УДК 004.023

ПРИМЕНЕНИЕ МУРАВЬИНЫХ И ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА С ОГРАНИЧЕНИЯМИ НА НАПРАВЛЕННОСТЬ МАРШРУТА

Семенюта Е.В., Привалов М. В.
Донецкий национальный технический университет
кафедра автоматизированных систем управления
uspex-uspet@yandex.ru, evilmax@gmail.com

Формализована задача коммивояжера с ограничениями на направленность маршрута. Предложена гибридная муравьиная система, состоящая из муравьиного алгоритма и локального поиска. Подробно рассмотрен муравьиный алгоритм. В качестве локального поиска предложен генетический алгоритм, для которого выбрано путевое представление пути, кроссовер порядка (ОХ) и модификация классической мутации - сальтация. Представлены результаты работы описанной гибридной муравьиной системы.

4

Введение

Муравьиные алгоритмы сравнительно эффективны для решения задачи коммивояжера, так как для небольшого количества узлов она может быть решена полным перебором, а для большого количества узлов у нее экспоненциальное время сходимости из-за ее вычислительной сложности (NP-сложная). Генетический алгоритм позволяет решать данную задачу не менее успешно.

Решение задачи коммивояжера с ограничениями на направленность маршрута алгоритмом муравьиной колонии и генетическим алгоритмом позволило выявить ряд недостатков в том и в другом методах, что послужило поводом к изобретению гибридной муравьиной системы, включающей в себя муравьиный алгоритм и, в качестве локального поиска, генетический алгоритм.

1 Задача коммивояжера

Задача коммивояжера является широко известной задачей оптимизации комбинаторного типа. Она формулируется следующим образом: дан полный взвешенный граф $G(X,V)$ порядка n, где $X=\{x_1,...,x_n\}$ – множество вершин; $V\subseteq X\times X$ – множество ребер, в нем необходимо найти Гамильтонов цикл, имеющий наименьший суммарный вес входящих в него ребер [1].

Или формально:

$$\begin{cases} Q(x)=\sum_{i=1}^{N}\sum_{j=1}^{N}c_{ij}x_{ij}\to\min \\ \sum_{i=1}^{N}x_{ij}=1,\forall j=\overline{1,N} \\ \sum_{j=1}^{N}x_{ij}=1,\forall i=\overline{1,N} \\ x_{ij}\in\{0,1\} \end{cases} \tag{1}$$

где c_{ij} - вес ребра (i,j)

$$x_{ij}=\begin{cases} 1, \textit{в цикле есть переход из } i \textit{ в } j \\ 0, \textit{перехода из } i \textit{ в } j \textit{ нет} \end{cases} \tag{2}$$

Цикл в графе называется Гамильтоновым циклом, если он содержит все его ребра, причем каждое ребро один и только один раз. Очевидно, что решением задачи является перестановка из n вершин, количество возможных перестановок равно n!, однако количество различных решений задачи с учетом направления обхода и сдвига начальной вершины будет $\frac{(n-1)!}{2}$ [1]. Задача не имеет алгоритма, позволяющего найти решение за приемлемое время, и решается в основном эвристическими методами.

2 Учет ограничений на порядок обхода городов

2.1 Учет ограничений в генетическом алгоритме

После получения новой хромосомы в результате операторов кроссинговера и мутации ее необходимо проверить на выполнение ограничений, наложенных на порядок обхода городов.

Поэтому если в хромосоме $h = (h_1,...,h_n)$, $\forall h_i \in R_j$ и $h_i = k_2$, где $R_j = \{k_1,k_2\}$ и $R = \{R_1,...,R_m\}$, $k_1 \notin \{h_1,...,h_{i-1}\}$, то гены h_i и $h_p = k_1$, где $p \in [i+1,n]$ меняются местами.

Здесь множество R – множество ограничений на порядок обхода, R_j – подмножество множества R, состоящее из двух городов: города-отправителя и города-получателя.

После проверки и, если потребовалось, корректировки хромосома добавляется в исходную популяцию, и продолжается выполнение генетического алгоритма.

2.2 Учет ограничений в муравьином алгоритме

Пусть есть область видимости муравья - множество $N = \{n_1,...,n_m\}$. И есть множество доступных для перехода городов $I = \{1,...,m\}$, тогда $\forall i$

$$\begin{cases} i \in R_j \cap i = k_2 \cap k_1 \notin [1,...,i-1], n_i = 0 \\ иначе, n_i = 1 \end{cases}$$

где $R_j = \{k_1,k_2\}$ и $R = \{R_1,...,R_p\}$.

Здесь множество R – множество ограничений на порядок обхода, R_j – подмножество множества R, состоящее из двух городов: города-отправителя и города-получателя.

3 Гибридная муравьиная система

Гибридная муравьиная система (ГМС) – алгоритм муравьиной колонии плюс локальный поиск. ГМС строит ряд решений алгоритмом муравьиной колонии, и затем процедура локального поиска приводит их к локальному оптимуму.

3.1 Муравьиный алгоритм

Муравьи размещаются в случайно выбранные города без совпадений. Каждый муравей начинает движение из своего узла и проходит новые узлы до тех пор, пока все узлы графа не будут посещены. Находясь в узле i, муравей выбирает вероятностно следующий узел j из набора допустимых узлов $J_{i,k}$ по правилу, определяющему вероятность перехода *k*-ого муравья из города *i* в город *j* [2]:

$$\begin{cases} P_{ij}(t) = \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum\limits_{l \in J_{i,k}} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}}, j \in J_{i,k}; \\ P_{ij}(t) = 0, j \notin J_{i,k}; \end{cases} \quad (3)$$

где *α*, *β* – параметры, задающие веса следа феромона. При *α* = 0 алгоритм вырождается до жадного алгоритма (будет выбран ближайший город). $J_{i,k}$ – общая зона всех городов.

Пройдя ребро (*i*,*j*), муравей откладывает на нём некоторое количество феромона, которое должно быть связано с оптимальностью сделанного выбора. Пусть $T_k(t)$ есть маршрут, пройденный муравьём k к моменту времени t, $L_k(t)$ – длина этого маршрута, а Q – параметр, имеющий значение порядка длины оптимального пути. Тогда откладываемое количество феромона может быть задано в виде [2]:

$$\Delta\tau_{ij,k}(t)=\begin{cases}\dfrac{Q}{L_k(t)},(i,j)\in T_k(t);\\ 0,(i,j)\notin T_k(t);\end{cases} \tag{4}$$

Правила внешней среды определяют, в первую очередь, испарение феромона. Пусть $p\in[0,1]$ есть коэффициент испарения, тогда правило испарения имеет вид [2]:

$$\tau_{ij}(t+1)=(1-p)\cdot\tau_{ij}(t)+\Delta\tau_{ij}(t);\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij,k}(t) \tag{5}$$

где m – количество муравьёв в колонии.

В начале алгоритма количество феромона на рёбрах принимается равным небольшому положительному числу. Общее количество муравьёв остаётся постоянным и равным количеству городов, каждый муравей начинает маршрут из своего города.

Обозначим через T* наилучший текущий маршрут, через L* – его длину. Тогда рёбра такого маршрута получат дополнительное количество феромона [2]:

$$\Delta\tau_e=Q/L^* \tag{6}$$

4

3.2 Локальный поиск

Как только будет получено решение задачи муравьиным алгоритмом, применим к нему локальный поиск. В качестве локального поиска будем использовать генетический алгоритм.

Используем путевое представление пути. Путевое представление – это естественное представление тура. Например, тур 5→1→7→8→9→4→6→2→3 представлен просто как (5 1 7 8 9 4 6 2 3).

В качестве оператора кроссовера будем использовать кроссовер порядка. Основная идея кроссовера порядка заключается в том, что важен порядок городов, а не их позиции.

В качестве оператора мутации используется модификация классической мутации - сальтация: в кодировке χ случайным образом выбираются 2k (2k < L) локусов, в каждой паре локусов переставляются значения аллелей [3].

4. Результаты работы гибридной муравьиной системы

Результаты работы описанной гибридной муравьиной системы приведены в таблице 4.1. При этом параметры муравьиного алгоритма следующие: $\alpha = 1$, $\beta = 10$, Q = 7, p = 0.9 Параметры генетического алгоритма: вероятность кроссовера – 0.7, вероятность мутации – 0.001, мощность популяции – 1000 особей. Заданы следующие ограничения на порядок обхода городов: 9 24 12 18 20 4 3 15 13 6, где каждый нечетный город – город-отправитель, а каждый четный – город-получатель, т.е. в полученном маршруте каждый четный город пары должен стоять после своего нечетного соседа.

Таблица 1 наглядно иллюстрирует, как изменяется результат работы программы в зависимости от времени жизни колонии.

Из таблицы1 видно, что чем больше время жизни колони, тем точнее результат, но при этом значительно возрастает вычислительное время.

На рисунке 1 показан лучший найденный маршрут, соответствующий расстоянию 47,7972905834986. Учтены следующие ограничения: Луцк – Чернигов, Одесса – Ужгород, Херсон - Запорожье, Житомир – Симферополь, Полтава – Киев. Порядок обхода городов в найденном маршруте: Полтава, Сумы, Харьков, Луганск, Донецк, Житомир, Херсон, Николаев, Одесса, Винница, Хмельницкий, Тернополь, Ровно, Луцк, Львов, Ужгород, Ив. Франковск, Черновцы, Симферополь, Киев, Чернигов, Черкассы, Кировоград, Запорожье, Днепропетровск, Полтава.

Таблица 1 Зависимость результатов работы ГМС от времени жизни колонии

Результат работы муравьиного алгоритма		Результат работы локального поиска	
Время жизни колонии 10			
Глобальный оптимум	Маршрут	Локальный оптимум	Маршрут
55,25177478	15, 20, 11, 12, 7, 22, 13, 1, 4, 2, 8, 19, 16, 6, 3, 0, 21, 14, 9, 17, 5, 18, 10, 23, 24, 15	50,8503447	7, 12, 11, 20, 3, 13, 1, 4, 2, 8, 19, 16, 6, 15, 0, 21, 17, 23, 5, 18, 10, 9, 14, 24, 22, 7
54,80879413	10, 5, 17, 9, 14, 21, 3, 0, 24, 22, 7, 11, 20, 12, 15, 4, 1, 13, 16, 19, 8, 2, 6, 23, 18, 10	51,0102819	7, 11, 12, 20, 3, 4, 1, 9, 8, 2, 13, 16, 6, 0, 21, 23, 18, 10, 5, 17, 9, 14, 15, 24, 22, 7
54,80879413	18, 10, 5 , 17, 9, 14, 21, 3, 0, 24, 22, 7, 11, 20, 12, 15, 4, 1, 13, 16, 19, 8, 2, 6, 23, 18	50,7362600	20, 3, 4, 11, 12, 7, 22, 13, 15, 0, 21, 14, 9, 17, 5, 18, 10, 23, 24, 16, 19, 8, 2, 6, 1, 20
Время жизни колонии 100			
52,69478765	10, 5, 17, 9, 14, 21, 0, 3, 12, 11, 20, 15, 4, 1, 13, 16, 19, 8, 2, 22, 7, 6, 24, 23, 18, 10	50,2083291	21, 0, 3, 22, 7, 12, 11, 20, 15, 4, 1, 19, 8, 2, 13, 16, 9, 6, 23, 17, 5, 18, 10, 24, 14, 21
52,69788502	8, 2, 19, 16, 13, 1, 20, 11, 12, 0, 21, 14, 9, 17, 23, 5, 18, 10, 3, 6, 24, 22, 7, 4, 15, 8	48,3568259	16, 19, 8, 2, 3, 20, 11, 12, 0, 21, 14, 9, 17, 5, 18, 10, 23, 15, 13, 24, 22, 7, 4, 1, 6, 16
54,10792413	17, 9, 14, 21, 0, 3, 22, 7, 11, 20, 12, 15, 4, 1, 13, 16, 19, 8, 2, 6, 24, 23, 5, 18, 10, 17	48,5314673	14, 21, 0, 3, 13, 9, 22, 7, 11, 20, 4, 1, 6, 16, 19, 8, 2, 15, 2, 23, 5, 18, 10, 17, 24, 14
Время жизни колонии 1000			
52,59700609	10, 5, 17, 9, 14, 21, 0, 3, 22, 7, 11, 20, 12, 15, 4, 1, 2, 8, 19, 16, 13, 6, 24, 23, 18, 10	48,5378816	21, 3, 22, 7, 12, 11, 20, 15, 4, 1, 2, 8, 19, 13, 16, 9, 6, 0, 23, 17, 5, 18, 10, 24, 14, 21
53,85533885	1, 13, 16, 19, 8, 2, 11, 20, 12, 0, 3, 21, 17, 14, 9, 10, 5, 18, 23, 6, 24, 22, 7, 15, 4, 1	48,9279319	9, 14, 3, 22, 7, 12, 11, 20, 15, 4, 1, 2, 8, 19, 16, 13, 24, 6, 0, 21, 23, 18, 10, 5, 17, 9
51,48311477	4, 1, 13, 16, 19, 8, 2, 20, 11, 12, 0, 3, 21, 17, 14, 9, 10, 18, 5, 23, 6, 24, 22, 7, 15, 4	47,7972905	13, 16, 19, 8, 2, 3, 20, 11, 12, 0, 21, 17, 14, 9, 10, 18, 5, 23, 15, 6, 24, 22, 7, 4, 1, 13

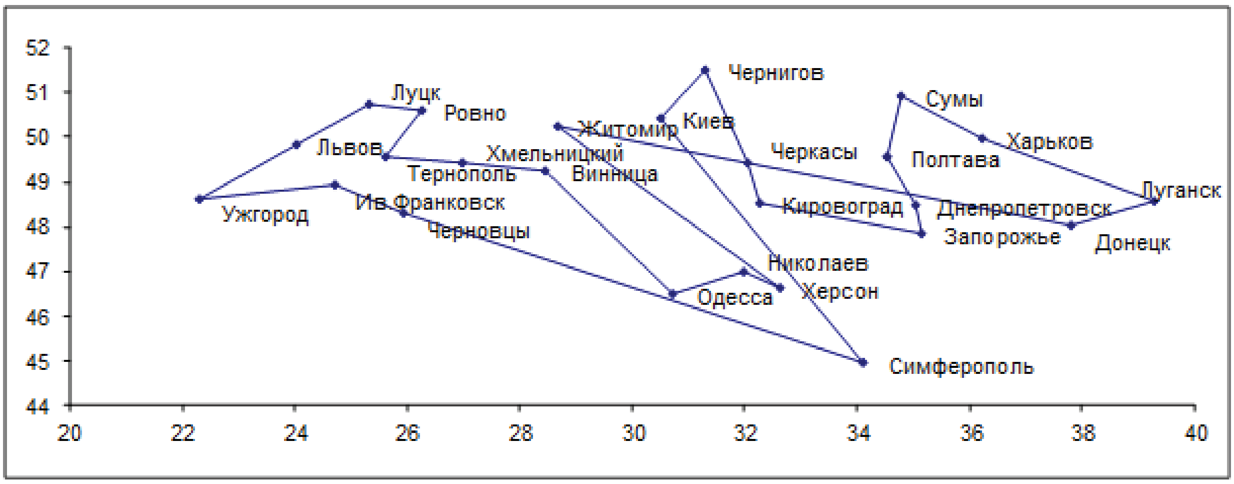

Рисунок 1. Лучший маршрут

Выводы

Задача коммивояжера с ограничениями на направленность маршрута была формализована и решена с помощью гибридной муравьиной системы, включающей в себя муравьиный алгоритм и локальный поиск. В качестве локального поиска был предложен генетический алгоритм. Такая гибридная муравьиная система позволяет за приемлемое время получать результаты решения задачи на 5% лучше, чем генетический алгоритм, и на 7% - чем муравьиный.

Литература

1. Методичні вказівки до виконання лабораторних робіт за курсом “ Еволюційні обчислення у технічних задачах ” (для студентів спеціальності 7.091503 “Спеціалізовані комп’ютерні системи” (СКС) / Укл. Ю.О. Скобцов, С.В. Хмільовий, Т. О. Васяєва – Донецьк, 2008 – 91 с.

2. Штовба С.Д. Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях, 2003, №4, с.70-75.

3. Батищев Д.И., Неймарк Е.А., Старостин Н.В. Применение генетических алгоритмов к решению задач дискретной оптимизации. Учебно-методический материал по программе повышения квалификации «Информационные технологии и компьютерное моделирование в прикладной математике». Нижний Новгород, 2007, 85с.