

Математика и математическое моделирование. 2020.  
№ 02. С. 25 – 45.

DOI: [10.24108/mathm.0220.0000217](https://doi.org/10.24108/mathm.0220.0000217)



© Д.О. Захаров, А.П. Карпенко, 2020

# Математика и Математическое моделирование

Сетевое научное издание

<http://mathmelpub.ru>

ISSN 2412-5911

УДК 519.6

## Исследование эффективности популяционного алгоритма лиги чемпионов для задачи глобальной оптимизации

Захаров Д. О.<sup>1,\*</sup>, Карпенко А.П.<sup>1</sup>

<sup>1</sup>МГТУ им. Н.Э. Баумана, Москва, Россия

\* [dimazakharov98@mail.ru](mailto:dimazakharov98@mail.ru)

---

Работа посвящена исследованию эффективности нового алгоритма лиги чемпионов (League Championship Algorithm, LCA). Исследование выполнено на основе сравнения эффективности алгоритма LCA с эффективностью широко известного алгоритма роя частиц (Particle Swarm optimization, PSO). Представляем программную реализацию этих алгоритмов, а также результаты широких вычислительных экспериментов с использованием разработанного программного обеспечения. Вычислительные эксперименты выполнены с использованием сферической функции, а также известных тестовых функций Розенброка, Растргина и Экли. Результаты экспериментов показали, что алгоритм LCA в некоторых случаях обеспечивает более высокую эффективность, чем алгоритм PSO, и имеет высокий потенциал развития.

**Ключевые слова:** глобальная оптимизация, мета-эвристический алгоритм, популяционный алгоритм, алгоритм лиги чемпионов, алгоритм роя частиц

---

Представлена в редакцию: 05.03.2020, исправлена 19.03.2020

---

### Введение

Активное использование интеллектуальных систем для решения различных инженерных и научных задач началось относительно недавно. Однако это направление исследований уже позволило добиться заметного прогресса в областях автоматизированного проектирования, компьютерного зрения, суррогатного моделирования и др. Обычно к сфере интеллектуальных систем относят искусственные нейронные сети, системы с нечёткой логикой, различные варианты эволюционных алгоритмов, например, алгоритмов роя частиц, искусственной иммунной системы и др. [1].

В последние годы метаэвристические алгоритмы глобальной параметрической оптимизации (ГПО), вдохновленные различными природными и социальными процессами, привлекают внимание многих исследователей из различных областей науки и техники

(см., например, библиографию в книге [1]). Этот интерес обусловлен высокой потенциальной эффективностью этих алгоритмов ГПО при решении практически важных задач оптимизации в таких областях, как промышленность, бизнес, экология и т.д.

Известны различные классификации метаэвристических алгоритмов ГПО. На верхнем уровне иерархии эти алгоритмы можно разделить на две большие группы — эволюционные и роевые алгоритмы.

Эволюционные алгоритмы инспирированы природной способностью живых существ эволюционировать, приспособляясь к окружающей среде. Можно сказать, что эти алгоритмы реализуют модели природных эволюционных процессов. Первым и самым известным эволюционным алгоритмом является генетический алгоритм [2].

Появление роевых алгоритмов связано с исследованиями колоний социальных организмов. Результаты изучения поведения особей в таких колониях вдохновили разработку весьма эффективных алгоритмов ГПО (а также ряда других алгоритмов, например, алгоритмов машинного обучения). Так, имитационные исследования движения стаи птиц привели к разработке оптимизационного алгоритма роя частиц, (Particle Swarm Optimization, PSO), а изучение поведения муравьев в процессе поиска ими пищи — алгоритма колонии муравьев [3].

В качестве общего наименования эволюционных и роевых алгоритмов используем термин популяционные алгоритмы (П-алгоритмы) [4].

В [5] предложен новый П-алгоритм ГПО, получивший наименование алгоритм лиги чемпионов (League Championship Algorithm, LCA), который имитирует процесс проведения чемпионата спортивной лиги. В алгоритме LCA некоторое число агентов, моделирующих команды, соревнуются в искусственной лиге в течение нескольких туров. Основываясь на расписании игр лиги, команды играют в парах. Результат каждой из игр (выигрыш/проигрыш) определяют, учитывая известную силу игры команды, обусловленную расстановкой членов команды. На основе опыта предыдущих игр, каждая из команд вырабатывает изменения в стиле игры, определяемые расстановкой, для соревнования в следующем туре.

В работе [5] представлен базовый вариант алгоритма лиги чемпионов, позволяющий решать задачу многомерной безусловной ГПО. Рассмотрены базовые процедуры, применяющиеся в алгоритме с целью формирования расписания лиги, определения победителя в матче, изменения состава команды (расстановки) для следующей итерации алгоритма. Приведены результаты сравнения эффективности алгоритма LCA и канонического алгоритма PSO для сферической тестовой функции, а также тестовых функций Розенброка, Растргина, Экли.

В настоящее время известно небольшое число работ, посвященных алгоритму LCA. Так, в работе [6] предложена модификация этого алгоритма, ориентированная на решение задачи многомерной условной ГПО. Представлена процедура определения победителя в матче, позволяющая решать задачи условной оптимизации. Приведены результаты оценки эффективности модифицированного алгоритма для некоторых тестовых задач многомер-

ной условной оптимизации. В работе [7] выполнено сравнение эффективности алгоритма LCA и некоторых известных П-алгоритмов в процессе решения задачи условной оптимизации ГПО из области проектирования механизмов.

Данная работа посвящена сравнительному исследованию эффективности алгоритма LCA и модифицированного алгоритма PSO. Выполнена программная реализация этих алгоритмов. С использованием разработанного программного обеспечения проведены вычислительные эксперименты по сравнительной оценке эффективности указанных алгоритмов. В качестве тестовых использованы сферическая функция, а также функции Розенброка, Растригина и Экли. Вычислительный эксперимент выполнен для размерностей вектора варьируемых параметров, равных 2, 4, 8, 16, 32, 64. С целью статистической оценки значений используемых индикаторов эффективности алгоритмов, использован режим мультистарта: каждый вычислительный эксперимент повторен 100 раз для разных начальных положений агентов популяции.

Результаты исследования показали, что алгоритм LCA в некоторых случаях обеспечивает более высокую эффективность, чем алгоритм PSO. На этом основании можно утверждать, что алгоритм LCA имеет высокий потенциал развития, например, представляется перспективной гибридизация этого алгоритма по той или иной схеме [8] с алгоритмами локальной оптимизации.

## **1. Предпосылки алгоритма лиги чемпионов (LCA)**

Спортивная лига – это группа спортивных команд или отдельных спортсменов, которые соперничают друг с другом в определенном виде спорта. Лига обычно используется для организации соревнований (чемпионатов) в командных, а не в индивидуальных видах спорта. Чемпионат может быть проведен в нескольких вариантах. Например, в одном из вариантов каждая команда играет против всех остальных команд определенное число раз (данный вариант используется в алгоритме LCA). При этом команда с лучшим результатом становится чемпионом, основываясь на некоторой системе учета побед и поражений, либо на системе очков, когда определенное количество очков присуждается за победу, поражение или ничью.

Обычно у каждой команды есть собственный стиль игры, который проявляется в использовании той или иной схемы расстановки игроков команды на игровом поле. Чтобы достигнуть желаемого результата по окончании турнира, тренеры команд разрабатывают подходящие состав и расстановку игроков для каждого из последующих матчей. После матча тренеры анализируют игру своей команды, а также игру соперника, чтобы спланировать, какие изменения целесообразно внести в расстановку команды для повышения вероятности победы в следующем матче.

Алгоритм LCA основан на следующих правилах:

- более вероятно, что команда, имеющая бóльшую игровую силу, выиграет игру;
- исход игры заранее не предсказуем даже при известных игровых силах соперников;

- вероятность того, что команда  $i$  победит команду  $j$ , равна вероятности того, что команда  $j$  проиграет команде  $i$ ;
- результатом игры может быть только победа или поражение;
- расстановка команды формируется, исходя из результата предыдущей игры и информации о следующем сопернике.

## 2. Алгоритм LCA

Рассматриваем задачу безусловной ГПО вида

$$\min_X f(X) = f(X^*) = f^*, X \in R^n,$$

где  $X = (x_1, x_2, \dots, x_n)$  – вектор варьируемых параметров;  $X^*$  – искомое оптимальное значение этого вектора;  $f(X)$  – целевая функция;  $f(X^*)$  – оптимальное значение целевой функции.

В алгоритме LCA используются следующие параллели: лига – популяция решений;  $i$ -я команда –  $i$ -ый агент популяции; расстановка команды – значения компонентов вектора варьируемых параметров; игровой тур – итерация алгоритма; игровая сила – значение целевой функции.

Блок-схема алгоритма LCA представлена на рис. 1. Начальная расстановка команды считается её текущей лучшей расстановкой для первой итерации алгоритма.

Результат матча между командами  $i, j$  на текущей итерации  $t$  определяется вероятностью победы команды  $i$ , равной

$$p_i^t = \frac{f(X_j^t) - f^{t*}}{f(X_j^t) + f(X_i^t) - 2f^{t*}},$$

где  $X_i^t, X_j^t$  – текущие расстановки команд  $i, j$  соответственно;  $f^{t*}$  – текущее лучшее достигнутое значение целевой функции.

Формирование расстановки команды для следующего игрового тура происходит на этапе послематчевого анализа. При формировании новой расстановки для команды  $i$ , которая на итерации  $t$  соревновалась с командой  $j$ , а на итерации  $t+1$  должна будет соревноваться с командой  $l$ , которая в свою очередь соревновалась с командой  $k$  на итерации  $t$ , используются представленные ниже правила (всюду имеется в виду, что  $d \in [1:n]$ ).

Команда  $i$  победила команду  $j$ , и команда  $l$  победила команду  $k$ :

$$x_{i,d}^{t+1} = b_{i,d}^t + y_{i,d}^t \left( c_1 r_1 (x_{i,d}^t - x_{k,d}^t) + c_1 r_2 (x_{i,d}^t - x_{j,d}^t) \right). \quad (1)$$

Команда  $i$  победила команду  $j$ , а команда  $l$  проиграла команде  $k$ :

$$x_{i,d}^{t+1} = b_{i,d}^t + y_{i,d}^t \left( c_2 r_1 (x_{k,d}^t - x_{i,d}^t) + c_1 r_2 (x_{i,d}^t - x_{j,d}^t) \right). \quad (2)$$

Команда  $i$  проиграла команде  $j$ , а команда  $l$  победила команду  $k$ :

$$x_{i,d}^{t+1} = b_{i,d}^t + y_{i,d}^t \left( c_1 r_2 (x_{i,d}^t - x_{k,d}^t) + c_2 r_1 (x_{j,d}^t - x_{i,d}^t) \right). \quad (3)$$

Команда  $i$  проиграла команде  $j$ , и команда  $l$  проиграла команде  $k$ :

$$x_{i,d}^{t+1} = b_{i,d}^t + y_{i,d}^t \left( c_2 r_2 (x_{k,d}^t - x_{i,d}^t) + c_2 r_1 (x_{j,d}^t - x_{i,d}^t) \right). \quad (4)$$

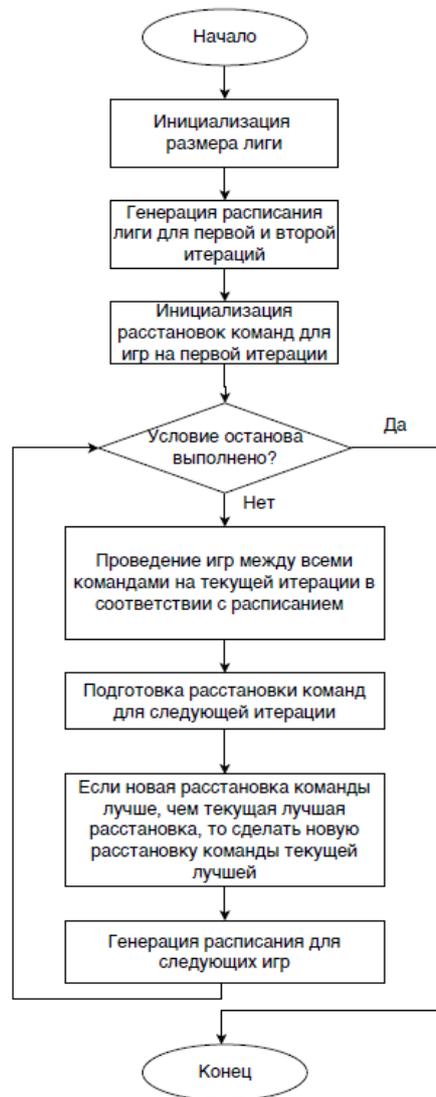


Рис. 1. Блок-схема алгоритма LCA

В формулах (1) – (4) использованы следующие обозначения:  $x_{i,d}^{t+1}$  –  $d$ -я компонента вектора варьируемых параметров команды  $i$  на итерации  $t+1$ ;  $r_1, r_2$  – независимые случайные величины равномерно распределенные в интервале  $[0;1]$ ;  $c_1, c_2$  – свободные параметры алгоритма;  $b_{i,d}^t$  –  $d$ -я компонента лучшей расстановки команды  $i$ , достигнутой за  $t$  итераций;  $y_{i,d}^t$  –  $d$ -я компонента бинарного вектора изменений  $Y_i^t$ , которая определяет будет ли изменена  $d$ -я компонента лучшей расстановки команды или нет. Если  $y_{i,d}^t = 1$ , то  $x_{i,d}^{t+1}$  будет отличаться от  $b_{i,d}^t$  на величину, которая может быть рассчитана в соответствии с правилами (1) – (4), в противном случае, если  $y_{i,d}^t = 0$  то  $x_{i,d}^{t+1} = b_{i,d}^t$ . Таким образом, бинарный вектор изменений  $Y_i^t$  состоит из нулей и единиц, расположенных в случайном порядке. Число единиц определяется величиной  $q_i^t$ .

В исходном алгоритме LCA для определения величины  $q_i^t$  используется усеченное геометрическое распределение [9], в соответствии с которым величина  $q_i^t$  может быть вычислена по формуле

$$q_i^t = \left\lceil \frac{\ln(1 - (1 - (1 - p_c)^n)r)}{\ln(1 - p_c)} \right\rceil,$$

где  $r$  – независимая случайная величина, равномерно распределенная в интервале  $[0;1]$ ;

$p_c$  – свободный параметр алгоритма.

Более детальное описание алгоритма представлено в [5].

### 3. Программное обеспечение

При разработке программного обеспечения (решателя LCA) мы исходили из следующих функциональных требований:

- программное обеспечение должно гарантировать возможность выбора размера лиги (числа агентов в популяции);
- должна быть предусмотрена возможность выбора размера расстановки команд (размерности вектора варьируемых параметров);
- должен быть простым переход к использованию другой тестовой функции;
- решение задачи ГПО должно производиться с использованием метода мультистарта.

Решатель LCA реализован на языке программирования C++ и содержит классы *Team*, *League*, *LCA\_Solver*, которые включают в себя следующие методы.

Класс *Team*: *void Init ()* – начальная инициализация всех свойств команды; *void TeamPrepare ()* – подготовка расстановки команды для следующей итерации алгоритма; *void Refresh ()* – обновление свойств команды.

Класс *League*: *void Init ()* – начальная инициализация свойств лиги; *void NewSchedule ()* – подготовка расписания игр команд для следующей итерации алгоритма; *void Tour ()* – проведение одного игрового тура спортивной лиги; *int Game ()* – проведение игры между двумя командами; *void Refresh ()* – обновление свойств лиги.

Класс *LCA\_Solver*: *void Init ()* – начальная инициализация всех свойств решателя; *void Solve ()* – однократное решение задачи оптимизации целевой функции с использованием алгоритма LCA; *void Refresh ()* – обновление свойств решателя.

### 4. Вычислительные эксперименты

Во всех вычислительных экспериментах использовались следующие значения свободных параметров алгоритма LCA:  $c_1 = 0,5$ ;  $c_2 = 0,5$ ;  $p_c = 0,01$ ;  $L = 50$ , где  $L$  – размер лиги [5].

Значения свободных параметров алгоритма PSO приняты равными  $b_c = b_s = 2$ . Инерционный параметр  $b_l$  изменялся в процессе итераций от максимального значения 0,7 до минимального, равного 0,2, по линейному закону [5]. Число частиц в рое принято равным 50; топология соседства – модифицированное «кольцо», в котором соседними для

данной частицы являются особи, порядковый номер которых отличался от порядкового номера этой частицы не более чем на семь [10].

Отметим, что указанный закон изменения значений параметра  $b_l$  обеспечивает повышение интенсификационных свойств алгоритма PSO с ростом числа итераций (конечно, ущерб его диверсификационных свойств). Использование модифицированной топологии соседства типа «кольцо» является авторским решением, направленным на повышение эффективности алгоритма PSO, прежде всего, на повышение вероятности локализации этим алгоритмом глобального минимума целевой функции (за счет диверсификации поиска).

В качестве условия останова использован критерий стагнации вычислительного процесса: улучшение в течение 30 итераций значений целевой функции не более чем на величину 0,00001. Применен метод мультистарта с числом стартов  $S = 100$ . Эксперименты выполнены для размерности вектора варьируемых параметров  $n$  равной 2, 4, 8, 16, 32, 64. Оценка эффективности алгоритмов реализована с помощью следующих индикаторов:

- среднее по мультистарту число итераций  $\bar{N} \rightarrow \min$  ;
- среднее отклонение найденного лучшего значения целевой функции от своего точного минимального значения

$$\bar{F} = M(|f^* - \tilde{f}^*|) = \overline{|f^* - \tilde{f}^*|} \rightarrow \min ,$$

где  $M(\cdot)$  – оценка математического ожидания;  $\tilde{f}^* = f(\tilde{X}^*)$  – найденное приближение к решению задачи.

- среднее расстояние между точками  $\tilde{X}^*, X^*$

$$\bar{X} = M(\|X^* - \tilde{X}^*\|_2) = \overline{\|X^* - \tilde{X}^*\|_2} \rightarrow \min$$

где  $\|\cdot\|_2$  – евклидова норма;

- оценка вероятности локализации минимума целевой функции, равная

$$\tilde{P} = \frac{S^+}{S} \rightarrow \max ,$$

где  $S^+$  – число стартов, в результате которых минимум был локализован.

В последней формуле использован критерий локализации

$$\|X^* - \tilde{X}^*\|_2 \leq \delta_x.$$

Здесь  $\delta_x$  – точность локализации минимума, определяемая формулой

$$\delta_x = \lambda a \sqrt{n},$$

где  $\lambda$  – константа, принятая равной 0,01;  $a$  – длина ребра гиперкуба, в котором осуществляется инициализация начальной популяции.

#### 4.1. Сферическая функция

В данном исследовании использована целевая функция вида

$$f(X) = \sum_{i=1}^n x_i^2,$$

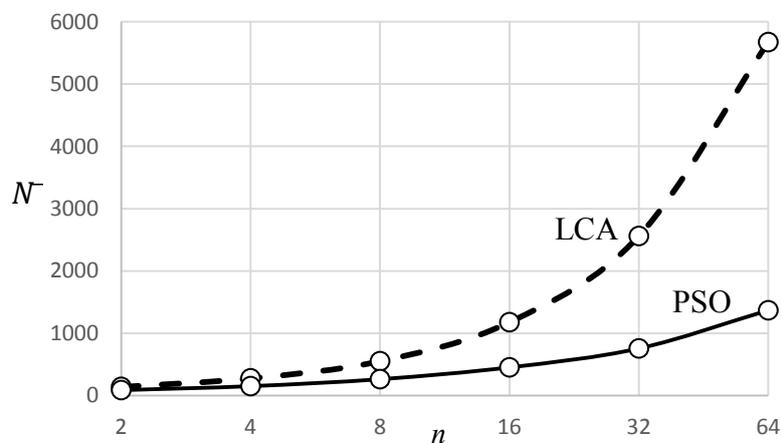
где  $x_i \in [-a; a]$ ;  $a = 100$ .

В табл. 1 и на рис. 2, иллюстрирующем табл. 1, представлены некоторые результаты вычислительных экспериментов.

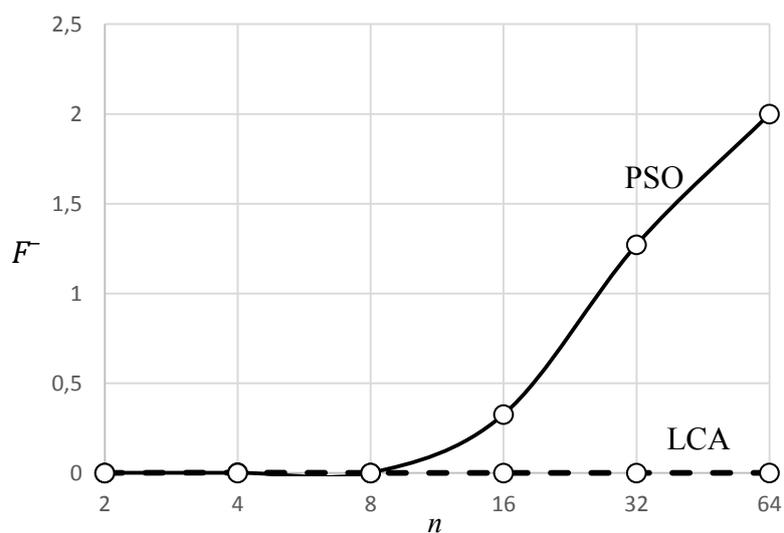
Табл. 1. Сравнение эффективности алгоритмов *LCA*, *PSO*: сферическая функция

Индикатор эффективности	$n$	<i>LCA</i>	<i>PSO</i>
$\bar{N}$	2	139,7	88,1
	4	276,5	151,5
	8	551,5	263,7
	16	1177,6	452,9
	32	2560,6	756,6
	64	5672,1	1368,6
$\bar{F}$	2	0,0	0,0
	4	0,0	0,0
	8	0,0	0,0
	16	0,0	0,3
	32	0,0	1,3
	64	0,0	2,0
$\bar{X}$	2	0,0	0,0
	4	0,0	0,0
	8	0,0	0,0
	16	0,0	0,1
	32	0,0	0,1
	64	0,0	0,2

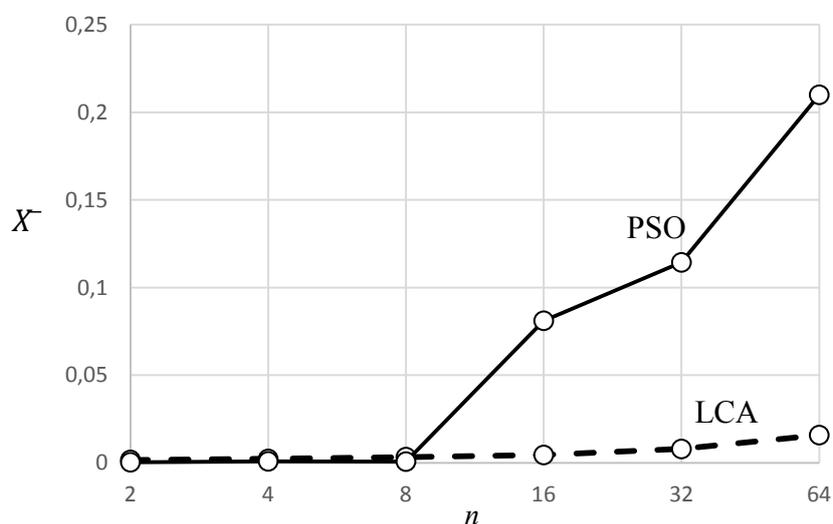
Вычислительный эксперимент показал, что, как и следовало ожидать, оба алгоритма обеспечили оценку  $\bar{F}$  вероятности локализации минимума сферической функции, равную единице, то есть во всех 100 стартах алгоритмы *LCA*, *PSO* отыскивали минимум этой функции с заданной точностью.



а)



б)



в)

**Рис. 2.** Значения индикаторов эффективности  $\bar{N}$ ,  $\bar{F}$ ,  $\bar{X}$  в функции размерности  $n$  вектора варьируемых параметров: сферическая функция

#### 4.2. Функция Розенброка

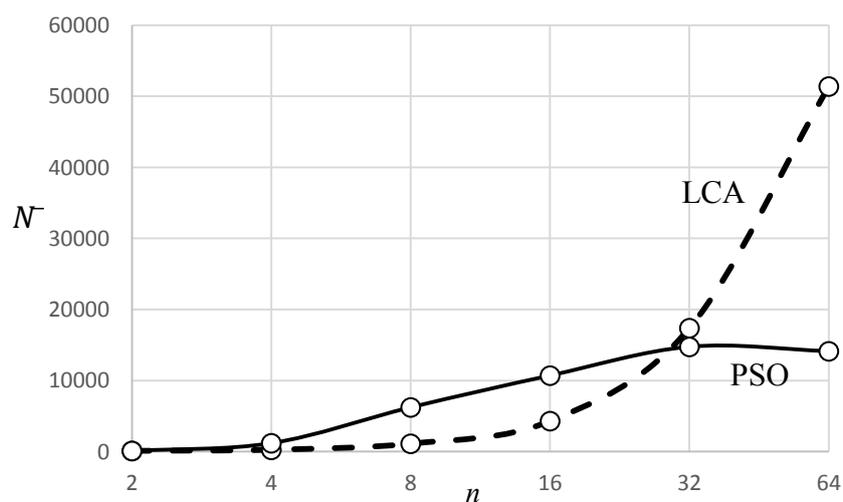
Функция Розенброка (овражная функция):

$$f(X) = \sum_{i=1}^{n-1} 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2; x_i \in [-a; a]; a = 2,048.$$

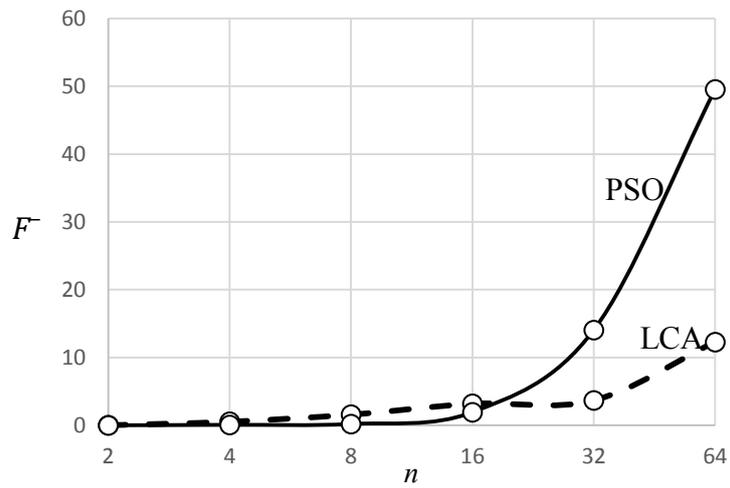
Результаты вычислительных экспериментов представлены в табл. 2 и на иллюстрирующих эту табл. рис. 3.

Табл. 2. Сравнение эффективности алгоритмов *LCA*, *PSO*: функция Розенброка

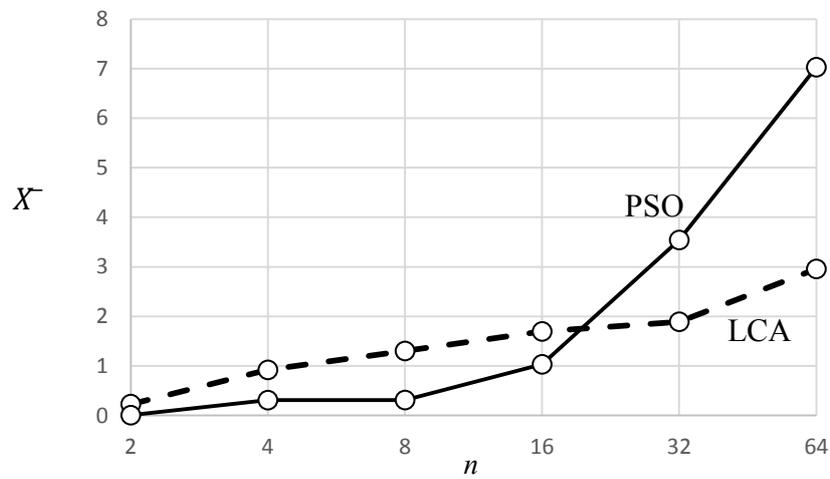
Индикатор эффективности	$n$	<i>LCA</i>	<i>PSO</i>
$\bar{N}$	2	92,8	94,2
	4	255,8	1185,6
	8	1109,7	6225,4
	16	4296,6	10702,5
	32	17380,3	14745,4
	64	51394,8	14124,3
$\bar{F}$	2	0,0	0,0
	4	0,6	0,1
	8	1,6	0,2
	16	3,2	1,9
	32	3,7	14,1
	64	12,3	49,6
$\bar{X}$	2	0,2	0,0
	4	0,9	0,3
	8	1,3	0,3
	16	1,7	1,0
	32	1,9	3,5
	64	3,0	7,0
$\bar{P}$	2	0,12	1,00
	4	0,01	0,27
	8	0	0,45
	16	0	0,23
	32	0	0,01
	64	0	0



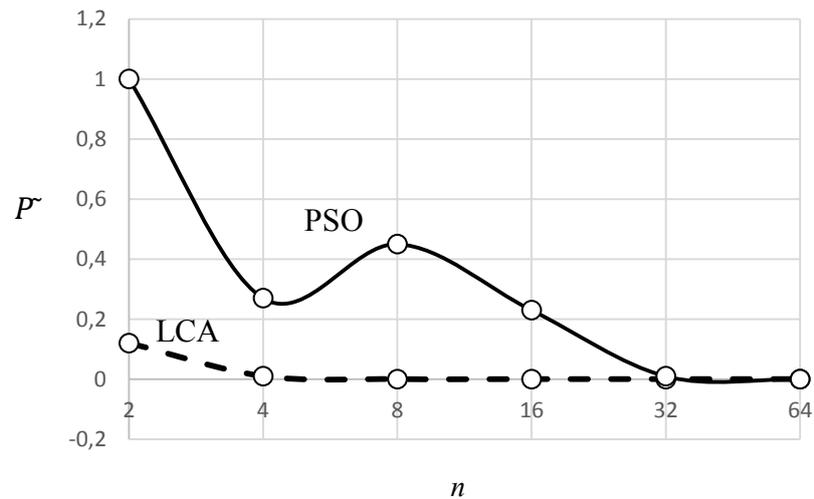
а)



б)



в)



г)

Рис. 3. Значения индикаторов эффективности  $\bar{N}$ ,  $\bar{F}$ ,  $\bar{X}$ ,  $\bar{P}$  в функции размерности  $n$  вектора варьируемых параметров: функция Розенброка

### 4.3. Функция Растригина

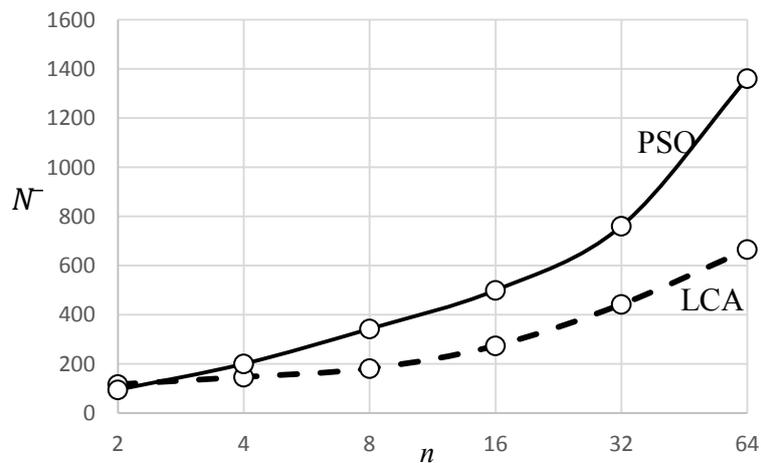
Функция Растригина (многоэкстремальная функция):

$$f(X) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10); \quad x_i \in [-a; a], a = 5,12.$$

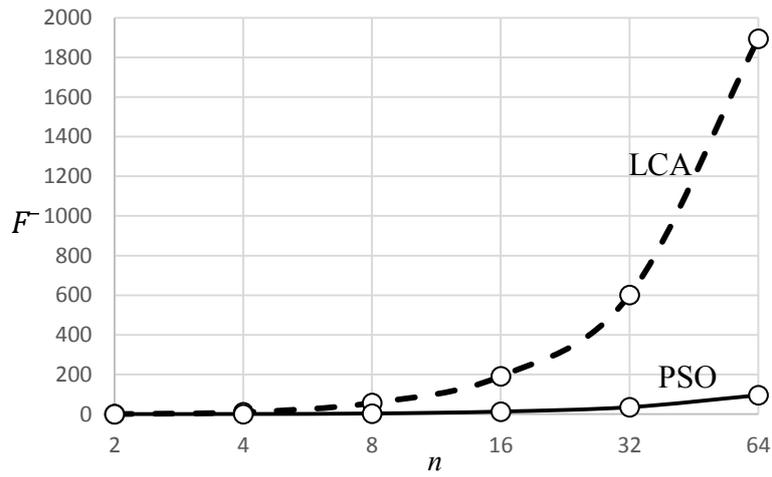
В табл. 3 представлены полученные экспериментальные результаты. Результаты приведенные в табл. 3 проиллюстрированы на рис. 4.

Табл. 3. Сравнение эффективности алгоритмов *LCA*, *PSO*: функция Растригина

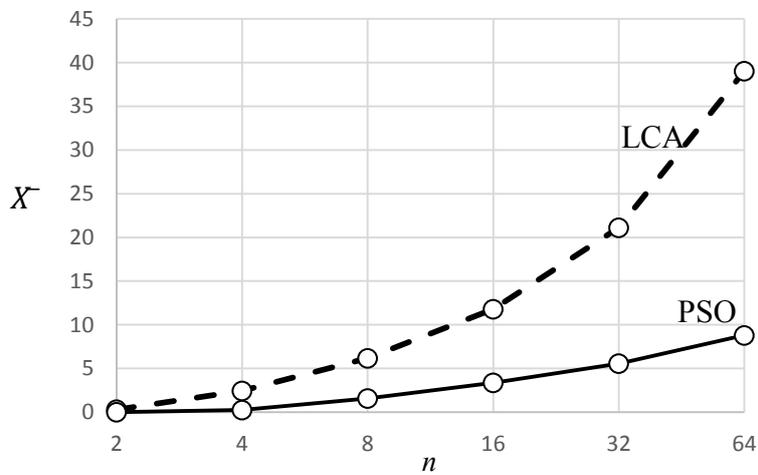
Индикатор эффективности	<i>n</i>	<i>LCA</i>	<i>PSO</i>
$\bar{N}$	2	116,6	94,9
	4	146,5	200,4
	8	181,4	342,7
	16	273,9	499,6
	32	442,4	760,5
	64	665,7	1360,8
$\bar{F}$	2	0,6	0,0
	4	10,1	0,3
	8	56,8	2,9
	16	191,5	12,7
	32	601,7	35,2
	64	1894,6	96,0
$\bar{X}$	2	0,3	0,0
	4	2,4	0,2
	8	6,2	1,6
	16	11,8	3,4
	32	21,1	5,6
	64	39,0	8,8
$\bar{P}$	2	0,73	0,99
	4	0,01	0,76
	8	0	0,04
	16	0	0
	32	0	0
	64	0	0



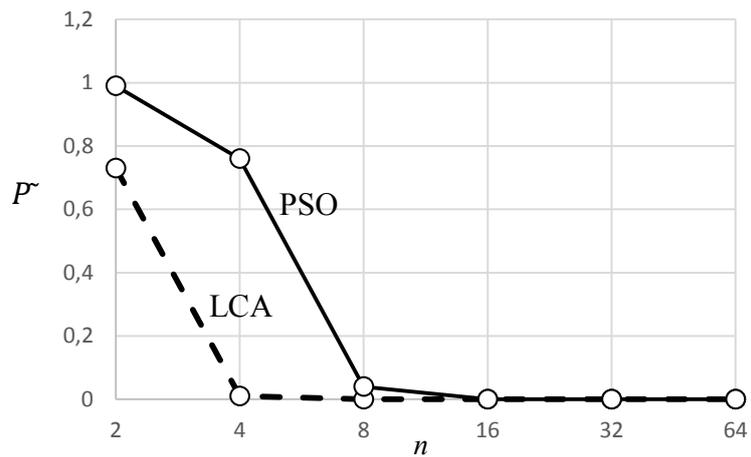
а)



б)



в)



г)

**Рис. 4.** Значения индикаторов эффективности  $\bar{N}$ ,  $\bar{F}$ ,  $\bar{X}$ ,  $\bar{P}$  в функции размерности  $n$  вектора варьируемых параметров: функция Растригина

#### 4.4. Функция Экли

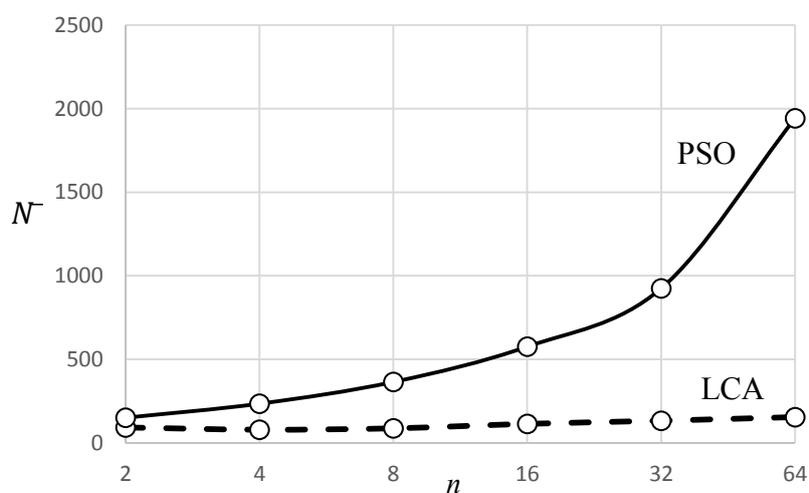
Функция Экли (многоэкстремальная):

$$f(X) = -20 \exp\left(-0,2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right); \quad x_i \in [-a; a], \quad a = 32,76.$$

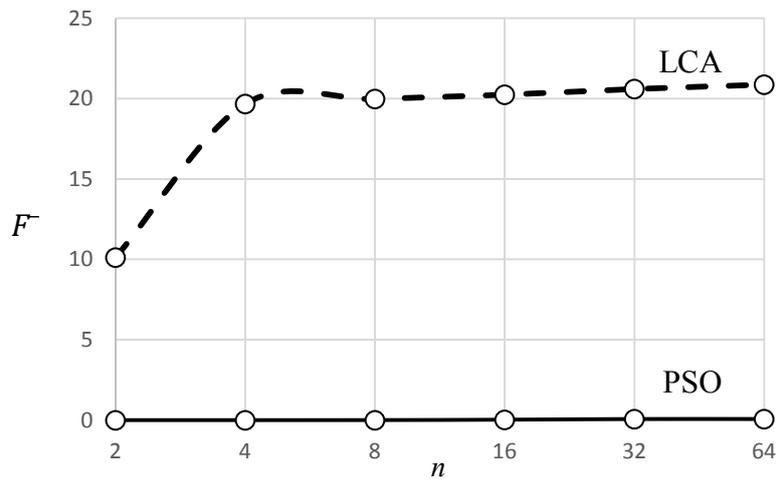
Результаты экспериментов представлены в табл. 4, а также на рис. 5.

Табл. 4. Сравнение эффективности алгоритмов *LCA*, *PSO*: функция Экли

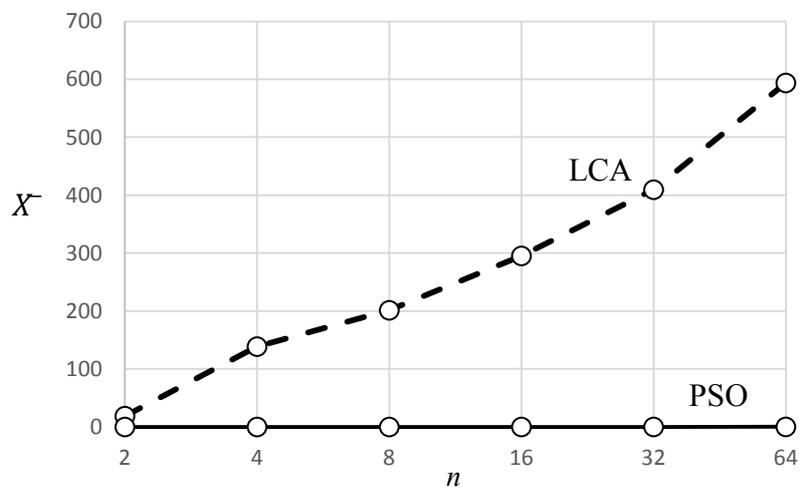
Индикатор эффективности	$n$	<i>LCA</i>	<i>PSO</i>
$\bar{N}$	2	92,3	150,7
	4	78,9	234,6
	8	87,1	364,8
	16	113,6	576,0
	32	132,2	924,4
	64	153,9	1942,2
$\bar{F}$	2	10,1	0,0
	4	19,7	0,0
	8	20,0	0,0
	16	20,2	0,0
	32	20,6	0,1
	64	20,9	0,1
$\bar{X}$	2	18,8	0,0
	4	138,8	0,0
	8	201,4	0,0
	16	295,3	0,0
	32	409,6	0,1
	64	593,9	0,1
$\bar{P}$	2	0,13	1,00
	4	0,01	1,00
	8	0,01	1,00
	16	0,01	1,00
	32	0,01	0,99
	64	0	0,99



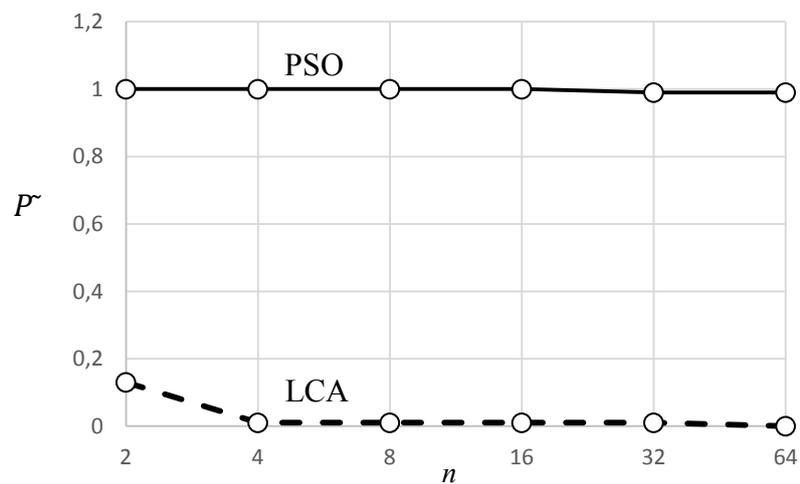
а)



б)



в)



г)

**Рис. 5.** Значения индикаторов эффективности  $\bar{N}$ ,  $\bar{F}$ ,  $\bar{X}$ ,  $\bar{P}$  в функции размерности  $n$  вектора варьируемых параметров: функция Экли

## 5. Обсуждение результатов экспериментов

Интегральные результаты сравнения эффективности алгоритмов LCA, PSO представлены в табл. 5.

**Табл. 5.** Сравнение эффективности алгоритмов LCA, PSO: (+, -, =) — алгоритм LCA лучше, хуже и эквивалентен алгоритму PSO соответственно

Индикатор	Функция	Размерность вектора варьируемых параметров					
		2	4	8	16	32	64
$\bar{N}$	Сфера	–	–	–	–	–	–
	Розенброка	+	+	+	+	–	–
	Растригина	–	+	+	+	+	+
	Экли	+	+	+	+	+	+
$\bar{F}$	Сфера	–	–	–	+	+	+
	Розенброка	–	–	–	–	+	+
	Растригина	–	–	–	–	–	–
	Экли	–	–	–	–	–	–
$\bar{X}$	Сфера	–	–	–	+	+	+
	Розенброка	–	–	–	–	+	+
	Растригина	–	–	–	–	–	–
	Экли	–	–	–	–	–	–
$\bar{P}$	Сфера	=	=	=	=	=	=
	Розенброка	–	–	–	–	–	=
	Растригина	–	–	–	=	=	=
	Экли	–	–	–	–	–	–

При анализе результатов, представленных в табл. 5, исключаем из рассмотрения сферическую функцию, поскольку для прикладных задач глобальной оптимизации такой ландшафт целевой функции нетипичен.

Во-первых, в табл. 5 обращает на себя внимание высокая эффективность алгоритма LCA для овражной функции Розенброка. Так, по индикатору эффективности  $\bar{N}$  этот алгоритм оказывается лучше рассматриваемого варианта алгоритм PSO для размерностей вектора  $X$  варьируемых параметров, равных 2, 4, 8, 16, по индикаторам  $\bar{F}$ ,  $\bar{X}$  — для размерностей 32 и 64. Отыскание глобального минимума овражных функций является сложной задачей для любого из алгоритмов ГПО, тем более функций, имеющих высокую размерность вектора  $X$ .

Во-вторых, из табл. 5 следует, что алгоритм LCA показывает в ряде случаев эффективность, равную или превышающую эффективность алгоритма PSO для многоэкстремальных функций Растригина и Экли с точки зрения индикатора эффективности  $\bar{N}$ . Этот

факт особенно важен, поскольку вычислительные затраты на решение задачи оптимизации пропорциональны значениям этого индикатора.

Отметим следующее обстоятельство. Известно, что локальные минимумы функции Растригина располагаются на  $n$ -мерной целочисленной решетке, и глобальный минимум (равный нулю) достигается в точке с нулевыми координатами. Число локальных минимумов функции Растригина может быть вычислено по формуле  $l = z^n - 1$ , где  $z$  — количество целых чисел в области исследования по одному измерению [11]. Поскольку в нашем случае, легко видеть,  $z = 11$ , получаем  $l = 11^n - 1$ , так что, например, для  $n = 32$  число локальных минимумов равно примерно  $10^{32}$ . Очевидно, что такое большое число локальных минимумов функции Растригина в рассматриваемой области исследования делает задачу отыскания ее глобального минимума чрезвычайно сложной.

В целом, указанные результаты сравнительной оценки эффективности алгоритма LCA позволяют характеризовать его как весьма перспективный.

## Заключение

Представленные результаты вычислительных экспериментов показывают, что в большинстве рассмотренных случаев алгоритм LCA проигрывает алгоритму PSO по всем рассматриваемым индикаторам эффективности. Однако следует отметить, что алгоритм PSO, даже в своем каноническом варианте, является одним из наиболее эффективных П-алгоритмов [12]. Мы же для сравнения использовали модифицированный вариант алгоритма PSO, который является более эффективным, чем канонический вариант.

В то же время, результаты исследования показали, что алгоритм LCA в некоторых случаях обеспечивает более высокую эффективность, чем алгоритм PSO. Это обстоятельство позволяет утверждать, что алгоритм LCA имеет высокий потенциал развития. Например, возможна параметрическая мета-оптимизация алгоритма, а именно подбор наилучших значений его свободных параметров [13]. Возможна также гибридизация алгоритма LCA с различными популяционными и классическими алгоритмами глобальной и локальной оптимизации. Так, на наш взгляд представляет значительный интерес гибридизация этого алгоритма с тем или иным алгоритмом локальной оптимизации по схеме последовательной гибридизации [14].

Использованная нами в исследовании модификация алгоритма PSO отличается от канонического алгоритма следующим: 1) использование инерционного свободного параметра алгоритма, значения которого линейно убывают с ростом числа итераций; 2) применение оригинальной топологии соседства частиц, построенной на основе классической топологии типа «кольцо». Таким образом, представленные результаты исследования эффективности этого модифицированного алгоритма PSO представляют самостоятельный интерес.

## Список литературы

1. Bo Xing, Wen-Jing Gao. Innovative computational intelligence: a rough guide to 134 clever algorithms. Cham: Springer, 2014. 451 p. DOI: [10.1007/978-3-319-03404-1](https://doi.org/10.1007/978-3-319-03404-1)
2. Blum C., Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison // ACM Computing Surveys. 2003. Vol. 35. No. 3. Pp. 268–308. DOI: [10.1145/937503.937505](https://doi.org/10.1145/937503.937505)
3. Engelbrecht A.P. Computational intelligence: an introduction. 2<sup>nd</sup> ed. Chichester; Hoboken: Wiley, 2007. 597 p.
4. Карпенко А.П. Современные алгоритмы поисковой оптимизации: алгоритмы, вдохновленные природой. М.: МГТУ им. Н.Э. Баумана, 2017. 448 с.
5. Kashan A.H. League championship algorithm: A new algorithm for numerical function optimization // 2009 intern. conf. of soft computing and pattern recognition (Malacca, Malaysia, December 4-7, 2009): Proc. N.Y.: IEEE, 2010. Pp. 43-48. DOI: [10.1109/SoCPaR.2009.21](https://doi.org/10.1109/SoCPaR.2009.21)
6. Kashan A.H., Karimi B. A new algorithm for constrained optimization inspired by the sport league championships // IEEE congress on evolutionary computation (Barcelona, Spain, July 18-23, 2010): Proc. N.Y.: IEEE, 2010. Pp. 487-494. DOI: [10.1109/CEC.2010.5586364](https://doi.org/10.1109/CEC.2010.5586364)
7. Kashan A.H. An efficient algorithm for constrained global optimization and application to mechanical engineering design: League championship algorithm (LCA) // Computer-Aided Design. 2011. Vol. 43. No. 12. Pp. 1769-1792. DOI: [10.1016/j.cad.2011.07.003](https://doi.org/10.1016/j.cad.2011.07.003)
8. Воробьева Е.Ю., Карпенко А.П., Селиверстов Е.Ю. Ко-гибридизация алгоритмов роя частиц // Наука и образование. МГТУ им. Н.Э. Баумана: электрон. журн. 2012. № 4. С. 28. Режим доступа: <http://technomag.edu.ru/doc/355792.html> (дата обращения 16.06.2020).
9. Chattopadhyay S., Murthy C.A., Sankar K. Pal. Fitting truncated geometric distributions in large scale real world networks // Theoretical Computer Science. 2014. Vol. 551. Pp. 22-38. DOI: [10.1016/j.tcs.2014.05.003](https://doi.org/10.1016/j.tcs.2014.05.003)
10. Карпенко А. П., Селиверстов Е.Ю. Обзор методов роя частиц для задачи глобальной оптимизации (particle swarm optimization) // Наука и образование. МГТУ им. Н.Э. Баумана: электрон. журн. 2009. № 3. С. 2-26. DOI: [10.7463/00309.0116072](https://doi.org/10.7463/00309.0116072)
11. Ершов Н.М. Неоднородные клеточные генетические алгоритмы // Компьютерные исследования и моделирование. 2015. Т. 7. № 3. С. 775–780. DOI: [10.20537/2076-7633-2015-7-3-775-780](https://doi.org/10.20537/2076-7633-2015-7-3-775-780)
12. Ямченко Ю.В., Андрусенко А.С. Исследование эффективности алгоритмов непрерывной поисковой оптимизации методом роя частиц // Политехн. молодежный журнал. 2016. № 1. С. 2-16. DOI: [10.18698/2541-8009-2016-1-7](https://doi.org/10.18698/2541-8009-2016-1-7)
13. Карпенко А. П., Свианадзе З.О. Метод мета-оптимизации поисковых алгоритмов оптимизации // Наука и образование. МГТУ им. Н.Э. Баумана: электрон. журн. 2011. № 1. С. 3-36. DOI: [10.7463/0111.0164546](https://doi.org/10.7463/0111.0164546)
14. Остроух Е.Н., Требухин А.В., Чернышев Ю.А., Панасенко П.А. Разработка и анализ гибридного алгоритма решения нелинейных задач оптимизации // Современные наукоемкие технологии. 2018. № 10. С. 87-91. Режим доступа: <http://top-technologies.ru/ru/article/view?id=37200> (дата обращения 17.06.2020).



## Study of League Championship Algorithm Efficiency for Global Optimization Problem

D.O.Zaharov<sup>1,\*</sup>, A.P. Karpenko<sup>1</sup>

<sup>1</sup>Bauman Moscow State Technical University, Moscow, Russia

\* [dimazakharov98@mail.ru](mailto:dimazakharov98@mail.ru)

---

**Keywords:** global optimization, meta-heuristic algorithm, population algorithm, League Championship algorithm, particle swarm optimization algorithm

---

Received: 05.03.2020, Revised: 19.03.2020

---

The article objective is to study a new League Championship Algorithm (LCA) algorithm efficiency by its comparing with the efficiency of the Particle Swarm optimization (PSO) algorithm.

The article presents a brief description of the terms used in the League Championship algorithm, describes the basic rules of the algorithm, on the basis of which the iterative process for solving the global optimization problem is built.

Gives a detailed description of the League Championship algorithm, which comprises a flowchart of the algorithm, as well as a formalization of all its main steps.

Depicts an exhaustive description of the software developed to implement the League Championship algorithm to solve global optimization problems.

Briefly describes the modified particle swarm algorithm. Presents the values of all free parameters of the algorithm and the algorithm modifications, which make it different from the classical version, as well.

The main part of the article shows the results of a great deal of computational experiments using two abovementioned algorithms. All the performance criteria, used for assessment of the algorithms efficiency, are given.

Computational experiments were performed using the spherical function, as well as the Rosenbrock, Rastrigin, and Ackley functions. The results of the experiments are summarized in Tables, and also illustrated in Figures. Experiments were performed for the vector dimension of the variable parameters that is equal to 2, 4, 8, 16, 32, and 64.

An analysis of the results of computational experiments involves a full assessment of the efficiency of the League Championship algorithm, and also provides an answer about expediency for further algorithm development.

It is shown that the League Championship algorithm presented in the article has a high development potential and needs further work for its study.

## References

1. Bo Xing, Wen-Jing Gao. Innovative computational intelligence: a rough guide to 134 clever algorithms. Cham: Springer, 2014. 451 p. DOI: [10.1007/978-3-319-03404-1](https://doi.org/10.1007/978-3-319-03404-1)
2. Blum C., Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 2003, vol. 35, no. 3, pp. 268–308. DOI: [10.1145/937503.937505](https://doi.org/10.1145/937503.937505)
3. Engelbrecht A.P. Computational intelligence: an introduction. 2<sup>nd</sup> ed. Chichester; Hoboken: Wiley, 2007. 597 p.
4. Karpenko A.P. *Sovremennye algoritmy poiskovoj optimizatsii: algoritmy, vdokhnovlennye prirodoj* [Modern search engine optimization algorithms: algorithms inspired by nature]. Moscow: BMSTU Publ., 2017. 448 p. (in Russian).
5. Kashan A.H. League championship algorithm: A new algorithm for numerical function optimization. *2009 intern. conf. of soft computing and pattern recognition* (Malacca, Malaysia, December 4-7, 2009): Proc. N.Y.: IEEE, 2010. Pp. 43-48. DOI: [10.1109/SoCPaR.2009.21](https://doi.org/10.1109/SoCPaR.2009.21)
6. Kashan A.H., Karimi B. A new algorithm for constrained optimization inspired by the sport league championships. *IEEE congress on evolutionary computation* (Barcelona, Spain, July 18-23, 2010): Proc. N.Y.: IEEE, 2010. Pp. 487-494. DOI: [10.1109/CEC.2010.5586364](https://doi.org/10.1109/CEC.2010.5586364)
7. Kashan A.H. An efficient algorithm for constrained global optimization and application to mechanical engineering design: League championship algorithm (LCA). *Computer-Aided Design*, 2011, vol. 43, no. 12, pp. 1769-1792. DOI: [10.1016/j.cad.2011.07.003](https://doi.org/10.1016/j.cad.2011.07.003)
8. Vorob'eva E.Yu., Karpenko A.P., Seliverstov E.Yu. Co-hybridization of PSO. *Nauka i obrazovanie. MGTU im. N.E. Baumana* [Science and Education of the BMSTU], 2012, no. 4, p. 28. Available at: <http://technomag.edu.ru/doc/355792.html>, accessed 16.06.2020 (in Russian).
9. Chattopadhyay S., Murthy C.A., Sankar K. Pal. Fitting truncated geometric distributions in large scale real world networks. *Theoretical Computer Science*, 2014, vol. 551, pp. 22-38. DOI: [10.1016/j.tcs.2014.05.003](https://doi.org/10.1016/j.tcs.2014.05.003)
10. Karpenko A.P., Seliverstov E.Yu. Overview of particle swarm optimization methods for the global optimization problem. *Nauka i obrazovanie MGTU im. N.E. Baumana* [Science and Education of the BMSTU], 2009, no. 3, pp. 2-26. DOI: [10.7463/00309.0116072](https://doi.org/10.7463/00309.0116072) (in Russian)
11. Ershov N.M. Non-uniform cellular genetic algorithms. *Komp'yuternye issledovaniia i modelirovanie* [Computer Research and Modeling], 2015, vol. 7, no. 3, pp. 775–780. DOI: [10.20537/2076-7633-2015-7-3-775-780](https://doi.org/10.20537/2076-7633-2015-7-3-775-780) (in Russian)
12. Yamchenko Yu.V., Andrusenko A.S. Study of continuous search optimization algorithm efficiency by particle swam. *Politekhnicheskij molodezhnyj zhurnal* [Polytechnical Student J. of BMSTU], 2016, no. 1, pp. 2-16. DOI: [10.18698/2541-8009-2016-1-7](https://doi.org/10.18698/2541-8009-2016-1-7) (in Russian)

13. Karpenko A.P., Svianadze Z.O. *Meta-optimization method for search optimization algorithms. Nauka i obrazovanie MGTU im. N.E. Baumana* [Science and Education of the BMSTU], 2011, no. 1, pp. 3-36. DOI: [10.7463/0111.0164546](https://doi.org/10.7463/0111.0164546) (in Russian)
14. Ostroukh E.N., Trebukhin A.V., Chernyshev Yu.A., Panasenko P.A. Development and analysis of the hybrid algorithm for solving nonlinear optimization problems. *Sovremennye naukoemkie tekhnologii* [Modern High Technologies], 2018, no. 10. pp. 87-91. Available at: <http://top-technologies.ru/ru/article/view?id=37200>, accessed 17.06.2020 (in Russian).