

Scientific journal  
**PHYSICAL AND MATHEMATICAL EDUCATION**  
 Has been issued since 2013.

ISSN 2413-158X (online)  
 ISSN 2413-1571 (print)

Науковий журнал  
**ФІЗИКО-МАТЕМАТИЧНА ОСВІТА**  
 Видається з 2013.



<http://fmo-journal.fizmatsspu.sumy.ua/>

*Sagun A.V., Haidurov V.V., Kunchenko-Kharchenko V.I. Method of stage wolves and its modification for solving the problem of finding the optimum way // Фізико-математична освіта : науковий журнал. – 2017. – Випуск 2(12). – С. 135-139.*

*Sagun A., Haidurov V., Kunchenko-Kharchenko V. Method Of Stage Wolves And Its Modification For Solving The Problem Of Finding The Optimum Way // Physical and Mathematical Education : scientific journal. – 2017. – Issue 2(12). – P. 135-139.*

УДК 519.87

**А.В. Сагун<sup>1</sup>, В.В. Хайдуров<sup>2</sup>, В.И. Кунченко-Харченко<sup>3</sup>**

<sup>1</sup>Черкасский государственный технологический университет, Украина  
 avd29@ukr.net

<sup>2</sup>Черкасский филиал ЧВУЗ «Европейский университет», Украина  
 allif@urk.net

<sup>3</sup>Черкасский государственный технологический университет, Украина  
 kunchenco@mail.ru

## МЕТОД СТАИ ВОЛКОВ И ЕГО МОДИФИКАЦИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПОИСКА ОПТИМАЛЬНОГО ПУТИ

**Аннотация.** Сегодня все больше и больше реальных технических задач сводится к решению задачи поиска оптимального пути (задача коммивояжера). Как известно, эта задача относится к классу NP-полных задач. В данной работе рассматривается классическая задача коммивояжера. Решение задачи проводится алгоритмом «стаи волков» и его модификацией. Произведен сравнительный анализ рассмотренных алгоритмов с разными эффективными алгоритмами решения данного типа задач.

К сожалению, не существует универсальных алгоритмов решения задач оптимизации [2]. Практика показывает, что градиентные численные методы не могут быть применимы к данной задаче ввиду ряда причин [3]. Основная причина – это время работы этих алгоритмов на рассматриваемой задаче и их вычислительная сложность.

В статье предложена модификация классического метода поиска глобального оптимума целевой функции стаей волков.

**Ключевые слова:** задача коммивояжера, глобальный минимум функции, оптимизация методом «стаи волков», популяция, хромосома.

**Введение.** Большое количество реальных технических, экономических и управленческих задач из реальной жизни являются достаточно ресурсоемкими и нуждаются в поиске нужных алгоритмов для получения эффективных решений. Под эффективностью алгоритма подразумевается, в данном случае, адаптация самого же алгоритма к решаемой задаче [1]. Реальные задачи, в частности и рассматриваемая, требуют поиск глобального оптимума некоторой функции или функционала [4]. Задача коммивояжера – это задача, которая сводится к нахождению глобального минимума функции расстояний между городами [2].

Данная работа посвящена рассмотрению и анализу эффективных алгоритмов классической задачи коммивояжера и их модификации.

**Цель работы.** Сравнить существующие алгоритмы решения задачи коммивояжера с алгоритмом стаи волков и его модификации для задачи коммивояжера.

### Постановка задачи

Дано:  $I = \{1, \dots, n\}$  – множество точек, матрица  $(c_{ij})$  – попарные расстояния между точками,  $1 \leq i, j \leq n$ .

Найти: контур минимальной длины, то есть цикл, проходящий через каждую вершину ровно один раз и имеющий минимальный вес.

*Математическая формулировка задачи*

*Переменные задачи:*

$$x_{ij} = \begin{cases} 1, & \text{если есть дорога} \\ 0, & \text{иначе.} \end{cases}$$

Найти целевую функцию вида:

$$J(x) = \min \sum_{i=1}^N \sum_{j=1}^N c_{ij} x_{ij}, \quad (1)$$

где  $c_{ij}$  – расстояние между точками  $i$  и  $j$

при таких ограничениях:

$$\sum_{j=1}^N x_{ij} = 1, \forall j = \overline{1, N}, \quad \sum_{i=1}^N x_{ij} = 1, \forall i = \overline{1, N}. \quad (2)$$

### Классический алгоритм поиска стай волков

Алгоритм поиска «стай волков» – метаэвристический алгоритм поиска глобального минимума функции, разработанный в 2007 году [3]. Из названия видно, что идея алгоритма базируется на природных факторах.

Для волков типичен семейный образ жизни: они живут стаями – семейными группами, состоящими из пары вожаков, их родственников, а также одиноких волков. Внутри стаи наблюдается строго обозначенная иерархия, на вершине которой находится вождь стаи, направляющий других особей на поиск жертвы. Волки «исследуют» местность на наличие жертвы, когда кто-то из них почувствует запах жертвы, начинается ее поиск. Чем сильнее ощущается запах, тем ближе волки к жертве. Таким образом, они перемещаются в направлении усиления запаха жертвы.

Метод поиска «стай волков» копирует процесс их охоты. Допустим, что местность, на которой волки охотятся – это область поиска в смысле оптимизации, а стая – это волки. Пусть сначала сгенерировано  $N$  «волков» в евклидовом пространстве размерности  $D$ , то есть каждый волк представлен в виде вектора  $x_i = (x_{i1}, \dots, x_{iD})$ , что определяет его координаты в пространстве. Таким образом, стая (популяция) представляет собой множество потенциальных решений, координаты которых так же, как и для стайного алгоритма оптимизации [2], обновляются на каждой итерации, пока не будет найдено оптимальное решение или не будет выполнено максимально заданное количество вычислений целевой функции. Тогда функция  $f(x)$ , характеризующий, насколько сильно ощущается запах жертвы волками, является целевой, а координаты самой жертвы – оптимальной точкой. Расстояние между двумя волками  $p$  и  $q$  описывается метрикой вида:  $L(p, q)$ . Алгоритм «стаи волков» осуществляют поиск оптимальной точки-жертвы. Они делятся на группы, перемещаясь в разных направлениях и обмениваются информацией между собой. Сам алгоритм поиска (совместной охоты) можно охарактеризовать с помощью следующих правил:

- «Волк» с лучшим значением целевой функции на данной итерации является вожаком. Если на следующей итерации найдется другая «особь» с лучшим значением целевой функции, чем у предводителя, то, соответственно, стая «найдет» нового лидера;
- остальные волки исследуют местность на наличие жертвы. Причем, функция  $f(x_i)$  характеризует, как сильно ощущается запах жертвы  $i$ -м «волком». Тогда величина  $G_{Best}$  характеризует, как сильно ощущается запах жертвы вожаком стаи;
- если  $f(x_i) > G_{Best}$ , тогда  $i$ -й «волк» находится ближе к жертве, чем вождь стаи, поэтому  $i$ -й волк становится вожаком на данном этапе  $f(x_i) = G_{Best}$ . Если же  $f(x_i) < G_{Best}$ , тогда «волк» перемещается в пространстве с некоторым заранее заданным шагом  $step$ ;
- вождь стаи «сообщит» другим «волкам» в стае о своем местонахождении, как о ближайшей сейчас расстоянии до жертвы, чтобы они переместились в его направлении.

На данном этапе «вождь» рассматривается почти также, как и жертва – цель, к которой необходимо приблизиться. Тогда «волки» стаи перемещаются в направлении вожака с заранее заданным шагом  $step$ , причем  $d$ -я координата  $i$ -ого «волка» на  $(k + 1)$ -й итерации вычисляется по формуле:

$$x_{iD}^{(k+1)} = x_{iD}^{(k)} + step \frac{G_{Best}^{(k)} - x_{iD}^{(k)}}{\|G_{Best}^{(k)} - x_{iD}^{(k)}\|}. \quad (3)$$

Из формулы (3) и описания алгоритма видно, что в методе поиска «стай волков» обновляются только координаты «волков» без учета скорости их перемещения в пространстве. И, если для стайного алгоритма необходимо выбрать четыре параметра (коэффициенты обучения, инерционный вес, размер популяции), то для метода поиска стай волков достаточно подобрать только два параметра – размер популяции  $N$  и шаг  $step$ , из которых перемещаются «волки» в направлении вожака и жертвы.

**Применение классического алгоритма к задаче коммивояжера**

Следует отметить, что формула (3) стандартным образом не может быть применима к задаче коммивояжера. В этом случае берут только основной ее принцип, а именно: другие «волки» должны быть достаточно «похожими» на своего вожака, который в текущей итерации находится ближе к жертве (по значению целевой функции) [2].

Пусть, существует популяция и соответствующее приспособление каждого волка. Пример работы алгоритма показан в таблицах 1-3. Реализация алгоритмов произведена на C++.

Таблица 1

**Пример «хромосом-волков» со значением приспособленности**

Хромосомы (порядок посещения городов коммивояжером)						Значение функции приспособленности (пройденный путь соответствующим «волком»)
3	1	6	4	2	5	0,352831
5	6	2	1	3	4	0,387573
3	4	2	5	1	6	0,026252
4	1	6	2	5	3	0,358242

Далее, определяем лучшего «волка» по его значению функции приспособленности (таблица 2)

Таблица 2

**Лучшая хромосома популяции**

Лучшая хромосома						Значение функции приспособленности
3	4	2	5	1	6	0,026252

На основе лучшей хромосомы (таблица 2) генерируем новую популяцию. Для того, чтобы выполнить генерацию новой популяции, выполняем взятие части хромосомы лучшего «волка», а другую часть переставляем произвольным образом (таблица 3).

Таблица 3

**Генерация новой популяции на основе волка с лучшим приспособлением  
(по длине пройденного им пути)**

Хромосомы (порядок посещения городов коммивояжером)						Значение функции приспособленности (пройденный путь соответствующим волком)
3	4	2	5	1	6	0,026252
2	3	4	5	1	6	0,257334
3	4	2	5	6	1	0,066456
1	4	2	5	6	3	0,286301

Цветом выделена часть хромосомы, наследуемая от «волка» с лучшим приспособлением (таблица 3).

Все последующие итерации выполняются аналогичным образом: находится лучший «волк» и на его основе генерируется новая популяция.

**Модификация классического алгоритма стаи волков для задачи коммивояжера**

Как видно, из данных, приведенных в таблицах 1, 2, 3, рассмотренный алгоритм поиска решения задачи коммивояжера работает достаточно эффективно. Чтобы повысить точность расчета, можно, к примеру, выбрать не одного, а нескольких «вожачков» стаи и всю популяцию разделить на приблизительно равное количество особей в каждой части популяции. Это дает возможность получить быстрее минимум расстояния. При таком алгоритме скрещивания для получения двух решений можно применять данный метод дважды, меняя маршруты «родителей» местами. Тогда второй потомок будет получен при условии, что при расхождении связей предпочтение будет отдаваться второму маршруту.

Сравнительный анализ полученных результатов работы проводился на основе двух критериев: критерий времени и критерий оптимальной пройденной коммивояжером расстояния найденного каждым алгоритмом для разного количества городов (от 30 до 300). Ниже приведены табличные результаты (таблица 4 и 5).

Таблица 4

**Сравнительный анализ классического и модифицированного алгоритма «стаи волков» для задачи коммивояжера за критерием целевой функции расстояния**

Количество вершин	Размер популяции	Максимальное количество итераций	Классический алгоритм стаи волков		Модифицированный алгоритм стаи волков		Точное решение на минимум
			<i>fBest</i>	Ошибка в %	<i>fBest</i>	Ошибка в %	
30	30	1000	23,95767	1,58%	23,58485	0,00%	23,584849
50	60	5000	429,57576	1,85%	422,67246	0,21%	421,787667
100	100	10000	534,58485	2,10%	531,74868	1,56%	523,584849
150	200	20000	338,66566	3,22%	332,75669	1,42%	328,087454
300	500	50000	894,76557	4,75%	870,54550	1,92%	854,154940

Как видим, что модифицированный алгоритм стаи волков дает значительно точнее результаты в отличии от классического.

Таблица 5

**Сравнительный анализ классического и модифицированного алгоритма стаи волков для задачи коммивояжера за критерием времени работы**

Количество вершин	Размер популяции	Максимальное количество итераций	Время работы алгоритма (в сек.)	
			Классический алгоритм стаи волков	Модифицированный алгоритм стаи волков
30	30	1000	2,15978	2,43579
50	60	5000	10,79888	14,64515
100	100	10000	53,99440	60,52515
150	200	20000	269,97198	277,50991
300	500	50000	1349,85991	1353,46832

По результатам расчета, приведенным, в таблице 5 можно заметить, что время работы модифицированного алгоритма чуть больше. Это обусловлено тем, что происходит деление всей популяции на подгруппы, у каждой из которых есть свой «вожак».

Теперь посмотрим, насколько эффективной является модификация данного алгоритма по сравнению с популяционными алгоритмами, такими как: оптимизация «роем частиц» и классическим генетическим алгоритмом, наиболее часто используемым для решения задачи коммивояжера (таблица 6 и 7).

Таблица 6

**Сравнительный анализ модифицированного алгоритма «стаи волков» с генетическим алгоритмом и алгоритмом роя частиц за критерием оптимального пройденного пути**

Количество вершин	Размер популяции	Классический алгоритм роя частиц		Классический генетический алгоритм		Модифицированный алгоритм стаи волков		Точное решение на минимум
		<i>fBest</i>	Ошибка в %	<i>fBest</i>	Ошибка в %	<i>fBest</i>	Ошибка в %	
30	30	24,78769	5,10%	24,48957	3,84%	23,58485	0,00%	23,584849
50	60	431,75587	2,36%	428,76556	1,65%	422,67246	0,21%	421,787667
100	100	538,56887	2,86%	538,85479	2,92%	531,74868	1,56%	523,584849
150	200	338,66566	3,22%	334,77659	2,04%	332,75669	1,42%	328,087454
300	500	902,66575	5,68%	887,56746	3,91%	870,54550	1,92%	854,154940

Как видим, что модифицированный алгоритм стаи волков здесь так же дает значительно точнее результаты в отличии от остальных двух рассмотренных.

Таблица 7

**Сравнительный анализ модифицированного алгоритма стаи волков с генетическим алгоритмом и алгоритмом «роя частиц» за критерием времени работы**

Количество вершин	Размер популяции	Максимальное количество итераций	Время работы алгоритма (в сек.)		
			Классический алгоритм роя частиц	Классический генетический алгоритм	Модифицированный алгоритм стаи волков
30	30	1000	2,74546	2,43579	2,42145
50	60	5000	11,64576	18,85740	14,64515
100	100	10000	53,65869	61,27783	60,52515
150	200	20000	269,76457	271,19442	277,50991
300	500	50000	1349,57659	1356,37387	1353,46832

По результатам, приведенным в таблице 5 можно увидеть, что время работы модифицированного алгоритма в среднем незначительно больше, чем у алгоритма «роя частиц», но и в среднем меньше, чем у генетического алгоритма. Основное время работы у генетического алгоритма отведено на производство скрещивания и получения пар хромосом.

**Выводы.** Проведенное тестирование показало, что полученная модификация метода поиска стай волков лучше справляется с разными входными данными к задаче коммивояжера, чем классический алгоритм поиска стай волков. Также она показала достаточно неплохие результаты по сравнению с известными алгоритмами решения данной задачи, такими как генетический алгоритм и алгоритм роя частиц.

## Список использованных источников

1. Chiorean, I.; Lupsa, L. & Neamtiu, L. (2008). Markov Models for the Simulation of Cancer Screening Process, In International Conference on Numerical Analysis and Applied Mathematics proceedings, 143-147, ISBN 978-0-7354-0576-9, Kos, Greece, September 2008, American Institute of Physics
2. Lupsa, R.; Lupsa, L. & Neamtiu, L. (2008): Optimal Model to Solve the Transport Model for Mammography Screening, 2008 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2008) THETA 16th edition proceedings, tome 3, pp. 139-142, ISBN 978-1-4244-2576-1, Cluj-Napoca
3. Manthey, B. (2009). On Approximating Multi-Criteria TSP, In Symposium on Theoretical Aspects of Computer Science, 637648, Freiburg, 2009, www.stacs-conf.org
4. Neamtiu, L. (2009). A Medical Resources Allocation Problem, Results in Mathematics, Vol. 53, nr. 3-4 (July), 341-348, ISSN 1422-6383

## References

1. Chiorean, I.; Lupsa, L. & Neamtiu, L. (2008). Markov Models for the Simulation of Cancer Screening Process, In International Conference on Numerical Analysis and Applied Mathematics proceedings, 143-147, ISBN 978-0-7354-0576-9, Kos, Greece, September 2008, American Institute of Physics
2. Lupsa, R.; Lupsa, L. & Neamtiu, L. (2008): Optimal Model to Solve the Transport Model for Mammography Screening, 2008 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2008) THETA 16th edition proceedings, tome 3, pp. 139-142, ISBN 978-1-4244-2576-1, Cluj-Napoca
3. Manthey, B. (2009). On Approximating Multi-Criteria TSP, In Symposium on Theoretical Aspects of Computer Science, 637648, Freiburg, 2009, www.stacs-conf.org
4. Neamtiu, L. (2009). A Medical Resources Allocation Problem, Results in Mathematics, Vol. 53, nr. 3-4 (July), 341-348, ISSN 1422-6383

**METHOD OF STAGE WOLVES AND ITS MODIFICATION FOR SOLVING THE PROBLEM OF FINDING THE OPTIMUM WAY****A. Sagun<sup>1</sup>, V. Haidurov<sup>2</sup>, V. Kunchenko-Kharchenko<sup>3</sup>**<sup>1</sup>*Cherkassy State Technological University, Ukraine*<sup>2</sup>*Cherkassy branch of the European University "European University", Ukraine*<sup>3</sup>*Cherkassy State Technological University, Ukraine*

**Abstract.** Today, more and more real technical problems is reduced to solving the problem of finding the optimal path (traveling salesman problem). As you know, this problem belongs to the class of NP-complete problems. In this paper we consider the classical traveling salesman problem. The decision problem is an algorithm "pack of wolves" and its modification. Comparative analysis of the considered algorithms with different efficient algorithms for solving problems of this type.

Unfortunately, there is no universal algorithms for solving optimization problems [2]. Practice shows that the gradient of the numerical methods may not be applicable to this task for a number of reasons [3]. The main reason is the operating time of these algorithms on the considered task and their computational complexity.

The paper proposes a modification of the classical method of finding the global optimum of the objective function by a pack of wolves.

**Key words:** traveling salesman problem, global minimum of function, optimization of wolf pack, population, chromosome.