

В. М. Курейчик, д-р техн. наук, проф., vmkureychik@sfedu.ru,

Ю. А. Логунова, аспирант, Julia1000@yandex.ru,

Федеральное государственное автономное образовательное учреждение высшего образования
"Южный федеральный университет", г. Таганрог

Анализ перспективности применения генетического алгоритма при решении задачи коммивояжера

Выполнен анализ перспективности применения генетического алгоритма (ГА) при решении актуальной в области искусственного интеллекта задачи коммивояжера. В модифицированном алгоритме применены показатели, которые характеризуют степень разнообразия популяции. Специально разработанное программное обеспечение позволило провести тестирование ГА на известной бенчмарке att-48 и показать полезность применения разработанных показателей на практике.

Ключевые слова: задача коммивояжера, генетический алгоритм, кроссинговер, оператор, популяция, хромосома

Введение

Сложные комбинаторные оптимизационные задачи имеют важное прикладное значение в различных областях человеческой деятельности: экономической, научной, производственной. Справедливо отметить, что решение оптимизационных задач является одним из основных видов научно-инженерной деятельности. Среди множества попыток разработки оптимизационных и поисковых процедур поиска решения подобных задач можно выделить отдельный класс методов оптимизации, которые имитируют реальные процессы, происходящие в природе. К ним относят:

1. Искусственные нейронные сети, которые построены по принципу функционирования сетей нервных клеток центральной нервной системы живого организма.

2. Алгоритм имитации отжига, в основе которого лежит физический процесс, происходящий при кристаллизации вещества.

3. Эволюционные алгоритмы: эволюционное программирование (Л. Дж. Фогель, 1960), генетические алгоритмы (Д. Холланд, 1975), генетическое программирование (Д. Коза, 1992) и другие методы, которые основаны на селекции

лучших элементов в популяции, в соответствии с эволюционной теорией Ч. Дарвина.

Следует отметить, что с развитием генетических и эволюционных алгоритмов в течение последних лет различия между ними постепенно нивелируются. В настоящее время используется новая парадигма исследований на основе генетических алгоритмов и ее различных модификаций [1].

В данной работе мы проанализируем генетический алгоритм для решения общеизвестной задачи комбинаторной оптимизации — задачи коммивояжера (ЗК). В неформальной постановке она заключается в отыскании кратчайшего пути (маршрута), проходящего через все заданные города по одному разу с последующим возвратом в исходный город. Данная задача проста в описании, но имеет высокую комбинаторную сложность. Она возникает в обширном классе приложений, например: поиск оптимального маршрута [2]; задача нахождения маршрута коня, проходящего через все поля шахматной доски по одному разу [3]; задача сбора монет из таксофонов [4]; задача предсказания функций протеинов [5]; задача построения изображения непрерывной линией [6] и многих других. Существует це-

лый класс методов, которые позволяют находить точное решение ЗК, однако они имеют медленную скорость сходимости и на практике для ЗК большой размерности зачастую оказываются малоэффективными. Для решения таких задач, как правило, приходится жертвовать точностью решения и применять эвристические методы. К таким методам относят генетические алгоритмы (ГА). Они обеспечивают высокое качество решений, однако оно напрямую зависит от выбора значений параметров алгоритма. В настоящее время разработано множество различных операторов ГА, комбинация которых также влияет на качество получаемых решений. В связи с этим появляется необходимость оценки перспективности применения ГА с выбранными операторами на предварительном этапе: еще до применения на реальной задаче.

Задачей данного исследования является проведение сравнительного анализа некоторых операторов и эвристик, помогающих выйти из локальных оптимумов. Поскольку наибольшее влияние на ГА оказывает оператор кроссинговера, то перед собой мы поставили задачу изучить влияние одних из лучших на сегодняшний день кроссинговеров на такой показатель, как вырождаемость популяции. Важность данного исследования состоит в том, чтобы понять, на каком этапе генетический алгоритм приводит к деградации популяции. В связи с этим возникла необходимость сравнивать хромосомы и определять степень их различия. Для сравнения хромосом в популяции будем использовать понятие из теории информации и компьютерной лингвистики — расстояние Дамерау—Левенштейна.

Новизна подхода состоит в том, чтобы использовать выведенные на основе расстояния Дамерау—Левенштейна показатели, которые могут быть рассчитаны в общем случае на любом этапе в блоке эволюционной адаптации. Это позволяет эффективно подбирать параметры генетического алгоритма и вероятность применения генетических операторов, следовательно, сократить время поиска локально-оптимального решения. **Принципиальное отличие** предложенного метода заключается в том, что авторы отошли от традиционной схемы решения ЗК с помощью ГА (практическая задача → модель → генетический алгоритм → реализация → анализ полученных результатов), а реализовали новый подход, который

позволяет с помощью показателей, характеризующих степень разнообразия особей в популяции, анализировать любой генетический алгоритм с использованием известных бенчмарков еще до начала применения его на практической задаче.

Постановка задачи и генетический алгоритм для решения ЗК

Математически классическая ЗК выглядит следующим образом: дано множество W , где $w_1, w_2, \dots, w_n$ — города, которые должен посетить коммивояжер. И даны расстояния между каждой парой городов: $d_{ij} = d(w_i, w_j) | i, j = 1, \dots, n$. Необходимо найти минимальное значение:

$\sum_{i=1}^{n-1} d_{\varphi(i), \varphi(i+1)} + d_{\varphi(n), \varphi(1)}$ среди всех $(n-1)!/2$ перестановок φ . Также в задаче накладываются определенные условия: неотрицательности $d_{ij} \geq 0$; симметричности $d_{ij} = d_{ji}$; запрет на петли $d_{ii} = \infty$, $i, j = \overline{1, n}$, и удовлетворения неравенству треугольника: $d_{ij} + d_{jp} \geq d_{ip}$, $i, j, p = \overline{1, n} | i \neq j, i \neq p, j \neq p$ [7].

В терминах теории графов моделью классической ЗК является реберно-взвешенный граф $G = (K, T)$, в котором необходимо найти гамильтонов цикл (замкнутый маршрут, который проходит все вершины графа по одному разу) минимальным значением суммарного числа расстояний. При этом вершины графа k_i , $i = 1, \dots, N$ — это города, которые необходимо посетить коммивояжеру, а ребра $t_{i,j}$, $i = 1, \dots, N$, $i \neq j$ — расстояния между городами [7].

В данной работе мы рассмотрим симметричную ЗК.

Генетические алгоритмы для решения ЗК

Генетические алгоритмы относятся к классу вероятностных вычислений, которые моделируют механизмы естественного отбора и эволюции в природе. В данных алгоритмах существуют такие понятия, как популяция, функция приспособленности, генетические операторы [8]. Фундаментальные основы ГА представлены во многих статьях и книгах, опубликованных за последние годы по эволюционной оптимизации *NP*-сложных задач из различных прикладных областей, таких как биология, микроэлектроника, криптоанализ, медицина, телекоммуникации и др. Пример

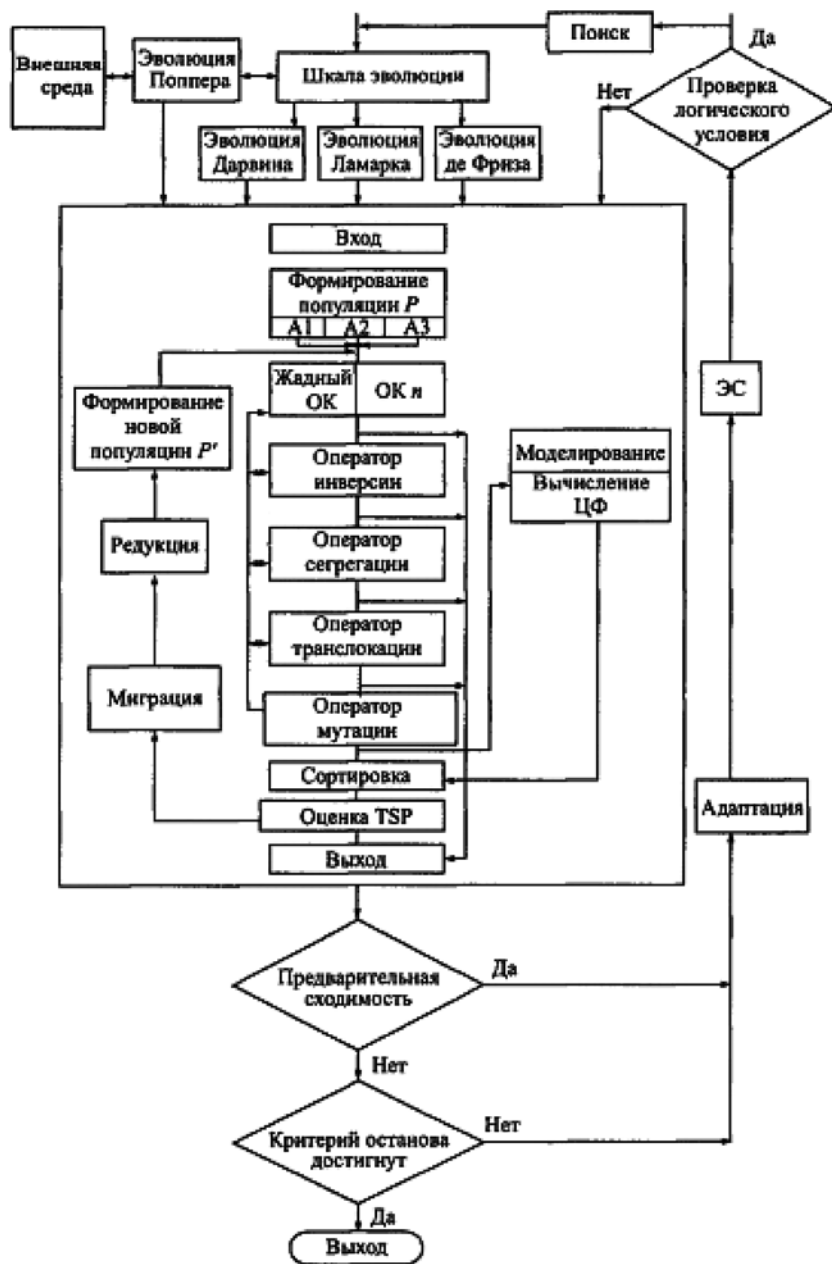


Рис. 1. Структурная схема генетического алгоритма решения задачи коммивояжера

структурной схемы работы ГА при решении ЗК приведен на рис. 1 [8].

Приведенный генетический алгоритм позволяет найти множество альтернативных решений и работает по принципу "выживает сильнейший". Каждое альтернативное решение в ГА соответствует какому-либо гамильтоновому циклу (хромосоме). В первом поколении все хромосомы генерируются случайно. Далее ГА начинает формировать новую популяцию. Как правило, размер популяции постоянен на всем этапе работы алгоритма. Процесс репродукции

начинается с наиболее важного оператора — кроссинговера.

Оператор кроссинговера — это языковая конструкция, позволяющая на основе преобразования (скрещивания) хромосом родителей (или их частей) создавать хромосомы потомков [9].

Структура оператора кроссинговера в основном определяет эффективность работы всего генетического алгоритма. В настоящее время разработано множество операторов кроссинговера. По результатам исследования в работе [10] наиболее эффективными по скорости сходимости считаются следующие кроссинговеры: Genetic edge recombination crossover (ERX), Partially-mapped crossover (PMX) и Position based crossover (POS). Для изучения эффективности генетического алгоритма нами был выбран кроссинговер PMX. Также в наш сравнительный анализ включен еще один эффективный кроссинговер — Edge assembly crossover (EAX-1AB), который по сравнению с упомянутыми кроссинговерами был разработан совсем недавно — в 2006 г. японским ученым Yuichi Nagata [11].

Partially-mapped crossover (PMX) — оператор кроссинговера, который был предложен Д. Голдбергом и Р. Линглом в 1985 г. [12]. Суть данного кроссинговера заключается в том, что часть хромосомы одного родителя заменяется частью хромосомы

второго родителя, все оставшиеся гены соответственно в хромосоме изменяются, образуя тем самым первого потомка. Проводится та же операция замены части хромосомы второго родителя на часть хромосомы 1-го и изменение оставшихся генов, формируя тем самым второго потомка.

Edge assembly crossover (EAX). Алгоритм данного кроссинговера работает следующим образом.

1. Пара родительских особей обозначаются как маршрут A и маршрут B . Граф T_{AB} получается путем наложения двух маршрутов: A и B .

2. Разделить ребра графа T_{AB} на циклы, которые представляют собой замкнутые циклы. В циклы поочередно входят ребра то из маршрута A , то из B .

3. Сформировать множество K , представляющее собой любую комбинацию из AB -циклов. Единственное ограничение: AB -циклы, сформированные из двух ребер, в построении множества K не участвуют.

4. Создается промежуточное решение: из множества K удаляются ребра, соответствующие маршруту A , и добавляются ребра из маршрута B .

5. Модифицируем промежуточное решение для того чтобы получить допустимое решение ЗК. Если два цикла связаны, то эвристически выбирается одно ребро из каждого цикла и удаляется, по такому же принципу добавляются два ребра для их соединения. На рис. 2 показан пример работы кроссингвера EAX [11].

Этот этап имеет наибольшую вычислительную сложность в данном алгоритме. Разберем этот шаг подробнее:

5-1. Пусть U_i ($i = 1, \dots, k$) — множество ребер, включенных в i -й подмаршрут, где k — число подмаршрутов в промежуточном решении.

5-2. Выбрать самый маленький подмаршрут из $|U_i|$ ($i = 1, \dots, k$), где $|U_i|$ — число ребер в U_i . Пусть U_r — выбранный подмаршрут.

5-3. Найти пару ребер, $e \in U_r$ и $e' \in U_j$ ($j \neq r$), так что он минимизирует $\{-w(e) - w(e') + w(e'') + w(e''')\}$, где e'' и e''' определяются таким образом, чтобы связывать два подмаршрута. Пусть U_s — подмаршрут, содержащий ребро e' . U_r и U_s объединяются следующим образом: $U_r := (U_r \cup U_s - \{e, e'\}) \cup \{e'', e'''\}$ и пустого U_s . $U_s := U_k$ и вычитаем 1 из k .

5-4. Если k равно 1, U_1 является допустимым маршрутом, то завершить, иначе переходим к п. 5-2.

В наших расчетах будет участвовать усовершенствованный кроссингвер EAX-1AB, поскольку он от 2 до 7 раз (в зависимости от параметров) быстрее EAX, и доказано на различных бенчмарках, что его можно эффективно применять на графе до 5915 вершин [11].

Отличие EAX-1AB от EAX состоит в следующем:

1. Множество K формируется только из одного цикла AB , при этом промежуточное решение имеет тенденцию быть сходным с маршрутом A , поскольку потомки генерируются путем удаления небольшого числа ребер из маршрута A и добавления того же числа ребер из B соответственно.

2. С помощью специальной эвристики сокращена вычислительная сложность п. 5 алгоритма EAX, что несомненно является преимуществом модифицированного кроссингвера, поскольку п. 5 алгоритма имеет наибольшее влияние на вычислительную сложность всего генетического алгоритма.

На рис. 1 представлены операторы инверсии, сегрегации, транслокации и мутации. В основном они служат для увеличения разнообразия популяции. В связи с этим имеет смысл рассчитывать показатель степени деградации популяции после работы кроссингвера. Далее эту информацию полезно использовать для настройки параметров следующих операторов.

Расстояние Дамерау—Левенштейна [13, 14] в рассматриваемом генетическом алгоритме будет участвовать для определения степени

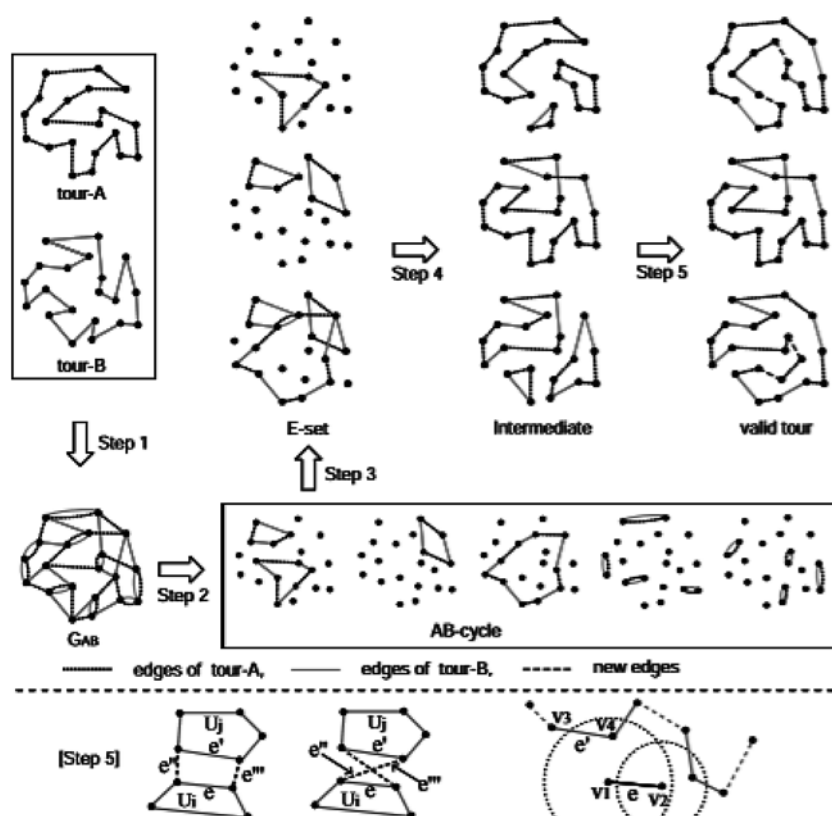


Рис. 2. Пример работы кроссингвера EAX

родства хромосом в популяциях. Данное расстояние характеризует минимальное число операций вставки, удаления, замены одного символа и транспозиции двух соседних символов, оно необходимо для того, чтобы перевести одну строку в другую. Очевидно, что чем меньше это расстояние в генетическом алгоритме между хромосомами, тем более родственными являются рассматриваемые хромосомы.

В общем случае расстояние Дамерау—Левенштейна может быть определено по рекуррентной формуле:

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i & ; j = 0, i > 0 \\ j & ; i = 0, j > 0 \\ \min(\\ \quad D(i, j - 1) + 1, \\ \quad D(i - 1, j) + 1, & ; j > 0, i > 0 \\ \quad D(i - 1, j - 1) + k(T_1[i], T_2[j]) \\ \quad) \end{cases},$$

где T_1 и T_2 — рассматриваемые строки.

Для оценки перспективности применения генетического алгоритма выведем следующие показатели:

Среднее значение различия хромосомы, соответствующей оптимальному решению в популяции, с остальными хромосомами в популяции:

$$Aver_{opt} = \sum_{j=1}^n d_{opt,j} / n,$$

где $j \neq opt$, n — размер популяции; $d_{opt,j}$ — расстояние Дамерау—Левенштейна между оптимальным решением и j -й хромосомой.

Среднее значение коэффициента различия хромосом в популяции:

$$Aver_{pop} = \left(\sum_{i=1}^n \sum_{j=1}^n d_{i,j} \right) / n,$$

где $i \neq j$, n — размер популяции; $d_{i,j}$ — расстояние Дамерау—Левенштейна между i -й и j -й хромосомами.

В общем случае тенденция уменьшения приведенных выше показателей свидетельствует о снижении разнообразия в популяции, увеличения риска попадания в локальный оптимум и, как следствие, необходимости изменения настроек ГА. Данные показатели могут рассматриваться на любом этапе эволю-

ционной адаптации. В данном исследовании в модифицированном алгоритме $Aver_{opt}$ и $Aver_{pop}$ применяют после оператора кроссинговера.

Разработка программы и экспериментальная часть

Генетический алгоритм решения задачи коммивояжера был запрограммирован на языке Java на платформе Windows.

При этом все исследования проводили на компьютере типа Intel Core i7 CPU 2.6 GHz и ОЗУ размером 16 Гбайт.

Настройки генетического алгоритма:

- 1) кроссинговеры: EAX-1, PMX;
- 2) способ выбора начальной популяции — алгоритм ближайшего соседа из каждой вершины графа;
- 3) размер начальной популяции — n , где n — число вершин;
- 4) за каждую итерацию выполняется $n/2$ кроссинговеров, родители для которых выбираются случайным образом из текущей популяции;
- 5) жадный оператор селекции;
- 6) число итераций — 50;
- 7) показатели $Aver_{opt}$ и $Aver_{pop}$ рассчитывают после оператора кроссинговера.

Эксперименты были проведены на бенчмарке att-48.

Результаты разработанной программы представлены на рис. 3, 4. В приведенных ниже графиках по оси x номер поколения, по оси y — значение коэффициента $Aver_{opt}$ или $Aver_{pop}$.

Как видно из графика рис. 3, ГА с кроссинговером PMX уже начиная с 5-й итерации,

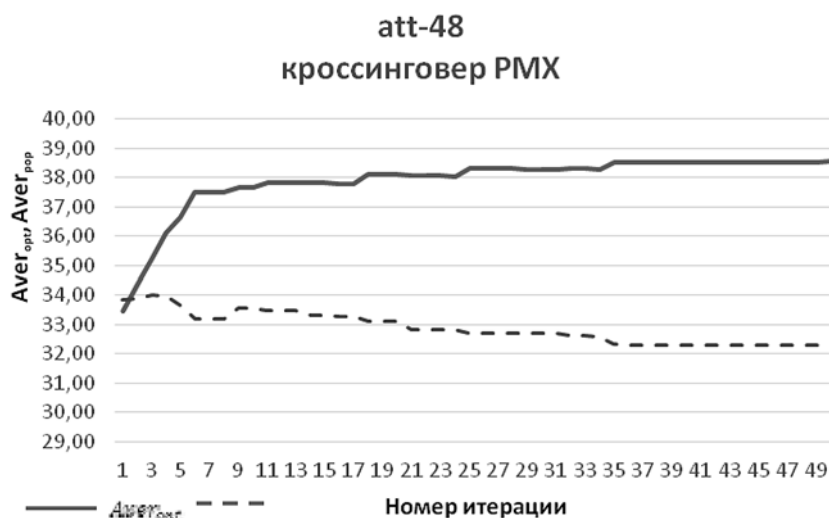


Рис. 3. Показатели $Aver_{opt}$ и $Aver_{pop}$ с кроссинговером PMX

att-48 кроссинговер EAX



Рис. 4. Показатели $A_{\text{вер_опт}}$ и $A_{\text{вер_поп}}$ с кроссинговером EAX

несмотря на тенденцию формирования разнообразных решений (показатель $A_{\text{вер_опт}}$ растет), показывает постоянное снижение в разнообразии популяции (показатель $A_{\text{вер_поп}}$ снижается).

На рис. 4 видно, что с 29-й по 36-ю итерацию алгоритм показывает рост разнообразия популяции, с 36-й по 43-ю временно происходит обратная тенденция, но затем снова наблюдается рост в разнообразии популяции.

Заключение

Данная работа относится к области искусственного интеллекта. ЗК активно используются в транспортных задачах, интеллектуальном проектировании, при решении задач поисковой оптимизации. Разработка методов решения ЗК по-прежнему остается актуальной задачей.

В результате исследования был разработан модифицированный генетический алгоритм, который показывает степень разнообразия популяции. Данная информация важна, поскольку деградация популяции может приводить к преждевременной сходимости алгоритма. Новизна подхода заключается в использовании специальных показателей: $A_{\text{вер_опт}}$ и $A_{\text{вер_поп}}$. Также было разработано специальное программное обеспечение. Результаты исследования на известной бенчмарке для ЗК с 48 городами (att-48) показали, что ГА с кроссинговером PMX имеет тенденцию к деградации популяции с каждым следующим поколением, начиная с 5-й итерации. Таким образом, ГА с PMX не является перспективным для запуска на больших объемах данных в исходном виде. Кроссинговер

EAX не имеет резкой тенденции к вырождаемости популяции, в связи с этим ГА с EAX не требует обязательного введения многочисленных дополнительных операторов для создания разнообразия в популяции, следовательно, он стабильно работает с низкими производительными затратами. Оптимальность (способность алгоритма выбирать "наилучший маршрут") ГА с EAX также выше, чем ГА с PMX, следовательно, можно считать ГА с EAX более перспективным.

В общем случае показатели $A_{\text{вер_опт}}$ и $A_{\text{вер_поп}}$ могут служить полезным критерием для оценки

генетического алгоритма, а именно: степени вырождаемости популяций и дальнейшей настройки его операторов. Важно отметить, что показатели $A_{\text{вер_опт}}$ и $A_{\text{вер_поп}}$ также увеличивают гибкость ГА. При этом исходный алгоритм целесообразно проверять с помощью показателей $A_{\text{вер_опт}}$ и $A_{\text{вер_поп}}$ на небольших объемах данных перед использованием на реальной задаче в любом блоке эволюционной адаптации.

Работа выполнена за счет частичного финансирования по гранту РФФИ № 18-0700050 и ГЗ № 25537.2017/6.7.

Список литературы

1. Гладков Л. А., Курейчик В. В., Курейчик В. М., Сороколетов П. В. Биотинспирированные методы в оптимизации. М.: Физматлит, 2009. 384 с.
2. Ismail Z., Ibrahim W. R. W. Traveling Salesman Approach for Solving Petrol Distribution Using Simulated Annealing // American Journal of Applied Sciences, 2008. Vol. 5, Is. 11. P. 1543–1546.
3. McKay B. D. Knight's tours of an 8×8 chessboard // Technical Report. Department of Computer Science, Australian National University. Canberra, 1997. № TR-CS-97-03. 4 p.
4. Lenstra J., Kan A. R. Some simple applications of the travelling salesman problem // Operational Research Quarterly, 1975. Vol. 26, N. 4. Part 1. P. 717–733.
5. Johnson O., Liu J. A traveling salesman approach for predicting protein functions // Source Code for Biology and Medicine. 2006. Vol. 1, N. 3. P. 1–7.
6. Borch R., Herman A. Continuous Line Drawings via the Traveling Salesman Problem // Operations Research Letters. 2004. Vol. 32. P. 302–303.
7. Колесников А. В., Кириков И. А., Листопад С. В., Румовская С. Б., Доманицкий А. А. Решение сложных задач коммивояжера методом функциональных гибридных интеллектуальных систем / Под ред. А. В. Колесникова. М.: ИПИ РАН, 2011.
8. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы. 2-е изд., испр. и доп. М.: Физматлит, 2006. 320 с.
9. Курейчик В. М. Модифицированные генетические операторы // Известия Южного Федерального Университета. 2009. № 12 (101). С. 7–10.

10. Starkweather T., McDaniel S., Mathias K., Whitley C., Whitley D. A Comparison of Genetic Sequencing Operators // Proceedings on the Fourth International Conference on Genetic Algorithms, edited by Belew R. and Booker L., Los Altos, CA: Morgan Kaufmann Publishers, 1991. P. 69–76.

11. Nagata Y. Fast EAX Algorithm Considering Population Diversity for Traveling Salesman Problems // Evolutionary Computation in Combinatorial Optimization 6th European Conference, EvoCOP 2006 Budapest, Hungary, 2006. P. 171–182.

12. Goldberg D. E., Lingle R. Alleles, Loci, and the Traveling Salesman Problem // Proceedings of the First International Conference on Genetic Algorithms and Their Application / ed. by Grefenstette J. Lawrence Erlbaum Associates, Hillsdale, NJ, 1985. P. 154–159.

13. Damerau F. J. A technique for computer detection and correction of spelling errors // Communications of the ACM, ACM. 1964. Vol. 7, N. 3. P. 171–176.

14. Левенштейн В. И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады Академии Наук СССР, 1965. Т. 163, № 4. С. 845–848.

V. M. Kureichik, Professor, CAD Department Senior Researcher, Vmkureychik@sfedu.ru,

J. A. Logunova, Graduate Student, Julia1000@yandex.ru,

Federal State-Owned Autonomy Educational Establishment of Higher Education

Southern Federal University, Taganrog, Russia

The Genetic Algorithm Application Prospects Analysis for the Traveling Salesman Problem Solution

This work belongs to the artificial intelligence field. The Traveling salesman problem (TSP) is considered in this article, which is actively used in transport tasks, intellectual design, in search optimization solving problems. The development of new solving methods for TSP is still an urgent task. It is NP-hard problem. Usually, the algorithm time complexity finding the exact solution is a factorial or exponential dependence on the input data. It is possible to allocate a separate class of optimization methods among a lot of search procedures for TSP solving, which imitates real nature processes: genetic and bio-inspired algorithms. These algorithms are used for large dimensions tasks. In this case the optimal solution cannot be obtained over the acceptable time. In this situation you need sacrifice the accuracy of the solution and find a good solution in a relatively short time. The solution quality depends greatly on choice of algorithm parameter values. The genetic algorithm can lead to the population degradation in the wrong choice of operator, and, as a result, the obtained solution is far from optimal. In this regard, we propose to use some special indicators for investigating the prospects of using the genetic algorithm. This indicators show the population diversity degree. Also they can be used to algorithm parameters adjust. Specially developed software allowed testing the GA on the well-known benchmark for TSP with 48 cities (att-48) and showed the usefulness of the developed indicators in practice.

Keywords: traveling salesman problem, genetic algorithm, crossover, population, chromosome, artificial intelligence, Damerau-Lowenstein distance

DOI: 10.17587/it.24.691-697

References

1. Gladkov L. A., Kureichik V. V., Kureichik V. M., Sorokoletov P. V. Bioinspirovannye metody v optimizatsii [Bioinspired methods in optimization], Moscow, Fizmatlit, 2009, 384 p.

2. Ismail Z., Ibrahim W. R. W. Traveling Salesman Approach for Solving Petrol Distribution Using Simulated Annealing, *American Journal of Applied Sciences*, 2008, vol. 5, is. 11, pp. 1543–1546.

3. McKay B. D. Knight's tours of an 8×8 chessboard, Technical Report. Department of Computer Science, Australian National University. Canberra, 1997. № TR-CS-97-03. 4 p.

4. Lenstra J., Kan A. R. Some simple applications of the travelling salesman problem, *Operational Research Quarterly*, 1975, vol. 26, no. 4, part 1, pp. 717–733.

5. Johnson O., Liu J. A traveling salesman approach for predicting protein functions, *Source Code for Biology and Medicine*, 2006, vol. 1, no. 3, pp. 1–7.

6. Borch R., Herman A. Continuous Line Drawings via the Traveling Salesman Problem, *Operations Research Letters*, 2004, vol. 32, pp. 302–303.

7. Kolesnikov A. V., Kirikov I. A., Listopad S. V., Rumovskaya S. B., Domanitsky A. A. Reshenie slozhnykh zadach kommi-voyazhera metodami funktsional'nykh gibridnykh intellektual'nykh sistem [Solving the complex traveling salesman problems with the method of functional hybrid intelligent systems], ed. A. V. Kolesnikov, Moscow, IPI RAS, 2011.

8. Gladkov L. A., Kureichik V. V., Kureichik V. M. Geneticheskie algoritmy [Genetic algorithms], 2nd ed., rev. and add. Moscow, Fizmatlit, 2006, 320 p.

9. Kureichik V. M. Modifitsirovannyye geneticheskie operatory [Modified genetic operators], *Proceedings of the Southern Federal University*, 2009, no. 12, pp. 7–10.

10. Starkweather T., McDaniel S., Mathias K., Whitley C., Whitley D. A Comparison of Genetic Sequencing Operators, *Proceedings on the Fourth International Conference on Genetic Algorithms*, edited by Belew R. and Booker L., Los Altos, CA, Morgan Kaufmann Publishers, 1991, pp. 69–76.

11. Nagata Y. Fast EAX Algorithm Considering Population Diversity for Traveling Salesman Problems, *Evolutionary Computation in Combinatorial Optimization 6th European Conference, EvoCOP 2006 Budapest, Hungary*, 2006, pp. 171–182.

12. Goldberg D. E., Lingle R. Alleles, Loci, and the Traveling Salesman Problem, *Proceedings of the First International Conference on Genetic Algorithms and Their Application*, edited by Grefenstette J., Lawrence Erlbaum Associates, Hillsdale, NJ, 1985, pp. 154–159.

13. Damerau F. J. A technique for computer detection and correction of spelling errors, *Communications of the ACM, ACM, March 1964*, 7 (3), pp. 171–176.

14. Levenshtein V. I. Dvoichnye kody s ispravleniem vypadeniy, vstavok i zameweniy simvolov [Binary codes with correction of fallouts, insertions and substitutions of symbols], *Reports of the Academies of Sciences of the USSR*, 1965, vol. 163.4, pp. 845–848.