

*Сухов И.В. Анализ алгоритма имитации отжига в решении задачи коммивояжёра // Академия педагогических идей «Новация». Серия: Научный поиск. – 2019. – №6 (июнь). – АРТ 37-эл. – 0,3 п.л. - URL: <http://akademnova.ru/series-scientific-search>*

**РУБРИКА: ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ**

**УДК 519.87**

**Сухов Илья Владимирович,**

студент 2 курса, факультет информатика и вычислительная техника  
ФГБОУ ВО «Донской государственный технический университет»,  
г. Ростов-на-Дону, Российская Федерация

e-mail: [neron.warcraft@rambler.ru](mailto:neron.warcraft@rambler.ru)

**АНАЛИЗ АЛГОРИТМА ИМИТАЦИИ ОТЖИГА В РЕШЕНИИ  
ЗАДАЧИ КОММИВОЯЖЁРА**

*Аннотация:* В статье рассмотрено решение задачи коммивояжёра с применением алгоритма имитации отжига, а также способы оптимизации алгоритма.

*Ключевые слова:* алгоритм имитации отжига, задачи маршрутизации, задача коммивояжёра, параллельная реализация, оптимизация алгоритма.

**Sukhov Ilya Vladimirovich,**

2nd year student, Computer Science and Computer Engineering Faculty  
FSBEI of HE "Don State Technical University",

Rostov-on-Don, Russian Federation

e-mail: [neron.warcraft@rambler.ru](mailto:neron.warcraft@rambler.ru)

## ANALYSIS OF THE ALGORITHM OF IMITATION OF ANTIC IN THE SOLUTION OF THE COMMUNICATION PROBLEM

*Abstract:* The article describes the solution of the problem of the algorithm of simulation of the algorithm.

*Keywords:* annealing simulation algorithm, routing problems, traveling salesman problem, sequential and parallel implementation, algorithm optimization.

### **Алгоритма имитации отжига**

Алгоритм имитации отжига (англ. Simulated annealing) — общий алгоритмический метод решения задачи глобальной оптимизации, особенно дискретной и комбинаторной оптимизации. Один из примеров методов Монте-Карло.

В применении к задаче коммивояжёра метод имитации отжига формируется следующим образом: множество всех точек-городов обозначается как  $C$ , общее количество обозначается  $|C|$ . Каждый город представляет собой спаренные координаты, содержащие внутри себя индексы.

По условию задачи коммивояжёра, его решением будет являться маршрут между всеми городами. Из этого следует, что множество состояний  $S$  — все возможные маршруты, проходящие через каждый город. Другими словами, множество всех упорядоченных последовательностей элементов  $C$ , в которой каждый элемент встречается ровно один раз. Следовательно, длина каждой такой последовательности  $|C|$ .

Для того чтобы использовать метод имитации отжига необходимо обозначить две функции, которые зависят от конкретной задачи в каждом отдельном случае. Это функция энергии  $E$  (или «целевая функция») и функция  $F$ , которая отвечает за порождение новых состояний.

$$E(s_i) = \left[ \sum_1^{|C|-1} \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \right] + \sqrt{(x_{|C|} - x_1)^2 + (y_{|C|} - y_1)^2}$$

Здесь представлена сумма Евклидовых расстояний между парами городов в маршруте  $s_i$ .

Одним из вариантов для генерации новых состояний в задаче является перестановка двух произвольных городов в маршруте. Недостатком такой идеи является изменение городов в маршруте. Оно будет непредсказуемо влиять на  $E_i$ , что в конечном итоге исказит результат в негативную сторону. Хорошим аналогом в таком случае будет выбор двух произвольных городов в маршруте, и проведения инвертирования путей между ними. Например, у нас был маршрут (1, 2, 3, 4, 5, 6, 7). Генератор случайных чисел выбрал города 2 и 7, мы выполнили процедуру и получили (1, 7, 6, 5, 4, 3, 2).

Существует два типа реализации метода имитации отжига — последовательная и параллельная. В статье используется метод имитации отжига с применением параллельной реализации.

Параллельная реализация: сгенерируем произвольный начальный путь, содержащий все вершины по одному разу и возвращающийся в начальную позицию и «раздадим» его каждому из параллельно работающих процессов/потоков. Каждый из них, независимо друг от друга, производит описанные операции по поиску оптимального пути и возвращает найденный

результат. Главный поток, если таковой имеется, выбирает из полученных результатов лучшее значение.

Параллельный вариант достигает простоты за счёт того, что не происходит борьбы за общие ресурсы и отсутствует обмен информацией между самими потоками, что на практике упрощает процесс программирования.

### Оптимизация отжига в решении TSP

Каждый из параллельных процессов начинает работу с одинаковыми начальными условиями, вероятность найти оптимальное решение для каждого из них будет одинаковой. Примем ее за  $p$ . Тогда вероятность не найти такое решение будет равна  $q = 1 - p$ . Таким образом, вероятность  $P$  того, что хотя бы один из процессов найдет оптимальный путь равна:

$$P(K) = 1 - q^k = 1 - (1 - p)^k$$

Где  $K$  — количество параллельно работающих процессов. Такая зависимость очень быстро стремится к 1 даже при небольших  $p$ . В соответствии с рисунком 1 показан график зависимости  $P(K)$  для  $p = 0,1$ :

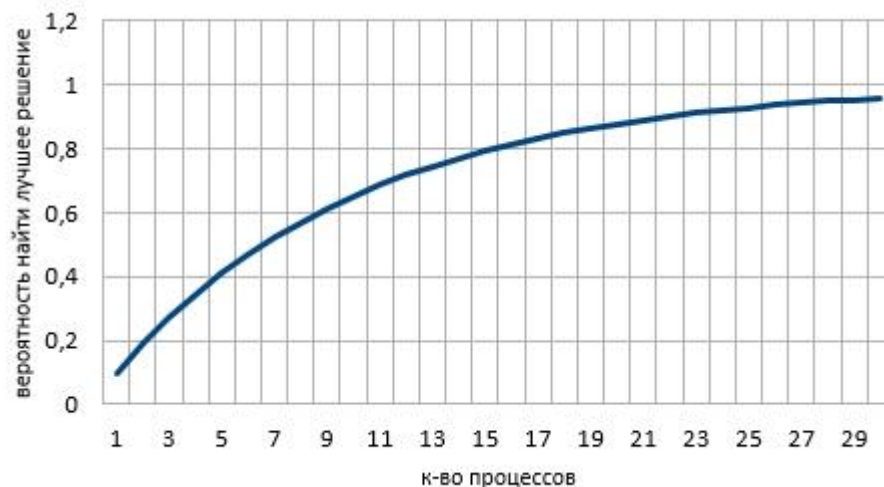


Рисунок 1 — график зависимости  $P(K)$  для  $p = 0,1$

При  $K = 6..7$  вероятность  $P$  становится равной 0,5. Из этого следует, что польза от использования нескольких процессов позволяет добиться увеличения вероятности.

Из этого следует, что для оценки искомой по заданию вероятности необходимо провести не один эксперимент, а комплекс. Для статьи была проведена серия испытаний, отличных параметром  $N$ . Ограничения счетчика —  $F = N^2$ . Для каждого  $N$  было проведено 10000 испытаний. Полученные результаты занесены в таблицу 1:

Таблица 1 — результаты работы алгоритма имитации отжига

| $N$ | $P_{min}$ | $P(1\%)$ | $P(5\%)$ | $P(10\%)$ | $P(15\%)$ |
|-----|-----------|----------|----------|-----------|-----------|
| 5   | 0,995     | 0,995    | 0,995    | 0,995     | 0,995     |
| 10  | 0,0381    | 0,1548   | 0,2801   | 0,5382    | 0,7962    |
| 20  | 0,0001    | 0,0003   | 0,0273   | 0,3132    | 0,8062    |
| 50  | 0,0002    | 0,0005   | 0,2267   | 0,9858    | 1         |
| 100 | 0,0001    | 0,0008   | 0,9003   | 1         | 1         |

$P(min)$  – вероятность найти оптимальное решение.

$P(X\%)$  – вероятность найти решение, отличающее от оптимального не более чем на  $X\%$ .

Подобный анализ не следует считать точным, т. к. даже 10000 экспериментов не позволят получить закономерности внутри алгоритма. Так же нет подтверждений, что алгоритм добился оптимального решения. Так или иначе, возможно сделать вывод о том, что полученные результаты могут быть применены для качественного понимания работы алгоритма.

Проиллюстрируем эти же результаты в соответствии с рисунком 13:

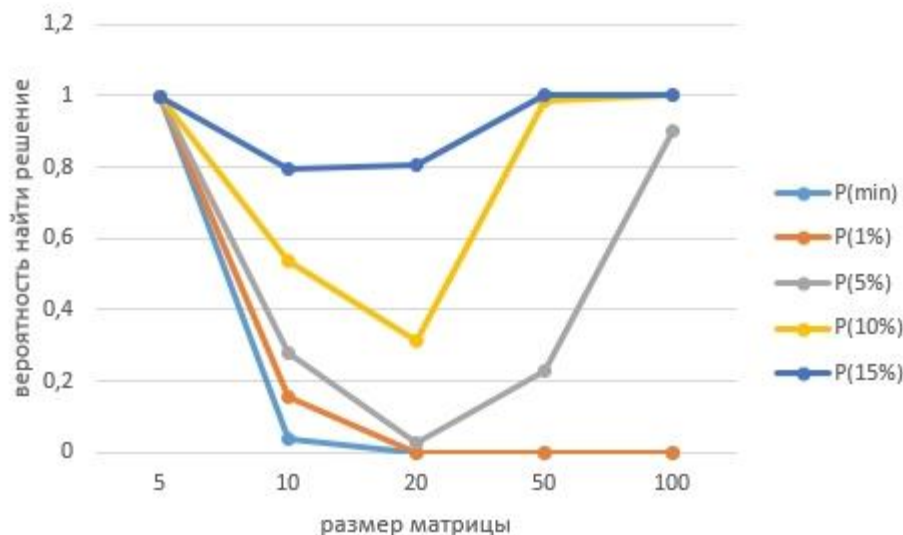


Рисунок 1 — результаты алгоритма имитации отжига

#### Достоинства метода:

- простота логики алгоритма;
- простота технической реализации;
- повышение эффективности алгоритма при параллельной структуре;
- высокая вероятность нахождения близкого к оптимальному решения.

#### Недостатки метода:

- быстро уменьшающаяся вероятность нахождения точного решения с увеличением параметра  $N$ ;
- сложность точной оценки вероятности нахождения решения;
- отсутствие взаимосвязи между потоками, что может привести к закликиванию в локальных минимумах.

#### Оптимизация алгоритма:

- добавление хранилища уже проверенных путей. В представленной реализации алгоритм не способен запоминать проверенные им пути, и никак

не контролирует получаемые на каждой итерации новые варианты. В связи с этим, возможно застревание результатов в локальном минимуме. Добавление функции проверки таких ситуаций положительно скажется на точности результатов;

- более неоднозначный способ модификации — подстройка контрольного значения счетчика. Из проверочных графиков видно, что при текущих настройках  $F = N^2$  появляется пробел в районе матриц размера  $N = 20$ . Подбор более удачного значения, вероятно, позволит улучшить результаты.

### **Заключение**

В данной статье был рассмотрен алгоритм имитации отжига в применении к решению транспортных задач, в частности к задаче коммивояжёра. Результатом исследования стал вывод о том, что данный алгоритм имеет широкие перспективы для развития с учётом возможности параллелизации алгоритма, а также простой технической реализации и формировании алгоритма. Последнее позволит упростить тестирование, расширив число потенциальных разработчиков.

Методы оптимизации, представленные в исследовании, позволят улучшить работу алгоритма в ситуации с большим числом точек, что также улучшит конкурентоспособность имитации отжига в вопросах решения транспортных задач.

### **Список использованной литературы:**

1. Каллан Роберт. Основные концепции нейронных сетей. — М.: Издательский дом Вильямс, 2003. — 288 с. — ISBN 5-8459-0219-X. — С. 146—148.
2. Кирсанов М. Н. Графы в Maple. — М.: Физматлит, 2007. — 168 с. — ISBN 978-5-9221-0745-7. — С. 151—154.

3. Джонс М. Т. Программирование искусственного интеллекта в приложениях. — М.: ДМК Пресс, 2004. — 312 с. — ISBN 5-94074-275-0. — С. 25—42.
4. В.И. Мудров. Задача о коммивояжёре. — М.: «Знание», 1969. — С. 62.
5. Aarts E. H. L., Korst J. H. M. Simulated annealing and Boltzmann machines. Chichester: Wiley, 1989.
6. А. Лопатин «Метод отжига» — М.: ДМК Пресс, 2008. — 94 с. — С. 25—42.

*Дата поступления в редакцию: 19.06.2019 г.*

*Опубликовано: 18.06.2019 г.*

*© Академия педагогических идей «Новация». Серия: «Научный поиск»,  
электронный журнал, 2019*

*© Сухов И.В., 2019*