

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.8

А.В.Протодьяконов, А.Н. Фомин, С.Е. Швец, Е.В. Марьенков

ВЛИЯНИЕ ПАРАМЕТРОВ ГЕНЕТИЧЕСКОГО АЛГОРИТМА НА РЕЗУЛЬТАТЫ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА

Задача коммивояжера – одна из наиболее популярных задач комбинаторной оптимизации, в простейшем варианте сводящаяся к поиску замкнутого маршрута минимальной длины, проходящего через все указанные города только по одному разу. Задача коммивояжера используется при составлении транспортных маршрутов, для управления автоматами, в кристаллографии, при планировании информационных сетей, при составлении расписаний и т. д. [2, 6, 8].

За последнее время разработано множество алгоритмов решения этой задачи, как точных, так и приближенных. В отличие от большинства из них, генетические алгоритмы (ГА), основанные на некоторых формализованных принципах естественного эволюционного процесса развития живых организмов [1, 3, 4], позволяют относительно просто и быстро решать подобные задачи.

Цель данной работы – изучить влияние параметров генетического алгоритма на результат решения задачи коммивояжера.



Рис. 1. Общая схема генетического алгоритма

Основные принципы генетического алгоритма предложил Джон Холланд в начале 60-х гг. Всеобщее признание генетические алгоритмы получили после выхода в 1975 г. его книги «Адаптация в естественных и искусственных системах». Генетический алгоритм был получен в процессе обобщения и имитации в искусственных системах таких свойств живой природы, как естественных отбор, приспособляемость к изменяющимся усло-

виям среды, наследование и изменчивость [6, 7, 9].

Общая схема работы генетического алгоритма представлена на рис. 1. В генетическом алгоритме популяция соответствует множеству рассматриваемых решений, а особь – конкретному решению.

При создании ГА рассматриваются следующие основные аспекты:

- схема кодирования решений;
- определение функции полезности особи (решения);
- метод выбора родителей;
- операторы кроссовера и мутации;
- схема редукции.

Схема кодирования решений

Каждое решение кодируется в некоторую последовательность, и генетический алгоритм оперирует полученной кодовой последовательностью безотносительно ее смыслового содержания. Кодовые последовательности называются хромосомами. Способов кодирования очень много, но наиболее часто применяют кодирование в виде битовых строк и числовых векторов. При решении задачи коммивояжера удобно представлять решения в виде векторов целых чисел, где значение элемента соответствует номеру города, а последовательность посещения городов определяется последовательностью чисел в векторе.

Вычисление функции полезности

Качество конкретного решения определяется функцией полезности. Существует два основных подхода к вычислению функции полезности: скейлинг и ранжирование.

Скейлинг является более традиционным подходом, но применим только для одноцелевых задач. При таком подходе функция полезности монотонна и связана с целевой функцией. В нашем случае целевой функцией является длина пути конкретного решения. Например, при решении задачи коммивояжера возможны следующие зависимости:

$$z_i = 1.1d_{max} - d_i$$

или

$$z_i = \exp(-d_i / d_{max}),$$

где z_i – функция полезности i -ой особи, d_i – длина пути i -ой особи, d_{max} – максимальная длина пути в

текущей популяции.

Ранжирование основано на упорядочивании (ранжировании) особей, по значению целевой функции. В этом случае значение функции полезности зависит от положения (ранга) особи в упорядоченном множестве. Могут применяться следующие формулы:

$$z_i = 1 - \frac{r_i}{K-1} \text{ или } z_i = \rho^{r_i} \quad 0 < \rho < 1,$$

где r_i - ранг i -той особи, K – размер популяции.

Методы выбора родителей

Чтобы создать хорошее потомство, важно отобрать хороших родителей, поэтому необходимо определить метод выбора родителей. Чаще всего используются методы колеса рулетки и парных турниров.

Метод колеса рулетки включает в себя несколько запусков, вероятность выбора каждой особи пропорциональна функции полезности и вычисляется по формуле

$$p_i = \frac{z_i}{\sum_{j=0}^K z_j},$$

где z_i – функция полезности i -ой особи, K – количество особей в популяции.

При методе парных турниров случайным образом выбираются две особи из популяции и сравниваются между собой. Вероятность выбора каждой особи пропорциональна функции полезности и вычисляется по формуле

$$p_1 = \frac{z_1}{z_1 + z_2},$$

где z_1 и z_2 – соответственно функции полезности первого и второго кандидатов на выбор.

Кроссовер

Для получения новых комбинаций генов необходимо использование операторов, которые вносили бы некоторый элемент случайности и новизны. Эти операторы можно разделить на две группы: кроссовер и мутация. Реализация этих операторов в значительной степени зависит от способа кодирования решений.

При кроссовере особи обмениваются частью своих генов. Условно это изображено на рис. 2.

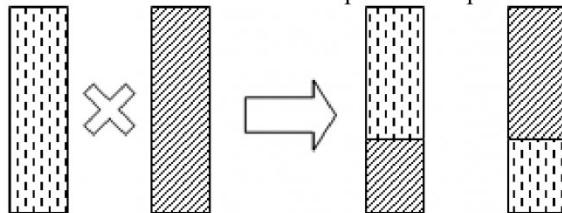


Рис. 2. Общая схема кроссовера

Кроссовер соответствует скрещиванию у живых организмов. В результате кроссовера получа-

ется особь, которая обладает признаками обоих родителей.

Существует множество алгоритмов кроссовера. В нашей работе мы использовали следующие алгоритмы:

- частично отображаемый кроссовер;
- упорядоченный кроссовер;
- кроссовер рекомбинации ребер;
- измененный кроссовер.

Мутация

В большинстве случаев применение одного лишь кроссовера недостаточно, т. к. в результате скрещивания потомок не может получить признак, которого нет у родителей. Поэтому вводится оператор мутации, который вносит дополнительное разнообразие в генотип особей. Обычно вероятность применения оператора мутации невелика.

В нашей работе используются следующие алгоритмы мутации:

- точечная мутация;
- мутация перемещения;
- жадная мутация.

Редукция

После получения достаточного количества потомков их необходимо ввести в существующую популяцию. Чтобы избежать постепенного увеличения размера популяции, часть особей удаляют. В простейшем случае популяция целиком заменяется особями из нового поколения. Но часто используются алгоритмы, в которых потомки конкурируют с родителями за право оставаться в популяции. Обычно потомки могут замещать

- своих родителей;
- худшие особи в популяции;
- самые старые особи в популяции;
- случайные особи в популяции.

В то же время эта замена может происходить

- безусловно;
- только если потомки лучше замещаемых особей;
- вероятностно, в зависимости от соотношения функций полезности потомков и замещаемых особей.

В нашей работе используется стратегия элитизма. То есть редукции подлежат лишь худшие особи популяции, а лучшие всегда переходят в следующее поколение. Уровень элитизма определяет процентное количество особей старой популяции, переходящих в новое поколение.

Результаты

Для проведения исследований была создана программа, реализующая генетический алгоритм для решения задачи коммивояжера. В качестве исходных данных была выбрана задача st70 из библиотеки TSPlib [5]. Эта задача представляет собой список из семидесяти городов с указанием их координат. Для этой задачи известно точное решение – 675 единиц длины

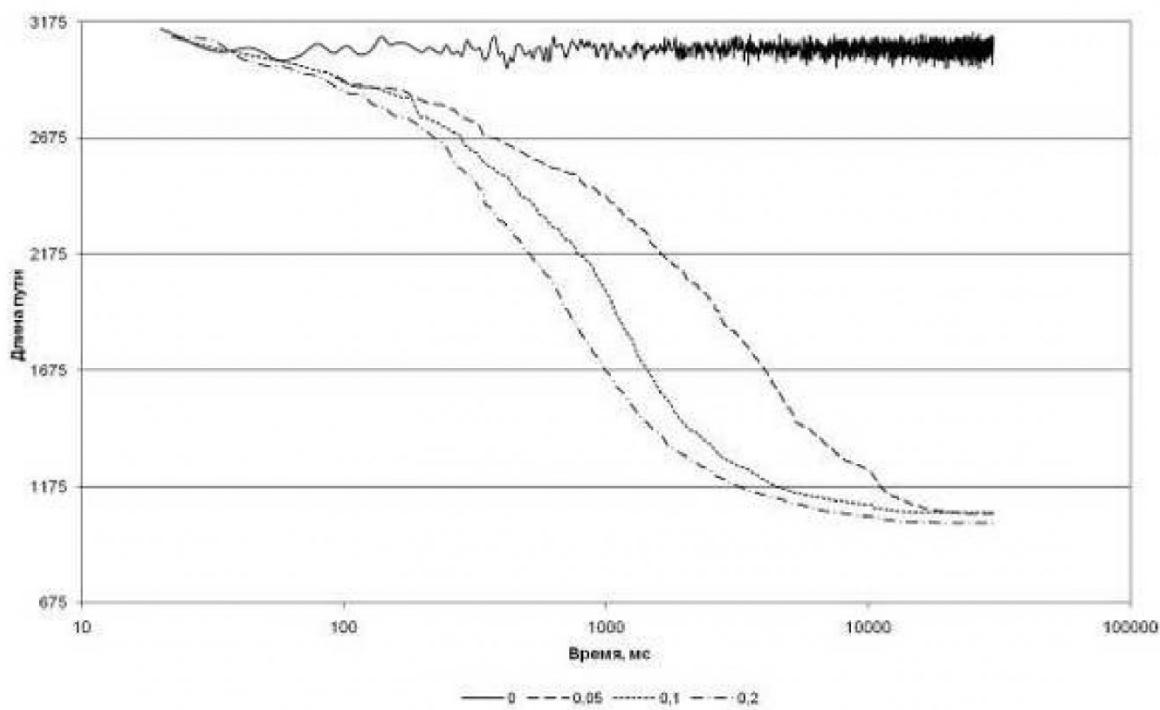


Рис. 3. Влияние уровня элитизма на результат

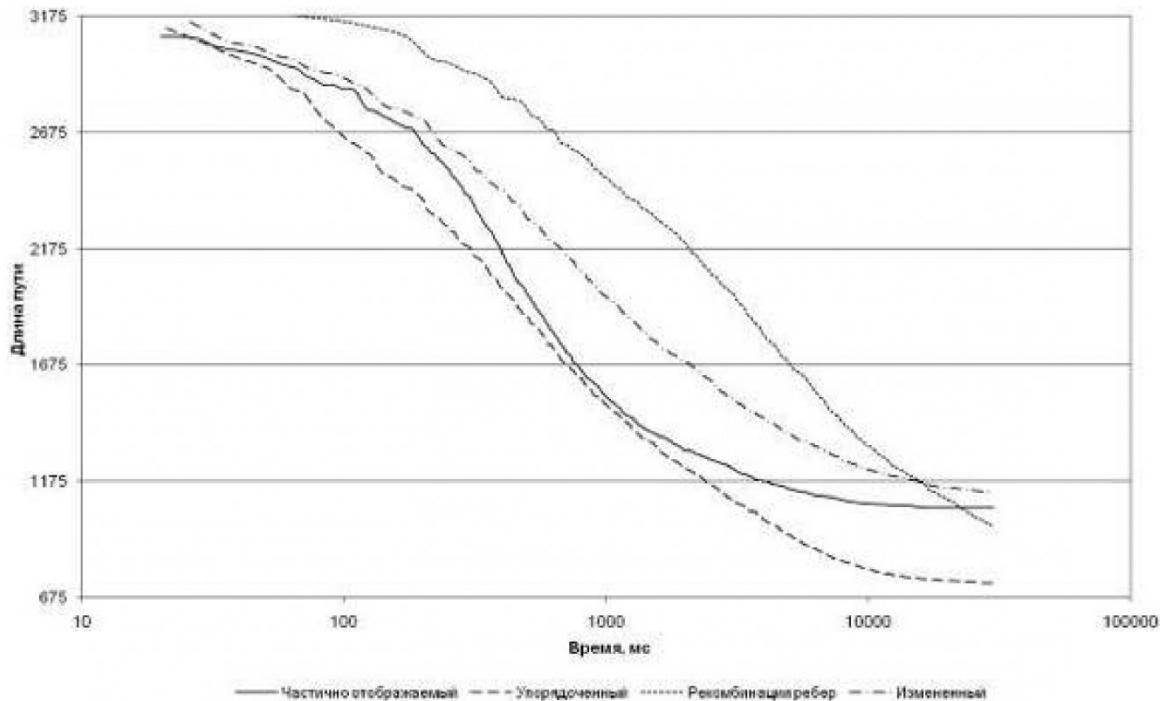


Рис. 4. Влияние вида кроссовера на результат

Были выбраны следующие изменяемые параметры:

- уровень элитизма;
- вид кроссовера;
- метод выбора родителей;
- метод вычисления функции полезности;
- вероятность точечной мутации;
- вероятность мутации перемещения;
- вероятность жадной мутации;
- количество особей в популяции.

Исследование каждого параметра проводилось в указанной выше последовательности.

Так как генетический алгоритм содержит в себе элементы случайности, то в качестве выходных значений используется средний путь за десять прогонов. Время работы каждого прогона ограничивалось 30 секундами.

Здесь приведены графики, показывающие влияние различных параметров на скорость получения и качество решений.

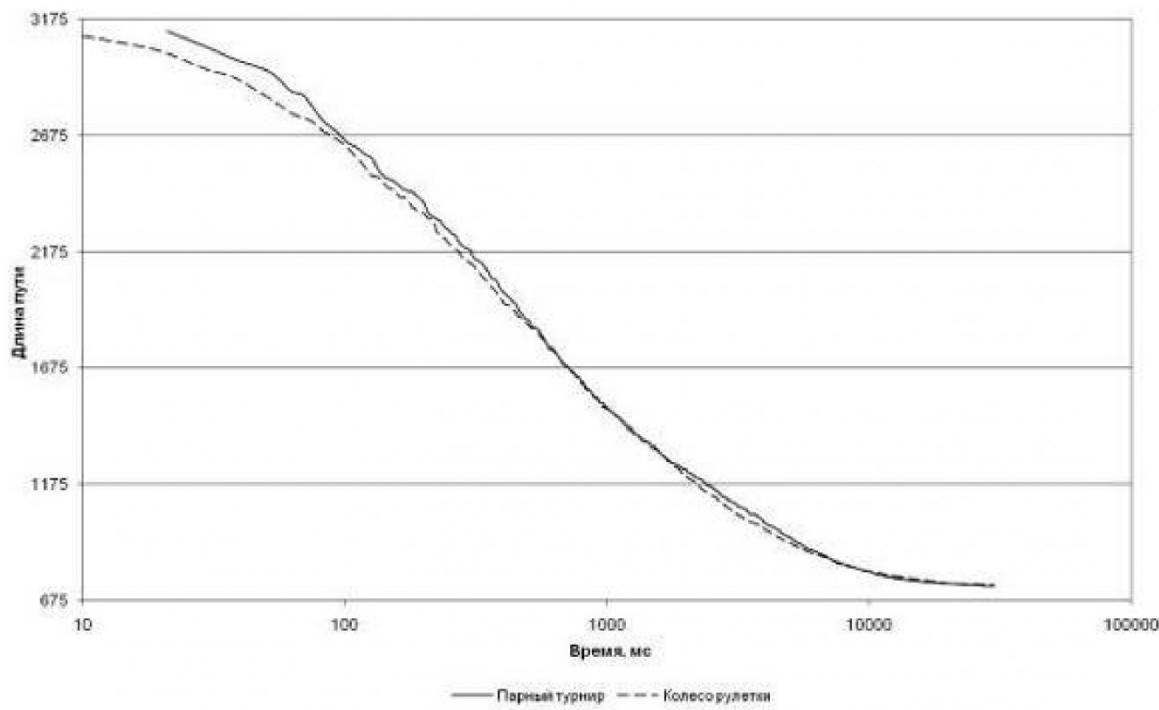


Рис. 5. Влияние метода выбора родителей на результат

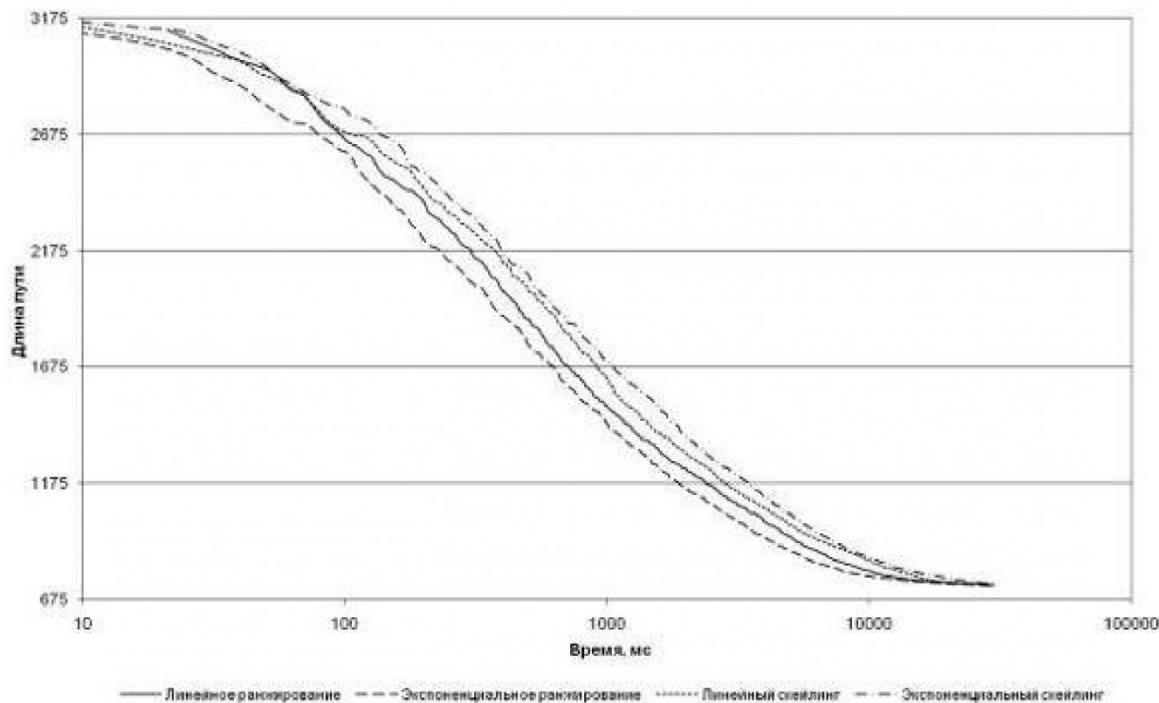


Рис. 6. Влияние способа вычисления функции полезности

Как видно из рис.3, элитизм благоприятно влияет на работу генетического алгоритма. При значении уровня элитизма более 0,2 улучшения качества и скорости поиска решений не наблюдается.

Можно отметить, что чем больше уровень элитизма, тем большее количество поколений успевает смениться за то же время.

Например, за 30 секунд при значении 0,2 успевает смениться порядка 3000 поколений, а при

0,5 – порядка 5000.

Из рис. 4 видно, что наилучшие результаты на всем временному промежутке показал упорядоченный кроссовер. Но можно заметить, что кривая кроссовера рекомбинации ребер не вышла на плато, поэтому можно ожидать от этого вида кроссовера еще лучшего результата при увеличении времени.

Как видно из рис. 5, метод выбора родителей не оказывает значительного влияния на результат.

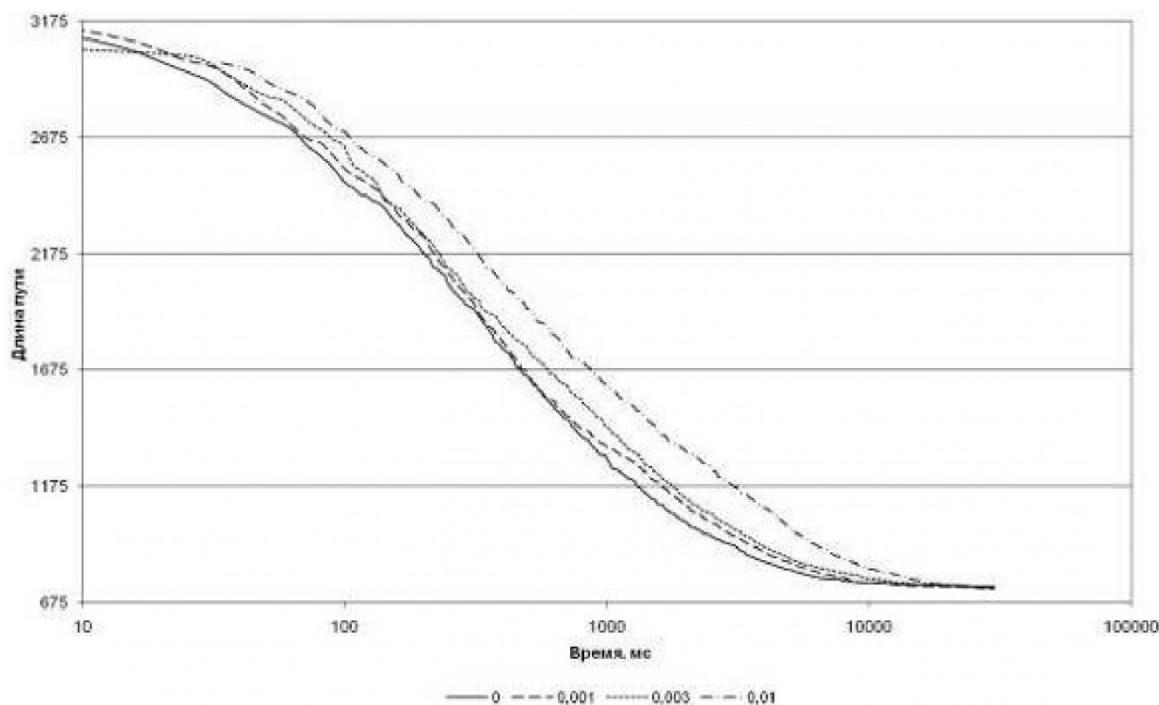


Рис. 7. Влияние уровня точечной мутации на результат при упорядоченном кроссовере

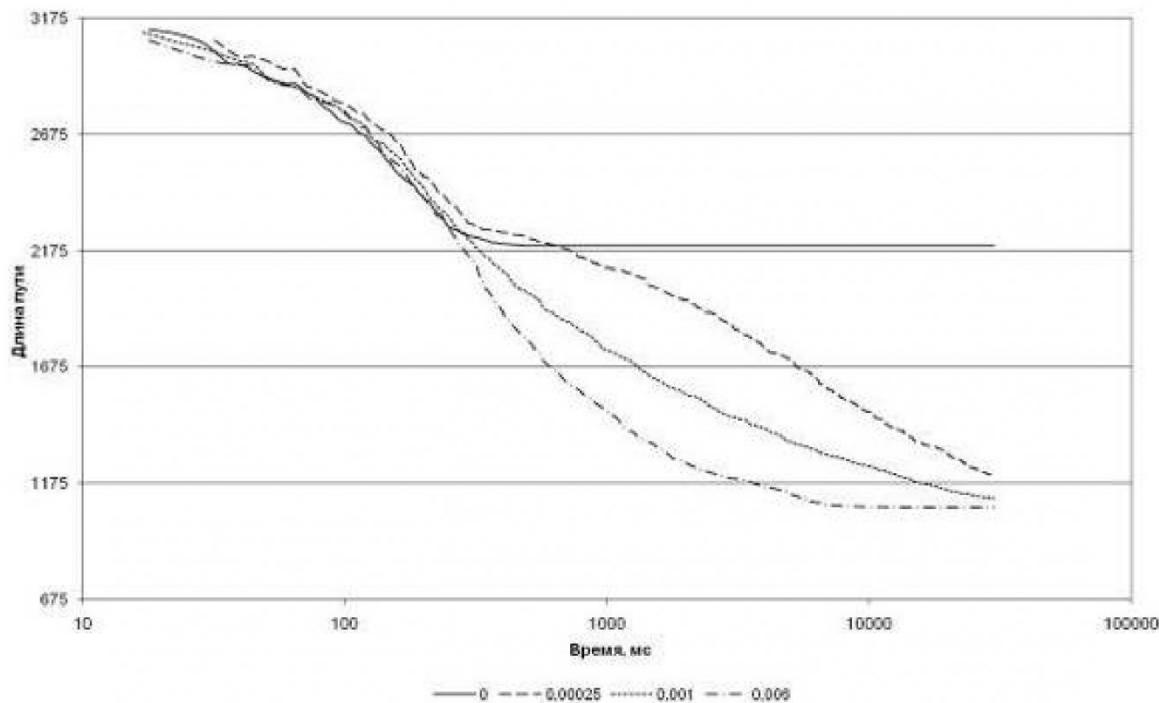


Рис. 8. Влияние уровня точечной мутации на результат при частично отображаемом кроссовере

Нет значительной разницы между различными способами вычисления функции полезности. Наилучшие результаты показало экспоненциальное ранжирование, худшие – экспоненциальный скейлинг (рис.6). В ходе экспериментов выяснилось, что действие различных видов мутаций в значительной степени зависит от вида используемого кроссовера.

Из рис. 7 и 8 видно, что при упорядоченном кроссовере точечная мутация мешает, а при час-

тически отображаемом – помогает достичь лучший результат.

Как показали эксперименты, различные виды мутации позволяют улучшить результаты только частично отображаемого кроссовера, поэтому для других видов мутаций приведены графики только с этим видом кроссовера.

Применение мутации перемещения при частично отображаемом кроссовере способно улучшить результаты.

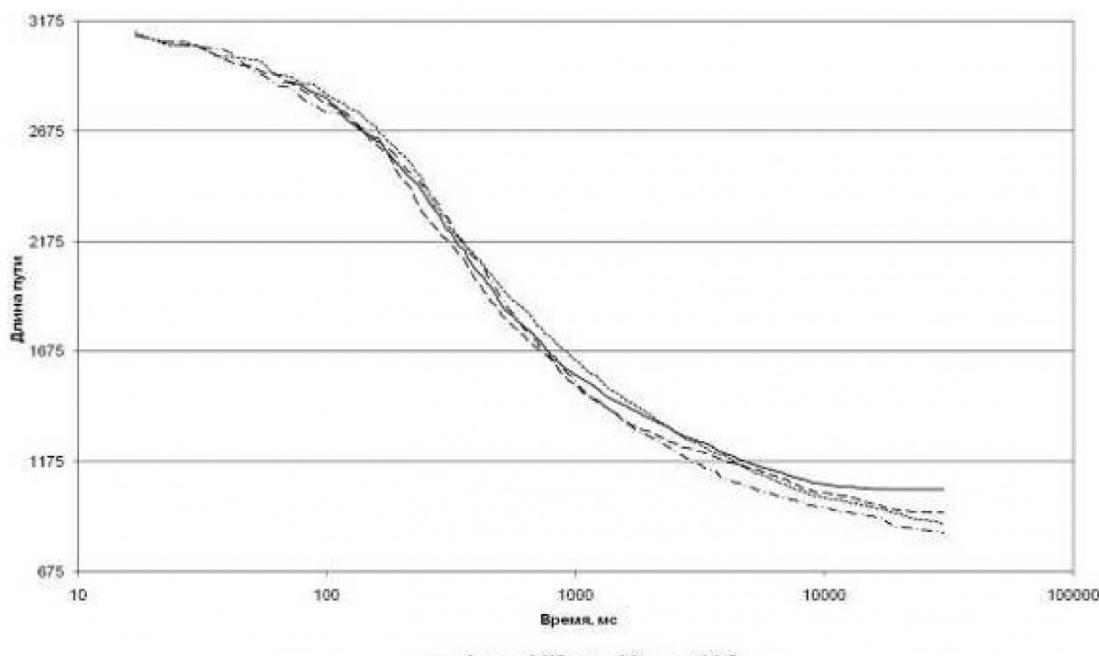


Рис. 9. Влияние уровня мутации перемещения на результат при частично отображаемом кроссовере

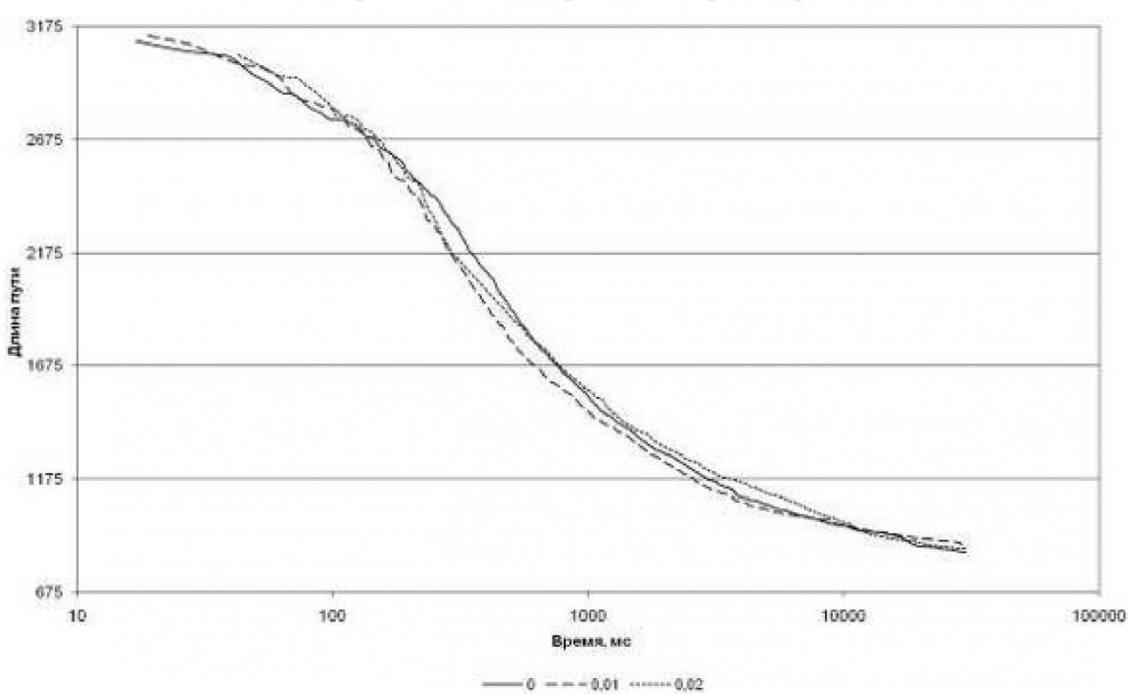


Рис. 10. Влияние уровня жадной мутации на результат при частично отображаемом кроссовере

Как видно из рис.10, жадная мутация не оказывает значительного влияния на результаты экспериментов.

Из рис.11 видно, что с увеличением количества особей в популяции скорость схождения уменьшается, а качество решений улучшается.

По приведенным выше результатам был проведен дополнительный эксперимент, цель которого – получить лучшее решение.

Были использованы следующие значения параметров: время вычислений – 100 с, количество

особей в популяции – 1000, уровень элитизма – 0,5, кроссовер – упорядоченный, способ вычисления функции полезности – экспоненциальное ранжирование, мутации отсутствуют. Найденное минимальное значение пути - 684,23. Отклонение найденного решения от оптимального-1,4%.

Выводы

В ходе выполнения данной работы с помощью программы реализации генетического алгоритма проведено исследование влияния его параметров на результаты решения. Было установлено, что

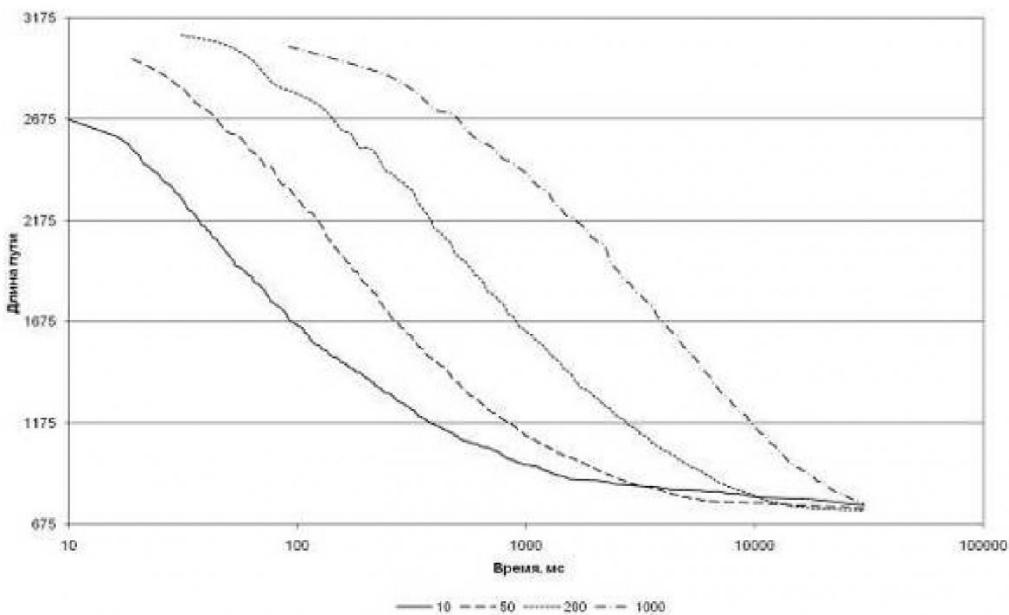


Рис. 11. Влияние количества особей в популяции на результат

основным параметром, который определяет работу всего алгоритма, является вид кроссовера. Лучшие результаты показал упорядоченный кроссовер. Также важен такой параметр как уровень элитизма, точнее важно само наличие или отсутствие элитизма. Введение стратегии элитизма позволяет значительно улучшить результаты работы генетического алгоритма. Такие параметры как

метод выбора родителей и способ вычисления целевой функции не оказали существенного влияния на результаты. Влияние параметров мутации зависит от вида кроссовера. При упорядоченном кроссовере введение мутации не улучшило решений, а при частично-отображаемом – мутация положительно влияет на результаты.

СПИСОК ЛИТЕРАТУРЫ

1. Ozcan, E. A Brief Review of Memetic Algorithms for Solving Euclidean 2D Traveling Salesrep Problem / Ender Ozcan, Murat Erenturk – Department of Computer Engineering, Yeditepe University, Istanbul, Turkey. – http://cse.yeditepe.edu.tr/~eozcan/research/papers/TSP_id135final.pdf
2. Hoffman, K. Traveling Salesman Problem / Karla Hoffman, Manfred Padberg – http://iris.gmu.edu/~khoffman/papers/trav_salesman.html
3. Pilat, M. L. Using Genetic Algorithms to optimize ACS-TSP / Marcin L. Pilat, Tony White – School of Computer Science, Carleton University, Ottawa, Canada. – <http://www.scs.carleton.ca/~arpwhite/documents/ANTS2002.pdf>
4. Catterjee, S. Genetic algorithms and traveling salesman problem / Sangit Catterjee, Cecilia Carrera, Lucy A. Lynch // European Journal of Operational Research, 1996. – Vol. 93. – Issue 3. – Pages 490-510
5. TSPLib – <http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsp/>
6. Назаров, А. В. Нейросетевые алгоритмы прогнозирования и оптимизации систем / А. В. Назаров, А. И. Лоскутов – СПб.: Наука и Техника, 2003. – 384 с.
7. Weil, Y. Parallel Search Strategies for TSPs using a Greedy Genetic Algorithm / Yingzi Wei, Yulan Hu, Kan-feng Gu // hird International Conference on Natural Computation (ICNC 2007), 2007. - vol. 3. – pages 786-790 – <http://intechweb.org/downloadpdf.php?id=5841>
8. Johnson D. S. The Traveling Salesman Problem: A Case Study in Local Optimization / David S. Johnson, Lyle A. McGeoch - <http://www.cs.uwaterloo.ca/~msemamit/TSPCaseStudybyJohnson.pdf>
9. Darrell Whitley. A Genetic Algorithm Tutorial - Computer Science Department Colorado State University, Fort Collins, USA, 1995. –http://www.cs.iastate.edu/~honavar/ga_tutorial.pdf

□ Авторы статьи:

Протодьяконов Андрей Владимирович - канд. техн. наук, доц. каф. информационных и автома- тизованных производст- венных систем КузГТУ. Email: protod@inbox.ru	Фомин Андрей Николаевич - соискатель каф. информа- ционных и автоматизирован- ных производственных систем КузГТУ. Email: fomin-an@mail.ru	Швец Сергей Евгеньевич - студент гр. ИТ-041 КузГТУ. Email: justserega@gmail.com	Марьенков Евгений Валерьевич - студент гр. ИТ-042 КузГТУ. Email: kemdeveloper@gmail.com
---	---	---	---