, туда и идти. Однако рулетка, размеченная функцией $P_{8j,k}$, $j = 1, \dots 7$, имеет неравные сектора. Чем ближе вершина и чем глубже туда след, тем больше сектор. Таким образом, муравей использует и опыт предшественников (τ_{ij}), и здравый смысл (n_{ij}), и случайный фактор, т.е. все как в жизни.

Для того чтобы не пропустить оптимальное решение, в муравьином алгоритме предусмотрено «испарение» следа. Это достигается введением коэффициента p в итеративной формуле $\tau_{ij} = (1 - p)\tau_{ij}$, применяющейся после каждого цикла обхода графа.

В алгоритме действует целая колония муравьев. Математически это означает, что в каждом цикле обхода движение производится из разных вершин независимым образом. Здесь приведен самый простой вариант алгоритма. Алгоритм может быть улучшен. Для ускорения сходимости иногда вводят так называемых *элитных* муравьев.

Рассмотрим вариант программной реализации алгоритма, описанного выше. Данная программа была запрограммирована в среде разработки “BORLAND DELPHI”, версия 7.0.

Решим задачу коммивояжера для 5 городов (Рис. 2).

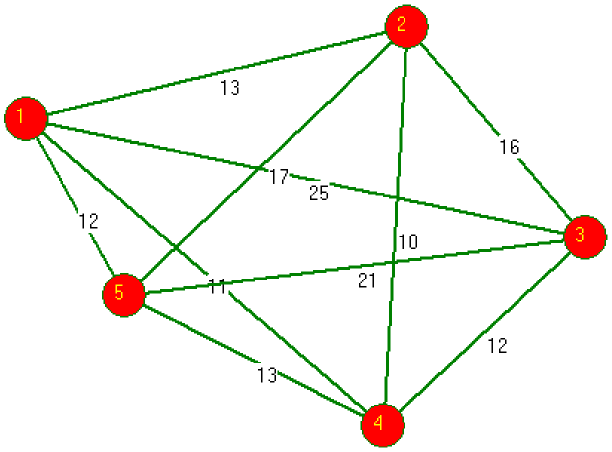


Рис. 2

	1	2	3	4	5
1	0	13	25	11	12
2	13	0	16	10	17
3	25	16	0	12	21
4	11	10	12	0	13
5	12	17	21	13	0

Рис. 3

В начале программы вводится матрица расстояний (Рис. 3). Затем вводятся константы задачи (Рис. 4) Эмпирические константы $a = \alpha, b = \beta$ и $0 < p < 1$ выбираются произвольно. Улучшение сходимости во многом зависит от их значений. Масштабная константа `Q` порядка длины маршрута выбирается пропорциональной порядку графа. При вычислении расстояний учитывается симметрия матрицы.

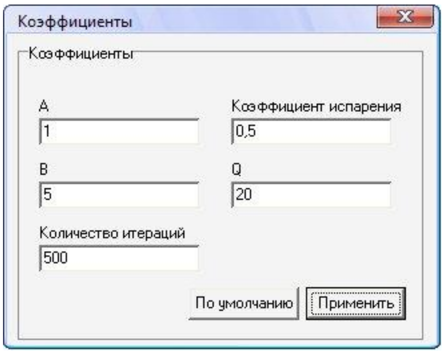


Рис. 4

В цикле, задающем время жизни колонии (для примера - 500 циклов), n муравьев размещаются по вершинам графа, и начинается движение. Первоначально феромон распределен по дугам равномерно, затем, после завершения муравьями своих маршрутов, уровень феромона увеличивается там, где муравьи ходили чаще.

Результат отображается графически. На Рис.5 представлен самый короткий цикл - `Lmin=66` - при $a = 1$, $b = 5$, $p = 0{,}5$.

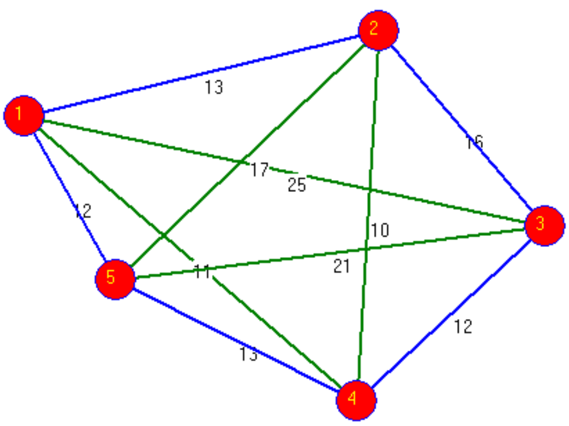
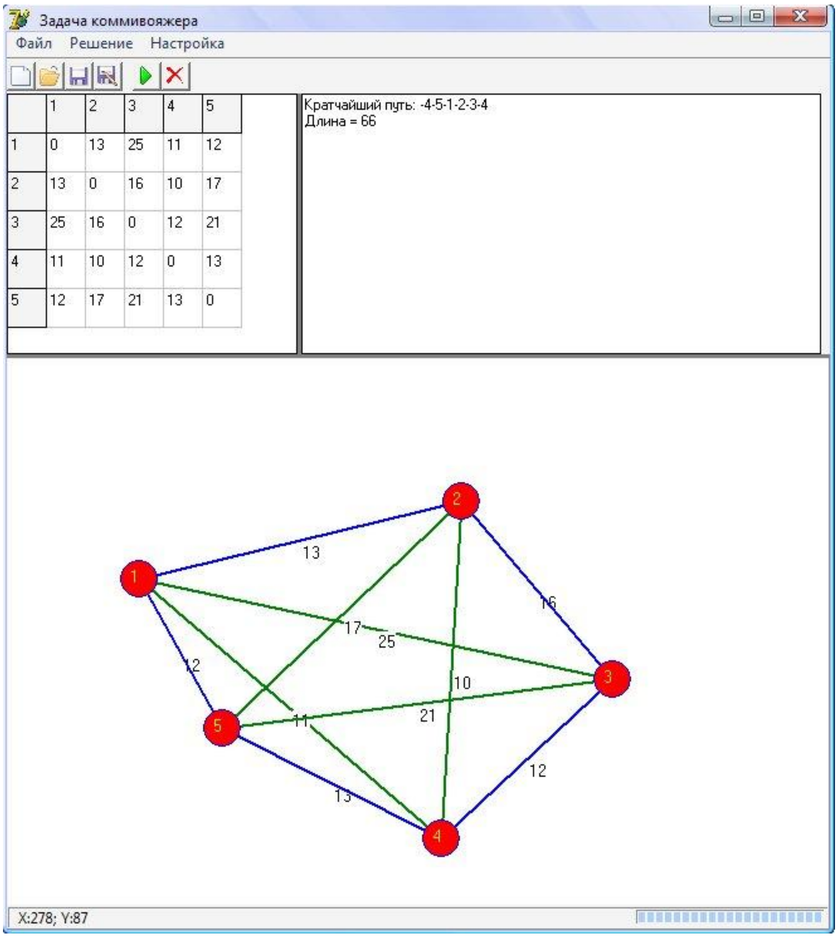


Рис. 5

Общий вид программы представлен на Рис. 6.



	1	2	3	4	5
1	0	13	25	11	12
2	13	0	16	10	17
3	25	16	0	12	21
4	11	10	12	0	13
5	12	17	21	13	0

Рис. 6