

УДК 519.87:004

СРАВНЕНИЕ РАЗЛИЧНЫХ КРОССОВЕРОВ ПРИ РЕШЕНИИ ЗАДАЧИ КОММИВОЯЖЕРА ГЕНЕТИЧЕСКИМ АЛГОРИТМОМ С ПУТЕВЫМ ПРЕДСТАВЛЕНИЕМ

В. Г. Кобак, А. А. Рязанов

Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

Аннотация. Цель данной статьи — сравнение различных кроссоверов при зафиксированных мутациях. Исследуется одна из фундаментальных задач комбинаторной оптимизации, а именно задача коммивояжера. Для ее исследования использовался генетический алгоритм модифицированной модели Голдберга. Чтобы решить задачи коммивояжера указанным способом, разработано программное средство, с помощью которого были получены результаты, а на их основе сделан вывод, что комбинация упорядоченного кроссовера с односточечной мутацией оказалась наиболее эффективной. Результатом работы авторов является обоснование выбора наиболее перспективного кроссовера, который позволяет решать практические задачи большей размерности.

Ключевые слова: задача коммивояжера, генетический алгоритм, путевое представление, кроссовер, мутация, односточечная мутация, двухточечная непарная мутация.

COMPARISON OF DIFFERENT CROSSOVERS IN SOLVING THE TRAVELING SALESMAN PROBLEM BY A GENETIC ALGORITHM WITH PATH REPRESENTATION

Valeriy G. Kobak, Alexandr A. Ryazanov

Don State Technical University (Rostov-on-Don, Russian Federation)

Abstract. The purpose of this work is to compare different crossovers with fixed mutations. In this paper, we study one of the fundamental problems of combinatorial optimization, namely the traveling salesman problem. The genetic algorithm of the modified Goldberg model was used for its research. To solve the traveling salesman problem in this way, a software tool was developed, with the help of which the results were obtained and on their basis it was concluded that the combination of an ordered crossover with a single-point mutation turned out to be the most effective. The result of the authors' work is the justification of the choice of the most promising crossover, which allows solving practical problems of a larger dimension.

Keywords: travelling salesman problem, genetic algorithm, path representation, crossover, mutation, single-point mutation, two-point unpaired mutation.

Введение. Различные ситуации, связанные с решением задачи коммивояжера, будоражат умы людей уже много лет. Математическую задачу смогли сформулировать впервые только в XIX веке, а к точному ее решению приступили в середине XX века.

Задача коммивояжера является одной из основных задач комбинаторной оптимизации [1]. Решение ее — это нахождение экономичного маршрута, который по одному разу проходит через каждую точку города и обязательно возвращается в ту точку, из которой было начато движение. В теории графов такая задача является нахождением кратчайшего гамильтонова цикла графа [2]. На сегодняшний день для решения этой задачи размерности больше 100 не выявлено алгоритмов, которые дают полиномиальное время. Из этого следует, что данная задача относится к классу NP-полных [3]. До сих пор ученые разных стран не привели никаких твердых математических

доказательств того, что такого алгоритма в принципе не существует. Исходя из вышесказанного, можно утверждать, что решение задачи коммивояжера большой размерности является актуальной темой.

Основная часть. Длина маршрута, общая стоимость проезда или совокупное время нахождения в пути могут являться показателями рентабельности маршрута. Простейшее точное решение задачи коммивояжера может быть получено перечислением всех различных вариантов маршрута с фиксацией наиболее короткого. Однако совокупное время решения задачи будет резко увеличиваться при решении графов большой размерности для получения лучшего маршрута. В этом случае, если число городов равно n , то число всех возможных маршрутов будет равно $n!$ По теории графов и из дискретной математики известно, что число всех различных вариантов решения задачи равно $\frac{n!}{2 \cdot n} = \frac{(n-1)!}{2}$. В этих же источниках утверждается, что к категории трансвычислительных задач относится и задача нахождения гамильтонова цикла [4].

Постановка задачи. Задачу коммивояжера можно представить в виде графа. В таком случае зададим полный взвешенный граф $G = (V, E)$, где $|V| = n$ — множество вершин (города), $|E| = m$ — множество рёбер (пути между городами). Задана матрица весов $W(i, j), i, j \in 1, 2, \dots, n$, где (i, j) — элемент, который равен весу ребра (расстоянию между вершинами v_i и v_j).

Нужно найти такой маршрут из множества V , что фитнес функция будет минимальна [5]:
 $Fitness(\phi)P(\phi(1), \phi(v)) + \sum_t \{P(\phi(t), \phi(t+1))\} \rightarrow \min.$

Методы решения. Для решения поставленной задачи используется множество алгоритмов, как точных, так и приближенных, но в этой работе используется генетический алгоритм — модифицированная модель Голдберга. Генетические алгоритмы имитируют биологический процесс естественного отбора. Они зачастую применяются в решении оптимизационных задач с использованием таких естественных методов, как селекция, скрещивание и мутация [6]. Схема канонического генетического алгоритма представлена на рис. 1.

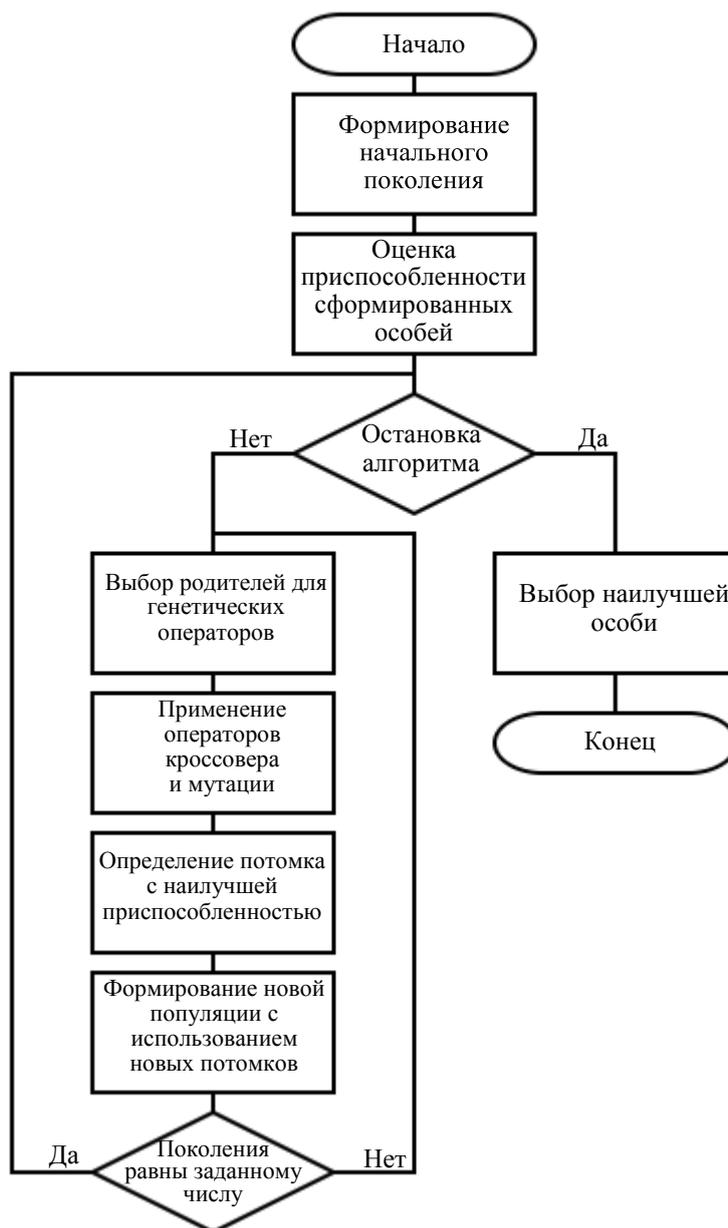


Рис. 1. Схема канонического генетического алгоритма (составлена авторами)

Модифицированная модель генетического алгоритма Голдберга отличается от канонической наличием турнирного отбора. При турнирном отборе после операции кроссовера производится мутация и худший потомок уничтожается, а оставшийся сравнивается с родительской особью, и если он лучше, то занимает ее место, а если получившийся потомок хуже родительской особи, то он уничтожается.

Наиболее популярным представлением маршрута при решении задачи коммивояжера генетическим алгоритмом является путевое представление [7]. Это популярное представление описывается в виде списка городов при движении по графу. Например, маршрут $3 \rightarrow 5 \rightarrow 2 \rightarrow 1 \rightarrow 8 \rightarrow 4 \rightarrow 7 \rightarrow 6 \rightarrow 9$ будет представлен в виде хромосомы $[3, 5, 2, 1, 8, 4, 7, 6, 9]$.

Упорядоченный кроссовер заключается во взятии части пути от первого родителя и сохранении определенной структуры хромосомы от второго родителя. Для отработки оператора кроссовера выбирается сначала одна точка, затем вторая. К примеру, $P1 = [1, 2 | 3, 4 | 5]$ и $P2 = [1, 3 | 2, 5 | 4]$. Все города, которые расположены между двумя точками основного родителя, передаются в порожденную особь — $C1 = [_, _ | 2, 5 | _]$. Затем используется вторая точка сечения, и элементы второго родителя копируются в том же порядке, исключая те, которые уже находятся в ребёнке —

[1, 3, 4]. Дойдя до конца хромосомы, продолжаем копировать элементы в ребенка из первой ячейки — $C1=[3, 4, 2, 5, 1]$. То же самое и для второго ребёнка — $C2=[2, 5, 3, 4, 1]$. Упорядоченный кроссовер изображен на рис. 2.

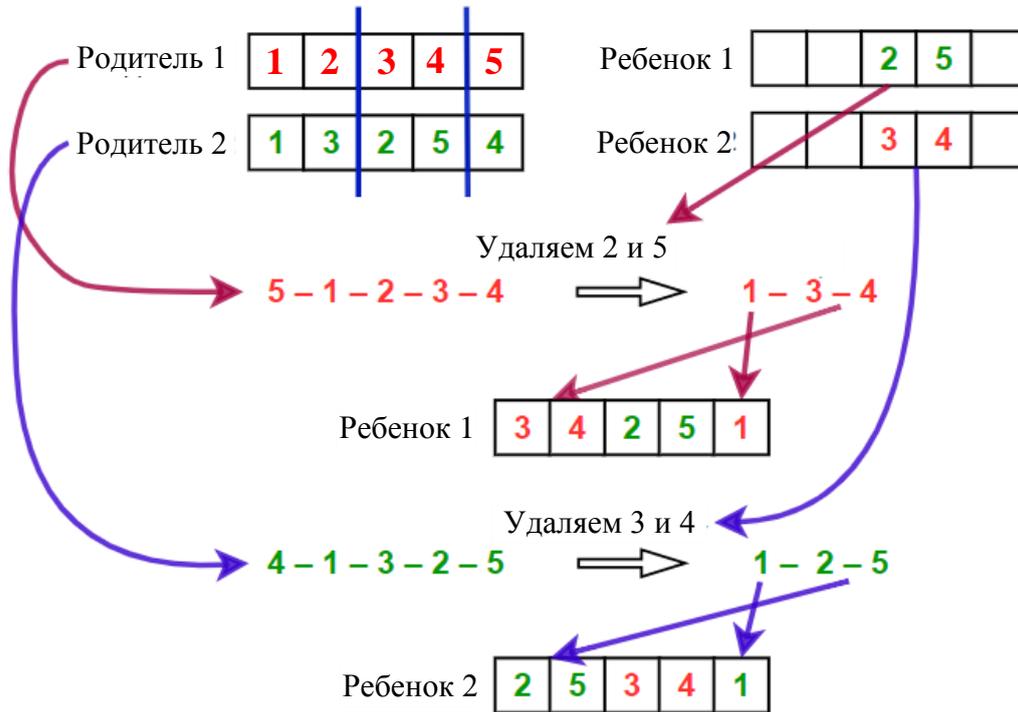


Рис. 2. Упорядоченный кроссовер

Измененный кроссовер работает следующим образом: случайно определяется точка кроссовера, и первая часть первого родителя (P1) переписывается в первого потомка, например $P1=[3,5,2,1,8 | 4,7,6,9]$ и $P2=[2,9,5,3,7 | 1, 4, 8, 6]$, ребенка — $C1=[3,5,2,1,8 | _, _, _, _]$. Для второй части ребенка используются составляющие второго родителя — $C1=[3,5,2,1,8 | 4, 6, _, _]$. Если такие составляющие ранее встречались в потомке, то они не используются, а оставшуюся часть потомка формируют составляющие первого родителя — $C1=[3,5,2,1,8 | 4, 6, 7, 9]$. По такому же принципу получаем второго ребенка — $C2=[2,9,5,3,7 | 4, 6, 1, 8]$. Изменённый кроссовер представлен на рис. 3.

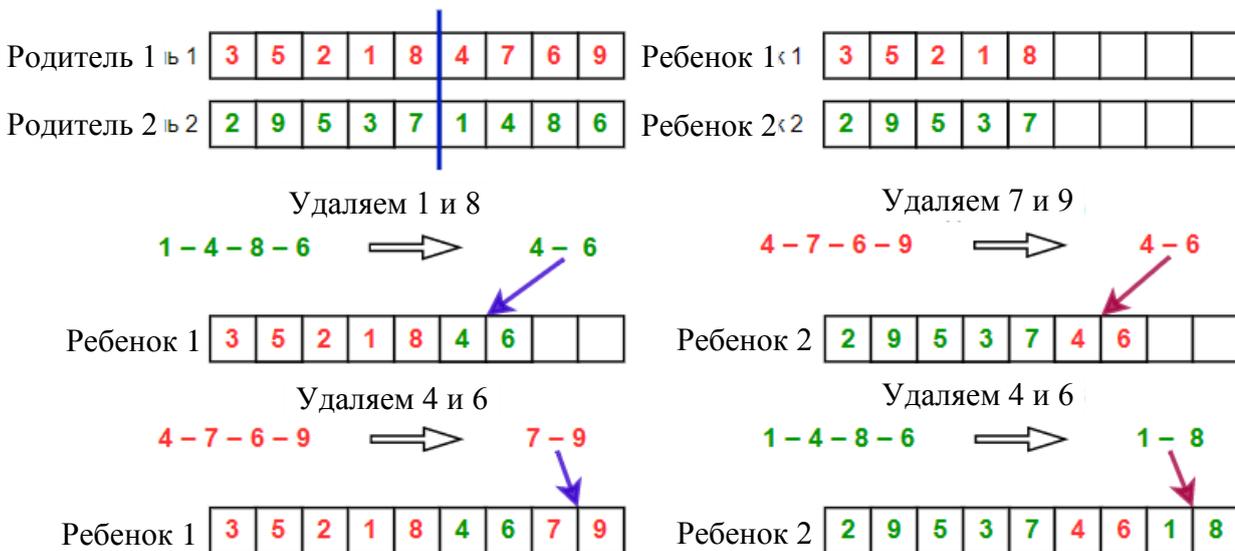


Рис. 3. Изменённый кроссовер

Одноточечная мутация заключается в том, что из хромосомы по какому-то случайному закону выбираются два города и меняются местами. В качестве примера рассмотрим особь $P=[1, 4, 2, 3, 5]$, где первый и третий составляющие должны поменяться местами, в результате получаем особь $P'=[2, 4, 1, 3, 5]$ (рис. 4).

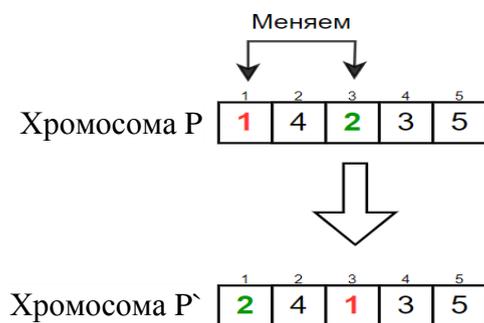


Рис. 4. Одноточечная мутация

Двухточечная непарная мутация состоит в том, что случайно находятся в особи два города, которые немедленно меняются местами с двумя другими случайно выбранными городами — $P=[1, 4, 2, 3, 5]$, первый и третий элементы меняются местами со вторым и четвертым — $P'=[4, 1, 3, 2, 5]$, что показано на рис. 5.

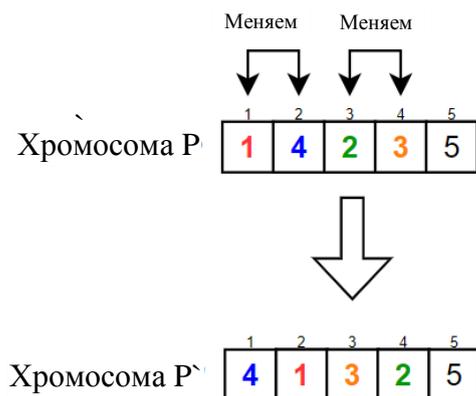


Рис. 5. Двухточечная непарная мутация

Двухточечная парная мутация осуществляется следующим образом: случайно выбирается пара рядом стоящих элементов хромосомы и меняется с другой случайно выбранной парой рядом стоящих элементов. Например, в хромосоме $P=[1, 4, 2, 3, 5]$ пара первого и второго элемента меняется местами с парой четвертого и пятого — $P'=[3, 5, 2, 1, 4]$. Графически данная мутация представлена на рис. 6.

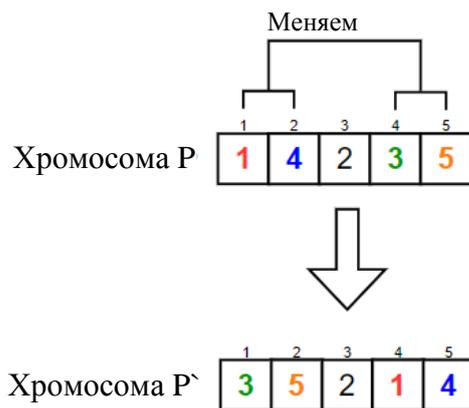


Рис. 6. Двухточечная парная мутация

Вычислительный эксперимент. Аналитически невозможно определить, какой из предложенных кроссоверов дает более точное решение задачи коммивояжера модифицированной моделью Голдберга. Были проведены масштабные вычислительные эксперименты, результаты которых позволили определить процент точных решений относительно количества запусков, а также усредненные значения времени выполнения. Кроме этого, масштабные вычислительные эксперименты позволили оценить эффективность использования модифицированной модели Голдберга при использовании основных операторов кроссовера [8].

Входные данные: n — количество городов, m — количество особей, pc — вероятность кроссовера, pm — вероятность мутации, $stop$ — условие остановки, s — количество запусков. Для решения задачи коммивояжера в виде программы были реализованы упорядоченный и изменённый кроссоверы с тремя вариантами мутаций: одноточечная, двухточечная непарная и двухточечная парная. Для полноты использования алгоритма модифицированной модели Голдберга была выбрана среда разработки Microsoft Visual Studio 2022 с популярным языком программирования C#. Вычислительные эксперименты проводились на компьютере с использованием операционной системы Microsoft Windows 10, процессора Intel i3-6006U 2.0 ГГц с оперативной памятью 4 Гб. Для полноты экспериментов была использована библиотека TSP_LIB, где находятся ранее созданные тестовые графы [9]. Алгоритм имеет следующие входные параметры: n — различно для каждого графа, m — 1000, pc — 1,00, pm — 1,00, $stop$ — 1000, s — 100. В таблицах 1 и 2 представлены конечные результаты машинных экспериментов.

Таблица 1

Результаты вычислительных экспериментов для графов bays29, swiss42 и gr48

Кроссовер	Мутация	Граф bays29		Граф swiss42		Граф gr48	
		точное решение, %	ср. время, сек.	точное решение, %	ср. время, сек.	точное решение, %	ср. время, сек.
Упорядоченный	Одноточечная	15	10,9	49	16,6	13	33,2
	Двухточечная непарная	9	23,6	38	56,5	7	67,8
	Двухточечная парная	22	11,3	51	18,2	9	41,7

Кроссовер	Мутация	Граф bays29		Граф swiss42		Граф gr48	
		точное решение, %	ср. время, сек.	точное решение, %	ср. время, сек.	точное решение, %	ср. время, сек.
Изменённый	Одноточечная	33	8,3	7	12,6	1	15,4
	Двухточечная непарная	24	16,6,7	7	34,9	0	50,1
	Двухточечная парная	15	9,5	4	14,1	0	25,7

Таблица 2

Результаты вычислительных экспериментов для графов berlin52, brazil58 и st70

Кроссовер	Мутация	Граф berlin52		Граф brazil58		Граф st70	
		точное решение, %	ср. время, сек.	точное решение, %	ср. время, сек.	точное решение, %	Ср. время, сек.
Упорядоченный	Одноточечная	21	31,6	10	45,6	0	44,2
	Двухточечная непарная	10	71,6	0	106,1	0	104,2
	Двухточечная парная	15	45,2	1	65,3	0	69,8
Изменённый	Одноточечная	1	17,1	0	23,8	0	35,5
	Двухточечная непарная	0	44,3	0	60,7	0	89,8
	Двухточечная парная	0	22,9	0	30,1	0	45,6

Заключение. Анализ результатов, полученных в процессе проведенных масштабных экспериментов, продемонстрировал эффективность работы генетического алгоритма при решении нахождения гамильтонова пути. Также было установлено, что с увеличением числа вершин упорядоченный кроссовер с одноточечной мутацией имеет наименьший процент отклонений от точных решений, но уступает в среднем по времени выполнения любой комбинации изменённого кроссовера с используемыми мутациями. Исходя из того, что процент получения точного решения является более значимым показателем при сравнении, чем среднее время выполнения, был сделан вывод, что использование комбинации упорядоченного кроссовера и одноточечной мутации эффективнее, чем любая комбинация изменённого кроссовера с используемыми мутациями. В то же время упорядоченный кроссовер с использованием других мутаций может быть не таким эффективным.

Библиографический список

1. Задача коммивояжера / Studfiles : [сайт]. — URL: <https://studfile.net/preview/987700/> (дата обращения: 21.01.2023).
2. Гладков, Л. А. Генетические алгоритмы : учеб. пособие / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик. — Ростов-на-Дону : Ростиздат, 2004. — 400 с.
3. NP-полные и NP-трудные задачи / Studfiles : [сайт]. — URL: <https://studfile.net/preview/5562531/page:42/> (дата обращения: 21.01.2023).
4. Трансвычислительная задача / Studwood : [сайт]. — URL: https://studwood.net/1788462/informatika/transvychislitel'naya_zadacha (дата обращения: 21.01.2023).
5. Емельянов, В. В. Теория и практика эволюционного моделирования / В. В. Емельянов, В. В. Курейчик, В. М. Курейчик. — Москва : Физматлит, 2003. — 432 с.
6. Оптимизация (математика) / Википедия : [сайт]. — URL: [https://ru.wikipedia.org/wiki/Оптимизация_\(математика\)](https://ru.wikipedia.org/wiki/Оптимизация_(математика)) (дата обращения: 22.01.2023).
7. Каширина, И. Л. Введение в эволюционное моделирование : учебное пособие / И. Л. Каширина ; под ред. Е. С. Котляровой. — Воронеж : Изд-во ВГУ, 2007. — 40 с.
8. Оператор кроссинговера / Studfiles : [сайт]. — URL: <https://studfile.net/preview/8889281/page:5/> (дата обращения: 22.01.2023).
9. TSP_LIB / Ruprecht-Karls-Universitat Heidelberg : [сайт]. — URL: <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/> (дата обращения: 22.01.2023).

Об авторах:

Кобак Валерий Григорьевич, профессор кафедры «Программное обеспечение вычислительной техники и автоматизированных систем» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), доктор технических наук, valera33305@mail.ru

Рязанов Александр Андреевич, магистрант кафедры «Программное обеспечение вычислительной техники и автоматизированных систем» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), ryazanov.alexandr@yandex.ru

About the Authors:

Valeriy G. Kobak, professor of the Computer Engineering and Automated Systems Software Department, Don State Technical University (1, Gagarina sq., Rostov-on-Don, RF, 344003), Dr. Sci. (Eng.), valera33305@mail.ru

Aleksandr A. Ryazanov, Master's degree student of the Computer Engineering and Automated Systems Software Department, Don State Technical University (1, Gagarina sq., Rostov-on-Don, RF, 344003), ryazanov.alexandr@yandex.ru