

ГИБРИДНЫЙ ЭВОЛЮЦИОННЫЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ ЗАДАЧ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ СВЕРХБОЛЬШОЙ РАЗМЕРНОСТИ

А.В. Вахнин¹

Е.А. Сопов^{1,2}

М.А. Рурич¹

alexeyvah@gmail.com

evgenysopov@gmail.com

mariar8@yandex.ru

¹ СибГУ им. М.Ф. Решетнева, г. Красноярск, Российская Федерация

² СФУ, г. Красноярск, Российская Федерация

Аннотация

При решении прикладных задач различных сфер человеческой деятельности возникает необходимость в поиске наилучшего набора параметров по заданному критерию. Обычно такую задачу формулируют в виде задачи параметрической оптимизации. Рассмотрены задачи оптимизации, представленные моделью типа черного ящика. С ростом размерности таких задач для многих традиционных подходов оптимизации сложно найти удовлетворительное решение даже при значительном увеличении числа вычислений целевой функции. Предложен новый гибридный эволюционный метод координации самонастраивающихся алгоритмов кооперативной коэволюции с локальным поиском COSACC-LS1 для решения задач глобальной вещественной оптимизации сверхбольшой размерности. В основе COSACC-LS1 лежит идея автоматического распределения вычислительных ресурсов между группой самонастраивающихся алгоритмов дифференциальной эволюции, основанных на кооперативной коэволюции, и алгоритмом локального поиска. Оценка эффективности предложенного алгоритма выполнена на 15 эталонных тестовых задачах из набора LSGO CEC 2013. Результаты работы алгоритма на основе COSACC-LS1 сравнивали с современными метаэвристическими алгоритмами, которые разработаны специально для решения задач оптимизации сверхбольшой размерности и являются победителями и призерами соревнований по оптимизации в рамках конференций IEEE CEC. С помощью численных экспериментов продемон-

Ключевые слова

Оптимизация сверхбольшой размерности, самонастраивающийся эволюционный алгоритм, дифференциальная эволюция, кооперативная коэволюция

стрировано, что по критерию средней точности Поступила 22.12.2022
найденного решения предложенный алгоритм луч- Принята 13.02.2023
ше большинства других популярных алгоритмов © Автор(ы), 2023

*Работа выполнена при поддержке Минобрнауки
России (грант № 075-15-2022-1121)*

Введение. Многие современные задачи глобальной вещественной оптимизации содержат большое число переменных, которое постоянно растет. Некоторые задачи представлены моделью типа черного ящика. Такие задачи сложны для решения, поскольку исчерпывающий анализ свойств целевой функции невозможен или затруднен, а свойства, которые удается выявить, не способствуют повышению эффективности решения задачи. Многие эволюционные алгоритмы оптимизации, в том числе и метаэвристические, демонстрируют снижение эффективности работы с увеличением числа переменных. Задачи вещественной глобальной оптимизации со 100 и более переменными можно отнести к классу задач сверхбольшой размерности. В иностранной литературе приведены подобные задачи, содержащие более тысячи переменных, соответствующий им класс называется Large-Scale Global Optimization (LSGO).

В общем виде задача глобальной оптимизации сверхбольшой размерности (задача LSGO) может быть сформулирована следующим образом:

$$f(x_1, x_2, \dots, x_N) \rightarrow \min_{x \in R^N}, \quad f: R^N \rightarrow R^1; \quad (1)$$

$$x_i \in [a_i, b_i], \quad i = \overline{1, N}, \quad (2)$$

где f — целевая функция; x_i — i -я вещественная переменная; $[a_i, b_i]$ — интервал поиска для i -й переменной; N — размерность задачи. На тип целевой функции не накладываются ограничения (линейность, непрерывность, дифференцируемость или выпуклость). Целевая функция задается алгоритмически, какая-либо априорная информация о ее свойствах и свойствах ее ландшафта отсутствует.

В настоящей работе предложен новый самонастраивающийся гибридный алгоритм, который совмещает в себе три подхода: алгоритм кооперативной коэволюции (Cooperative Coevolution, CC); самонастраивающийся алгоритм дифференциальной эволюции; алгоритм локального поиска. Предложенный алгоритм получил название — координация самонастраивающихся алгоритмов кооперативной коэволюции с локальным поиском (Coordination of Self-Adaptive Cooperative Coevolution Algorithms, COSACC-LS1).

Обзор современных биоинспирированных подходов к решению задач оптимизации сверхбольшой размерности. Сложность практических задач LSGO за последние три десятка лет существенно выросла и наблюдается положительная тенденция роста. При решении многих прикладных задач глобальной оптимизации исследователи часто прибегают к использованию различных биоинспирированных алгоритмов (или алгоритмов, вдохновленных природой). Такие алгоритмы включают в себя совокупность нескольких классов, к которым можно отнести эволюционные алгоритмы, алгоритмы роевого интеллекта и др. [1, 2]. Отметим, что эволюционные алгоритмы являются наиболее популярными, применяемыми в задачах глобальной оптимизации алгоритмически заданных целевых функций. Применение традиционных биоинспирированных алгоритмов для решения задач LSGO становится нецелесообразным из-за снижения их эффективности с увеличением размерности задачи. Одним из первых обсуждений задач LSGO можно считать специальную сессию конференции IEEE CEC в 2008 г. [3], на которой было предложено три эталонных набора задач для оценки эффективности работы алгоритмов: LSGO CEC 2008, 2010 и 2013 [4–6]. В 2018 г. был предложен инструментарий для автоматического сравнения эффективности алгоритмов TACO [7]. Согласно принятой систематизации [8–11] выделяют следующие основные подходы к решению задач данного типа:

- модификация стандартных операторов эволюционного алгоритма без использования декомпозиции вектора решения;
- статическая декомпозиция вектора решения с использованием кооперативной коэволюции;
- случайная декомпозиция вектора решения с использованием кооперативной коэволюции;
- группировка переменных и декомпозиция вектора решения на основе обучения с использованием кооперативной коэволюции.

Гибридные меметические алгоритмы демонстрируют высокую эффективность работы для задач LSGO всех классов из эталонного набора. Алгоритм SHADE-ILS [12] является победителем соревнований по решению задач LSGO в 2018 г. и удерживает лидерство и в настоящее время. Алгоритм MOS [13] был абсолютным лидером в течение 5 лет — с 2013 по 2018 гг.

Модификация стандартных операторов эволюционного алгоритма без использования декомпозиции вектора решения. Алгоритм DMS-PSO [14] — это один из первых алгоритмов, эффективность которого оценена при решении задач из эталонного набора LSGO CEC 2008. Алгоритм C-DEEPSO [15] основан на комбинации алгоритма дифференциальной

эволюции и метода роя частиц (PSO). Алгоритм VMODE [16] использует стандартный алгоритм дифференциальной эволюции в качестве алгоритма глобальной оптимизации. Алгоритм MTS [17] состоит из алгоритмов MTS-LS1, MTS-LS2, MTS-LS3. В основе каждого алгоритма лежит по координатный поиск с заранее определенной длиной шагов. Несмотря на простоту идеи, алгоритм MTS демонстрирует высокую эффективность работы при решении задач LSGO. Алгоритм MTS-LS1 используется как алгоритм локального поиска во многих гибридных меметических алгоритмах. Алгоритм IHDELS [18] — первый меметический алгоритм, который занял призовое (второе) место в соревновании по решению задач LSGO в 2015 г. в рамках конференции IEEE CEC, в его основе лежат алгоритмы SaDE [19], L-BFGS-B [20] (локального поиска) и MTS-LS1. Версия алгоритма MOS (2013) [21] реализует гибрид, состоящий из алгоритмов MTS-LS1, Solis and Wets и генетического алгоритма. Алгоритм SHADE-ILS [12] использует комбинацию самонастраивающегося алгоритма SHADE [22] для глобального поиска и алгоритмов MTS-LS1, L-BFGS-B для локального поиска. Алгоритм MPS-CMA-ES [23] занял второе место в соревновании по решению задач LSGO на конференции IEEE CEC в 2019 г.

Статическая декомпозиция вектора решения с использованием кооперативной коэволюции. Впервые кооперативная коэволюция применена для стандартного генетического алгоритма в работе [24]. Предложены два варианта реализации кооперативной коэволюции: алгоритмы CCGA-1 и CCGA-2. Число субкомпонент (подзадач) равно числу переменных. В алгоритме CCGA-1 каждая субкомпонента оптимизируется собственным эволюционным алгоритмом с использованием лучших результатов из других субкомпонент. Алгоритм CCGA-2 применяет случайные индивиды из других субкомпонент для формирования целого вектора решения. Эффективность работы алгоритмов оценивалась на пяти классических задачах оптимизации, максимальное число переменных достигало 30. Численные эксперименты продемонстрировали, что эффективность работы алгоритмов CCGA-1 и CCGA-2 выше, чем эффективность работы генетического алгоритма.

Случайная декомпозиция вектора решения с использованием кооперативной коэволюции. Алгоритм DECC-G [25] реализует комбинацию подхода случайной группировки и алгоритма оптимизации SaNSDE [26]. Алгоритм DECC-G является эталоном (оценкой снизу) для сравнения с другими аналогичными подходами. Алгоритм MLSHADE-SPA [27] использует гибрид из трех алгоритмов глобальной оптимизации и одного алгоритма локального поиска.

Группировка переменных и декомпозиция вектора решения на основе обучения с использованием коэволюционной коэволюции. Основная идея подходов к алгоритмам из данной группы состоит в определении взаимосвязей между переменными и объединении их в субкомпоненты для дальнейшей оптимизации. В алгоритме DECC-DML [28] применяется дельта-группировка. Алгоритм DECC-DG [29] и модифицированный алгоритм DECC-DG2 [30] используют алгоритм SaNSDE для оптимизации найденных субкомпонент. В алгоритме DGSC [31] переменные объединяются в субкомпоненты с помощью алгоритма, основанного на кластерном анализе. Некоторые оригинальные подходы по группировке переменных предложены в следующих алгоритмах: CC-CMA-ES [32], SACC [33], BICCA [34] и SGCC [35]. Алгоритм RDG [36] улучшает уже ранее предложенную группировку переменных [29]. Алгоритм CC-RDG3 [37] использует комбинацию подходов алгоритмов CMA-ES и группировки переменных RDG для эффективного нахождения субкомпонент с пересекающимися группами переменных. Алгоритм CC-RDG3 занял первое место в соревновании по решению задач LSGO на конференции IEEE CEC в 2019 г.

Большинство лидирующих алгоритмов для решения задач глобальной оптимизации сверхбольшой размерности включают в себя три компонента: коэволюционную коэволюцию (CC), разновидность дифференциальной эволюции в качестве алгоритма глобального поиска (DE) и алгоритм локального поиска для уточнения решений (LS). В табл. 1 приведены участники соревнований по решению задач LSGO на конференциях IEEE CEC в разные годы, в последних трех столбцах — компоненты CC, DE и LS, из которых состоит алгоритм.

Таблица 1

Алгоритмы-победители и участники соревнований по решению задач LSGO

Алгоритм	Год участия	Место	Компоненты алгоритма		
			CC	DE	LS
CC-CMA-ES	2015	3	+	-	-
DEEPSO	2015	-	-	+	-
IHDELS	2015	2	-	+	+
SACC	2015	-	+	+	+
VMODE	2015	-	-	+	-
BICCA	2018	-	+	-	-
MPS	2019	2	-	-	-
SGCC	2019	3	+	+	-
DECC-G	2015, 2018	-	+	+	-
MOS	2015, 2018	1, 3	-	+	+

Окончание табл. 1

Алгоритм	Год участия	Место	Компоненты алгоритма		
			CC	DE	LS
MLSHADE-SPA	2018	2	–	+	+
SHADE-ILS	2018	1	–	+	+
CC-RDG3	2019	1	+	+	–
DGSC	2019	–	+	+	–

В обзорной части рассмотрены два основных направления решения задач сверхбольшой размерности: подходы без декомпозиции и с декомпозицией решения. Подходы на основе декомпозиции обычно демонстрируют большую эффективность. Декомпозиция (группировка переменных) бывает трех типов: статическая, случайная и на основе обучения. Методы статической группировки эффективны только для сепарабельных задач и требуют априорной информации для формирования групп переменных, поэтому практически не используются. Методы случайной группировки одинаково эффективны на сепарабельных и несепарабельных задачах, но не используют обратную связь для адаптации способа группировки. Методы, основанные на обучении, эффективно формируют группы переменных на основе анализа процесса оптимизации, но весьма затратные в плане вычислений целевой функции. В некоторых работах показано, что на разных этапах процесса оптимизации выгодно использовать различные группы, а методы на основе обучения формируют единственный вариант декомпозиции. В результате анализа преимуществ и недостатков каждого типа декомпозиции для дальнейшего развития выбран подход на основе декомпозиции, использующий случайную группировку переменных.

При рассмотрении ведущих алгоритмов и алгоритмов-победителей соревнований по решению задач глобальной оптимизации сверхбольшой размерности выявлено, что эти алгоритмы используют в различных сочетаниях три компонента: кооперативную коэволюцию при декомпозиции решения, дифференциальную эволюцию в качестве алгоритма глобального поиска и локальный поиск для уточнений решения. В настоящей работе предложен новый алгоритм COSACC-LS1, который для реализации декомпозиции на основе случайной группировки переменных объединяет три компонента (кооперативную и дифференциальную эволюции, а также локальный поиск).

Предложенный алгоритм решения задач оптимизации сверхбольшой размерности. Предлагаемый подход использует адаптивное приме-

нение различного числа субкомпонент. В основе лежит динамическое перераспределение вычислительных ресурсов. Как было показано в работах [38, 39], применение различного числа субкомпонент на разных этапах оптимизации показывает себя лучше, чем статичное значение в течение всего процесса оптимизации. В предлагаемом алгоритме целевую функцию оптимизируют несколько алгоритмов, которые имеют разное число субкомпонент. В конце каждого цикла происходит перераспределение ресурсов между алгоритмами, в рассматриваемом случае ресурс — это число поколений. Больше ресурса получает тот алгоритм, который продемонстрировал лучшую эффективность на прошлых итерациях. Набор субкомпонент (из M штук) определяется как множество $\{CC_1, CC_2, \dots, CC_M\}$, где $CC_1 \cap CC_2 \cap \dots \cap CC_M = \emptyset$. Сначала каждый алгоритм получает одинаковое количество ресурса ($G_i, i = 1, \dots, M$). Далее начинается последовательное применение алгоритмов с разным числом субкомпонент. В каждом новом цикле последовательность применения алгоритмов случайная, так как неизвестно, какой алгоритм покажет лучшую эффективность. После того как ресурсы всех алгоритмов закончились, необходимо выполнить перерасчет числа поколений. Эффективность каждого алгоритма вычисляется по формуле:

$$performance_i = \left(\frac{best_fitness_before_i - best_fitness_after_i}{best_fitness_after_i} \right) \frac{1}{G_i}, \quad (3)$$

где $best_fitness_before_i$ и $best_fitness_after_i$ — значения фитнес-функции найденного лучшего решения до и после применения i -го алгоритма, $i = 1, \dots, M$.

Необходимо выполнить нормировку по числу использованных поколений, чтобы сравнение было корректным. Значение, характеризующее улучшение, необходимо разделить на число поколений G_i , которое затратил каждый алгоритм CC-SHADE. После основного цикла оптимизации алгоритмы ранжируются по их эффективности. Лучший алгоритм получает дополнительно G_{add} поколений, данное значение вычисляется как сумма G_{lose} поколений, которые потеряли другие алгоритмы. Для всех алгоритмов CC-SHADE заранее предопределено пороговое число поколений G_{min} . После перераспределения числа поколений для уточнения найденного лучшего решения применяется алгоритм MTS-LS1, новое лучшее решение часто оказывается далеко в пространстве поиска от других решений-индивидов из основной популяции. Алгоритм CC-SHADE не всегда может сразу улучшить решение, но он улучшает среднее значение функции

пригодности для всей популяции. Для корректной оценки эффективности применения алгоритмов CC-SHADE модифицирована формула (3). Вместо найденного лучшего решения используется медианное значение функции пригодности из популяции:

$$performance_new_i = \left(\frac{median_fitness_before_i - median_fitness_after_i}{median_fitness_after_i} \right) \frac{1}{G_i}, \quad (4)$$

где $median_fitness_before_i$ и $median_fitness_after_i$ — медианные значения функции пригодности в популяции до и после применения алгоритма CC-SHAD_i, $i = 1, \dots, M$. Подход к перерасчету числа поколений включает в себя формулы:

$$PER = \left\{ i : performance_new_i = \max_{j=1, \dots, M} \{ performance_new_j \} \right\}; \quad (5)$$

$$pool = \sum_{j=1}^M \begin{cases} G_{lose}, & \text{если } (G_i - G_{lose} \geq G_{min}) \wedge i \notin PER; \\ 0, & \text{иначе;} \end{cases} \quad (6)$$

$$G_i = \begin{cases} G_i + G_{win}, & \text{если } i \in PER; \\ G_i - G_{lose}, & \text{если } (G_i - G_{lose} \geq G_{min}) \wedge i \notin PER; \\ G_{min}, & \text{если } (G_i - G_{lose} < G_{min}) \wedge i \notin PER, \end{cases} \quad (7)$$

где PER — индекс алгоритма CC-SHADE, который показал лучшую эффективность в предыдущем цикле; $pool$ — сумма вычтенных поколений у алгоритмов, которые показали худшую эффективность. Если лучшие эффективности хотя бы одной пары алгоритмов равны, то увеличивается число поколений у алгоритма с наименьшим числом субкомпонент, так как в случае применения алгоритма с наименьшим числом субкомпонент будет использовано меньшее число оценок функции пригодности.

В качестве алгоритма глобальной оптимизации субкомпонент для кооперативной коэволюции выбран алгоритм SHADE из-за его высокой эффективности работы, подтвержденной численными экспериментами. В оригинальном алгоритме SHADE используется схема мутации $current - to - pbest / 1$. Для увеличения эффективности работы алгоритма SHADE протестированы различные схемы мутации, лучшую эффективность показала схема

$$v_{i,G} = x_{i,G} + F(x_{pbest,G} - x_{i,G}) + F(x_{t,G} - x_{r_2,G}), \quad (8)$$

где v_i — пробный вектор; x_i — родительский вектор; x_{pbest} — индивид, выбранный из доли p лучших индивидов по равномерному закону распределения; x_{r_2} — случайно выбранный индивид из объединения текущей популяции и архива решений, $i \neq pbest \neq r_1 \neq r_2$; x_t — индивид, выбранный с помощью турнирной селекции, размер турнира равен двум, $i \neq pbest \neq t \neq r_2$.

Важный параметр, влияющий на эффективность работы и необходимый для настройки, — это размер популяции, его выбирают произвольно, исходя из размерности пространства поиска и ограничений на число вычислений целевой функции. В настоящей работе применяется адаптивный механизм расчета популяции, который основан на разнообразии индивидов [40]. Разнообразие в популяции рассчитывается по формуле

$$DI = \sqrt{\frac{1}{NP} \sum_{i=1}^{NP} \sum_{j=1}^n (x_{ij} - \bar{x}_j)^2}, \quad (9)$$

где NP — размер популяции; n — размерность пространства поиска; \bar{x}_j — среднее значение всех индивидов в популяции по j -й координате.

После каждого цикла обновляется текущий размер популяции:

$$RD = \frac{DI}{DI_{init}}; \quad (10)$$

$$RFES = \frac{FEVs}{\max FEVs}; \quad (11)$$

$$rRD = 1 - \frac{RFES}{0,9}; \quad (12)$$

$$NP = \begin{cases} NP + 1, & \text{если } (NP + 1 \leq \max NP) \wedge (RD < 0,9rRD); \\ NP - 1, & \text{если } (NP - 1 \geq \min NP) \wedge (RD > 11rRD); \\ NP, & \text{иначе,} \end{cases} \quad (13)$$

где RD — текущее относительное разнообразие популяции; $RFES$ — доля оставшегося ресурса работы алгоритма; $FEVs$ и $\max FEVs$ — число оставшихся и максимальное допустимое число вычислений функции пригодности; rRD — ожидаемое значение относительного разнообразия популяции RD ; $\min NP$ и $\max NP$ — минимальное и максимальное допустимое число индивидов. Добавление случайного индивида происхо-

дит с помощью случайной генерации. Размер популяции уменьшается путем удаления худшего индивида из основной популяции и перемещения его во внешний архив.

Изложенные идеи реализованы в алгоритме COSACC-LS1. Алгоритм имеет наиболее важный и существенный гиперпараметр — множество M алгоритмов с разным числом субкомпонент. Из-за высокой вычислительной сложности задач протестировано только восемь различных комбинаций субкомпонент (выбранных на основе опыта и рациональных предположений). Во всех случаях число алгоритмов CC-SHADE равно трем: $\{1, 2, 4\}$, $\{1, 2, 8\}$, $\{1, 2, 10\}$, $\{1, 4, 8\}$, $\{1, 4, 10\}$, $\{1, 8, 10\}$, $\{2, 4, 8\}$ и $\{2, 4, 10\}$. Начальное значение числа поколений для всех алгоритмов $G_{init} = 20$, минимальное число поколений для каждого алгоритма $G_{min} = 5$. После завершения основного цикла лучшему алгоритму добавляется $M - 1$ поколение. Остальные алгоритмы теряют по одному поколению. Начальное число индивидов равно 100, минимальное и максимальное — 25 и 150. После того как алгоритм исчерпал 90 % бюджета вычислений функции пригодности, популяция становится равной ее минимальному значению [40]. Псевдокод алгоритма COSACC-LS1 состоит из следующих шагов.

Шаг 0. Задать число алгоритмов M для кооперативной коэволюции, число субкомпонент для каждого алгоритма, $\min NP$, $\max NP$, n , G_{init} , G_{min} , G_{lose} , $\max FEVs$.

Шаг 1. Инициализировать популяцию.

Шаг 2. Вычислить DI_{init} (9).

Шаг 3. Для всех алгоритмов инициализировать число поколений, $G_i = G_{init}$, $i = 1, \dots, M$.

Шаг 4. Если допустимое число вычислений функции больше нуля, то перейти к шагу 5, иначе — к шагу 13.

Шаг 5. Сгенерировать случайную последовательность применения алгоритмов и случайно распределить переменные в субкомпоненты.

Шаг 6. Если не все алгоритмы израсходовали свой ресурс поколений, то перейти к шагу 7, иначе — к шагу 10.

Шаг 7. Применить операторы алгоритма CC-SHADE для популяции.

Шаг 8. Вычислить новый размер популяции (13).

Шаг 9. Вычислить PER (5) для алгоритма CC-SHADE, вернуться к шагу 6.

Шаг 10. Вычислить новое число поколений для всех алгоритмов (7).

Шаг 11. Применить алгоритм MTS-LS1 к найденному лучшему решению.

Шаг 12. Перейти к шагу 4.

Шаг 13. Вывод найденного лучшего решения.

Результаты численных экспериментов. Эффективность предложенного алгоритма COSACC-LS1 оценивалась при решении задач из эталонного тестового набора LSGO CEC 2013 [6]. Набор состоял из 15 задач вещественной оптимизации (минимизации) размерностью 1000 переменных. Задачи наделены специальными свойствами, затрудняющими работу поисковых алгоритмов, и поделены на пять классов: F1–F3 — полностью сепарабельные задачи; F4–F7 и F8–F11 — задачи с сепарабельной и без сепарабельной субкомпоненты; F12–F14 — задачи с пересекающимися субкомпонентами; F15 — полностью несепарабельные задачи. Эффективность алгоритмов LSGO на всех эталонных задачах определялась как ошибка между найденным лучшим решением и истинным значением глобального минимума, усредненная по 25 независимым запускам. Максимальное разрешенное число вычислений функции для независимого запуска составляло $3,0E + 06$. Алгоритм COSACC-LS1 реализован на языке программирования C++ и выполнялся на высокопроизводительном вычислительном кластере. В табл. 2 приведены результаты (медиана, среднее и среднеквадратическое отклонение (СКО)) лучшей настройки алгоритма COSACC-LS1 при $M = \{1, 2, 4\}$.

Таблица 2

**Результаты работы лучшей настройки алгоритма COSACC-LS1
(конференция IEEE CEC в 2013 г.)**

Задача	F1	F2	F3	F4	F5
Медиана	0,00E + 00	1,11E + 03	2,00E + 01	1,05E + 09	1,18E + 06
Среднее	0,00E + 00	1,11E + 03	2,00E + 01	1,32E + 09	1,18E + 06
СКО	0,00E + 00	5,80E + 01	1,32E – 04	7,80E + 08	1,78E + 05
Задача	F6	F7	F8	F9	F10
Медиана	1,04E + 06	2,28E + 03	5,25E + 13	1,37E + 08	9,26E + 07
Среднее	1,04E + 06	1,59E + 04	5,37E + 13	1,35E + 08	9,26E + 07
СКО	2,67E + 03	5,33E + 04	2,88E + 13	1,87E + 07	2,81E + 05
Задача	F11	F12	F13	F14	F15
Медиана	1,85E + 06	7,77E – 04	1,17E + 06	7,96E + 06	9,12E + 05
Среднее	3,18E + 06	1,03E + 01	3,39E + 06	1,13E + 07	9,51E + 05
СКО	3,82E + 06	3,17E + 01	7,53E + 06	1,58E + 07	2,97E + 05

Усредненный график сходимости алгоритма COSACC-LS1 приведен на рис. 1, а. По оси абсцисс (рис. 1, а–в) отложено число вычислений функции пригодности. Ось ординат на рис. 1, а показывает значение функции пригодности, непрерывная красная кривая — усредненное по 25 независимым запускам значение функции пригодности, оранжевая область — СКО. Динамика изменения числа индивидов во время процесса оптимизации приведена на рис. 1, б, где ось ординат — текущее число индивидов в популяции. Синяя кривая показывает усредненное число индивидов по всем запускам, бирюзовая область — СКО числа индивидов. С начала процесса оптимизации число индивидов постоянно растет и достигает максимального значения 150. Алгоритм с первых этапов работы демонстрирует высокую сходимость, при которой текущее разнообразие популяции становится меньше, чем ожидаемое, следовательно, алгоритм добавляет индивидов для поддержания разнообразия в популяции. Динамика изменения числа ресурсов внутри алгоритма COSACC-LS1 приведена на рис. 1, в. Для сокращения описания использована следующая нотация, $CC-i$, где CC — алгоритм CC -SHADE, i — число субкомпонент, ось ординат — доля поколений от суммы всех поколений у алгоритмов. Каждый цвет соответствует своему алгоритму, штриховые линии — это усредненные значения по всем независимым запускам, область вокруг штриховых линий — СКО числа поколений. Согласно усредненным значениям примерно до половины бюджета вычислений функции пригодности приоритет отдается алгоритму ($CC-4$) с самым большим числом субкомпонент, после чего приоритет смещается в пользу того алгоритма ($CC-1$), который не использует декомпозицию. Как отмечалось ранее, численные эксперименты демонстрируют, что на первых этапах поиска лучше всего применять декомпозицию с разбиением на группы с малым числом переменных, а ближе к концу процесса оптимизации работать с целым вектором решения. В связи с ограничением на объем статьи в деталях описана динамика только одной функции набора, однако аналогичные результаты наблюдаются и во всех остальных задачах из набора.

Сравнение эффективности алгоритмов с лучшей настройкой алгоритма COSACC-LS1 выполнено с использованием базы данных алгоритмов TACO [7]. Ранее приведены различные методы работы с задачами сверхбольшой размерности. Известные алгоритмы редко используют только один подход, большинство лидирующих по эффективности алгоритмов объединяют несколько идей. Поэтому для сравнения подходов нет смысла выбирать методы по выделенным ранее типам, так как известно, что некоторые подходы менее эффективны, но могут быть основой гибридного алгоритма.

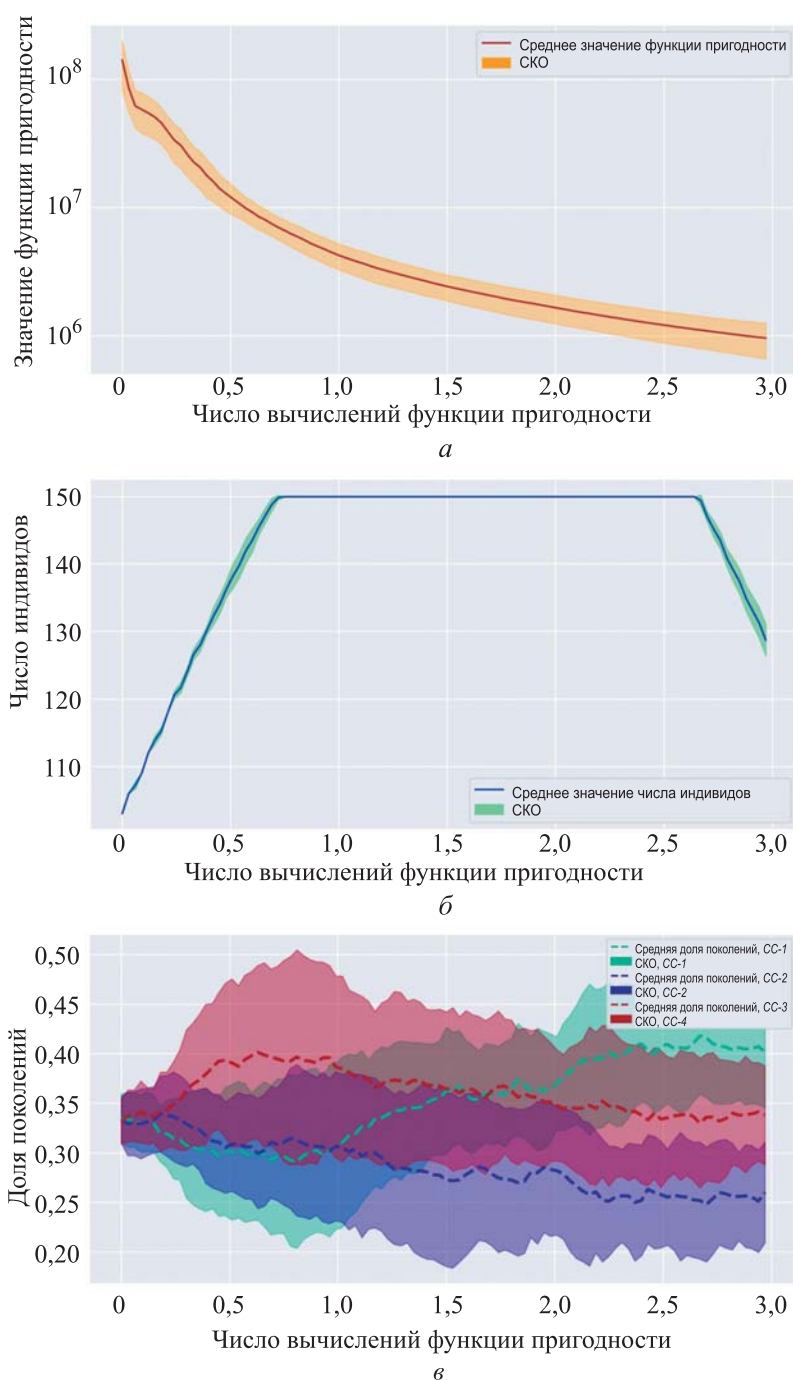
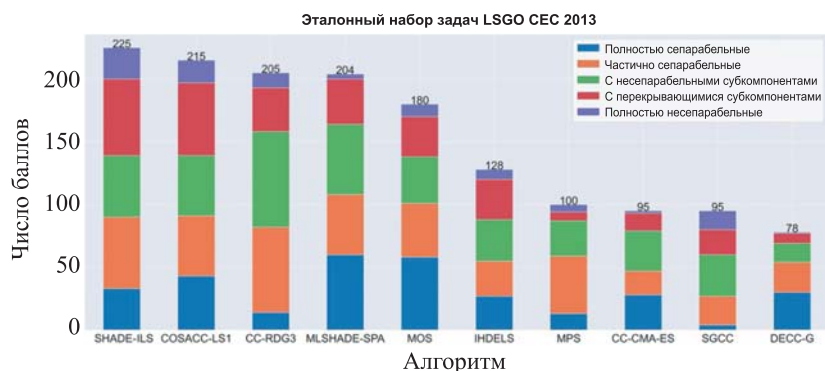


Рис. 1. Динамика работы алгоритма COSACC-LS1 при решении задачи F15 из набора LSGO CEC 2013:

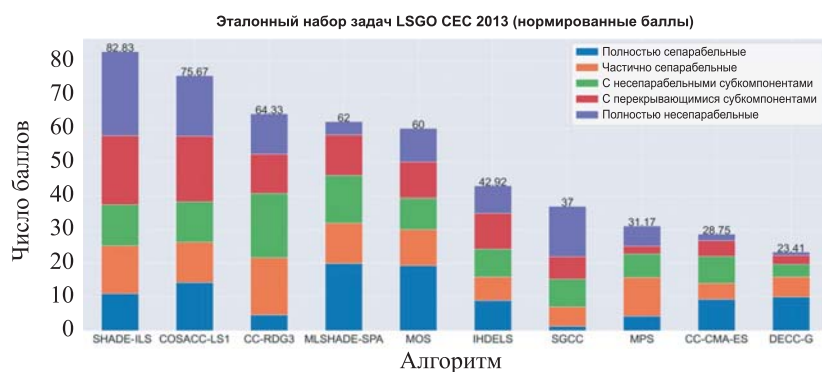
а — график сходимости алгоритма COSACC-LS1; *б* — динамика изменения числа индивидов во время процесса оптимизации; *в* — распределение ресурсов внутри алгоритма COSACC-LS1

Для сравнения взяты все алгоритмы, которые занимали призовые места на соревнованиях по решению задач LSGO в рамках конференций IEEE CEC, среди них следующие алгоритмы: CC-CMA-ES, CC-RDG3, IHDELS, MLSHADE-SPA, MOS, MPS, SGCC и SHADE-ILS (см. табл. 1).

Кроме того, для сравнения добавлен алгоритм DECC-G, поскольку он часто используется как нижняя оценка эффективности алгоритмов LSGO. Результаты сравнения алгоритмов в виде столбчатой диаграммы приведены на рис. 2, а. По оси абсцисс отложены алгоритмы в порядке убывания набранных ими баллов на основе рангов, по оси ординат — общая сумма баллов. Баллы начисляются, как в чемпионате «Формула-1». Разные цвета участков столбчатых диаграмм отображают различные классы задач, высота столбцов — число набранных баллов. Алгоритм COSACC-LS1 занимает второе место среди современных алгоритмов LSGO. Полученные алгоритмами баллы нормировали на каждом классе задач, разделив число баллов на число задач (из-за неодинакового числа задач каждого класса). Результаты приведены на рис. 2, б.



а



б

Рис. 2. Сравнение результатов работы алгоритма COSACC-LS1 и других алгоритмов при решении задач LSGO из эталонного набора LSGO CEC 2013 (а) и набора LSGO CEC 2013 с нормированными баллами (б)

Заключение. Предложен новый подход к автоматическому перераспределению ресурсов между поисковыми алгоритмами, на основе которого реализован и исследован алгоритм COSACC-LS1 для решения задач оптимизации сверхбольшой размерности. Оценена эффективность его работы при решении задач из набора LSGO CEC 2013, выполнено сравнение с другими популярными алгоритмами. Предлагаемый алгоритм лучший из всех алгоритмов-участников соревнований за прошлые годы, кроме алгоритма SHADE-ILS. Общая идея подхода к решению задач оптимизации сверхбольшой размерности не учитывает специфики конкретных алгоритмов глобального и локального поисков, поэтому можно применять любые другие алгоритмы, что даст потенциал для улучшения эффективности. Взаимодействие трех алгоритмов-компонентов, основанных на кооперативной коэволюции, демонстрирует высокую эффективность за счет адаптивного распределения ресурсов в зависимости от решаемой задачи и конкретной ситуации. Путем визуализации распределения ресурсов показано, как алгоритм подстраивается под новую среду.

Несмотря на то, что эталонный тестовый набор LSGO CEC 2013 содержит множество задач с различными свойствами, большая часть реальных задач оптимизации сверхбольшой размерности изучена недостаточно хорошо, в результате может потребоваться более тонкая настройка параметров алгоритма COSACC-LS1. В дальнейшем будет рассмотрен вопрос разработки подхода для самоадаптации внутренних параметров алгоритмов-компонентов.

ЛИТЕРАТУРА

- [1] Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. М., Изд-во МГТУ им. Н.Э. Баумана, 2021.
- [2] Del Ser J., Osaba E., Molina D., et al. Bio-inspired computation: where we stand and what's next. *Swarm Evol. Comput.*, 2019, vol. 48, pp. 220–250. DOI: <https://doi.org/10.1016/j.swevo.2019.04.008>
- [3] Tang K. Summary of results on CEC'08 competition on large-scale global optimization. China, USTC, 2008.
- [4] Tang K., Yao X., Suganthan P.N., et al. Benchmark functions for the CEC'2008 special session and competition on large-scale global optimization. China, USTC, 2007.
- [5] Tang K., Li X., Suganthan P.N., et al. Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization China, USTC, 2008.
- [6] Li X., Tang K., Omidvar M.N., et al. Benchmark functions for the CEC'2013 special session and competition on large-scale global optimization. Australia, RMIT University, 2013.

- [7] Molina D., La Torre A. Toolkit for the automatic comparison of optimizers: comparing large-scale global optimizers made easy. *Proc. IEEE CEC*, 2018.
DOI: <https://doi.org/10.1109/CEC.2018.8477924>
- [8] Mahdavi S., Shiri M.E., Rahnamayan S. Metaheuristics in large-scale global continuous optimization: a survey. *Inf. Sc.*, 2015, vol. 295, pp. 407–428.
DOI: <https://doi.org/10.1016/j.ins.2014.10.042>
- [9] Singh A., Dulal N. A survey on metaheuristics for solving large-scale optimization problems. *Int. J. Comput. Appl.*, 2017, vol. 170, no. 5, pp. 1–7.
DOI: <https://doi.org/10.5120/ijca2017914839>
- [10] Omidvar M.N., Li X., Yao X. A review of population-based metaheuristics for large-scale black-box global optimization. Part I. *IEEE Trans. Evol. Comput.*, 2022, vol. 26, no. 5, pp. 802–822. DOI: <https://doi.org/10.1109/TEVC.2021.3130838>
- [11] Omidvar M.N., Li X., Yao X. A review of population-based metaheuristics for large-scale black-box global optimization. Part II. *IEEE Trans. Evol. Comput.*, 2022, vol. 26, no. 5, pp. 823–843. DOI: <https://doi.org/10.1109/TEVC.2021.3130835>
- [12] Molina D., La Torre A., Herrera F. SHADE with iterative local search for large-scale global optimization. *Proc. IEEE CEC*, 2018.
DOI: <https://doi.org/10.1109/CEC.2018.8477755>
- [13] La Torre A., Muelas S., Pena J.-M. Multiple offspring sampling in large-scale global optimization. *Proc. IEEE CEC*, 2012. DOI: <https://doi.org/10.1109/CEC.2012.6256611>
- [14] Zhao S.Z., Liang J.J., Suganthan P.N., et al. Dynamic multi-swarm particle swarm optimizer with local search for large-scale global optimization. *Proc. IEEE CEC*, 2008, pp. 3845–3852. DOI: <https://doi.org/10.1109/CEC.2008.4631320>
- [15] Marcelino C., Almeida P., Pedreira C., et al. Applying C-DEEPSO to solve large-scale global optimization problems. *Proc. IEEE CEC*, 2018.
DOI: <https://doi.org/10.1109/CEC.2018.8477854>
- [16] Lopez E., Puris A., Bello R. VMODE: a hybrid metaheuristic for the solution of large-scale optimization problems. *Investig. Operacional*, 2015, vol. 36, no. 3, pp. 232–239.
- [17] Tseng L.-Y., Chen C. Multiple trajectory search for large-scale global optimization. *Proc. IEEE CEC*, 2018, pp. 3052–3059.
DOI: <https://doi.org/10.1109/CEC.2008.4631210>
- [18] Molina D., Herrera F. Iterative hybridization of DE with local search for the CEC'2015 special session on large-scale global optimization. *Proc. IEEE CEC*, 2015, pp. 1974–1978. DOI: <https://doi.org/10.1109/CEC.2015.7257127>
- [19] Qin A.K., Huang V.L., Suganthan P.N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.*, 2009, vol. 13, no. 2, pp. 398–417. DOI: <https://doi.org/10.1109/TEVC.2008.927706>
- [20] Morales J., Nocedal J. Remark on “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization”. *ACM Trans. Math. Softw.*, 2011, vol. 38, no. 1, pp. 1–4. DOI: <https://doi.org/10.1145/2049662.2049669>

- [21] La Torre A., Muelas S., Pena J.-M. Large-scale global optimization: experimental results with MOS-based hybrid algorithms. *Proc. IEEE CEC*, 2013, pp. 2742–2749. DOI: <https://doi.org/10.1109/CEC.2013.6557901>
- [22] Tanabe R., Fukunaga A. Evaluating the performance of SHADE on CEC'2013 benchmark problems. *Proc. IEEE CEC*, 2013, pp. 1952–1959. DOI: <https://doi.org/10.1109/CEC.2013.6557798>
- [23] Bolufe-Rohler A., Fiol-Gonzalez S., Chen S. A minimum population search hybrid for large-scale global optimization. *Proc. IEEE CEC*, 2015, pp. 1958–1965. DOI: <https://doi.org/10.1109/CEC.2015.7257125>
- [24] Potter M., De Jong K. A cooperative coevolutionary approach to function optimization. *PPSN 1994*. Berlin, Springer Verlag, 1994, pp. 245–257. DOI: https://doi.org/10.1007/3-540-58484-6_269
- [25] Yang Z., Tang K., Yao X. Large-scale evolutionary optimization using cooperative coevolution. *Inf. Sc.*, 2008, vol. 178, no. 15, pp. 2985–2999. DOI: <https://doi.org/10.1016/j.ins.2008.02.017>
- [26] Yang Z., Tang K., Yao X. Self-adaptive differential evolution with neighborhood search. *Proc. IEEE CEC*, 2008, pp. 1110–1116. DOI: <https://doi.org/10.1109/CEC.2008.4630935>
- [27] Hadi A., Wagdy A., Jambi K. LSHADE-SPA memetic framework for solving large-scale optimization problems. *Complex Intell. Syst.*, 2019, vol. 5, no. 1, pp. 25–40. DOI: <https://doi.org/10.1007/s40747-018-0086-8>
- [28] Omidvar M.N., Li X., Yao X. Cooperative coevolution with delta grouping for large-scale non-separable function optimization. *Proc. IEEE CEC*, 2010, pp. 1–8. DOI: <https://doi.org/10.1109/CEC.2010.5585979>
- [29] Omidvar M.N., Li X., Mei Y., et al. Cooperative coevolution with differential grouping for large-scale optimization. *IEEE Trans. Evol. Comput.*, 2014, vol. 18, no. 3, pp. 378–393. DOI: <https://doi.org/10.1109/TEVC.2013.2281543>
- [30] Omidvar M.N., Yang M., Mei Y., et al. DG2: a faster and more accurate differential grouping for large-scale black-box optimization. *IEEE Trans. Evol. Comput.*, 2017, vol. 21, no. 6, pp. 929–942. DOI: <https://doi.org/10.1109/TEVC.2017.2694221>
- [31] Li L., Fang W., Wang Q., et al. Differential grouping with spectral clustering for large-scale global optimization. *IEEE Trans. Evol. Comput.*, 2019, pp. 334–341. DOI: <https://doi.org/10.1109/CEC.2019.8790056>
- [32] Liu J., Tang K. Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution. *IDEAL 2013*. Berlin, Springer Verlag, 2013, pp. 350–357. DOI: https://doi.org/10.1007/978-3-642-41278-3_43
- [33] Mahdavi S., Rahnamayan S., Shiri M. Cooperative coevolution with sensitivity analysis-based budget assignment strategy for large-scale global optimization. *Appl. Intell.*, 2017, vol. 47, no. 3, pp. 888–913. DOI: <https://doi.org/10.1007/s10489-017-0926-z>

- [34] Ge H., Zhao M., Hou Y., et al. Bi-space interactive cooperative coevolutionary algorithm for large-scale black-box optimization. *Appl. Soft Comput.*, 2020, vol. 97-A, art. 106798. DOI: <https://doi.org/10.1016/j.asoc.2020.106798>
- [35] Liu W., Zhou Y., Li B., et al. Cooperative coevolution with soft grouping for large-scale global optimization. *Proc. IEEE CEC*, 2019, pp. 318–325. DOI: <https://doi.org/10.1109/CEC.2019.8790053>
- [36] Sun Y., Kirley M., Halgamuge S. A recursive decomposition method for large-scale continuous optimization. *IEEE Trans. Evol. Comput.*, 2018, vol. 22, no. 5, pp. 647–661. DOI: <https://doi.org/10.1109/TEVC.2017.2778089>
- [37] Sun Y., Li X., Ernst A., et al. Decomposition for large-scale optimization problems with overlapping components. *Proc. IEEE CEC*, 2019, pp. 326–333. DOI: <https://doi.org/10.1109/CEC.2019.8790204>
- [38] Vakhnin A., Sopov E. Investigation of the iCC framework performance for solving constrained LSGO problems. *Algorithms*, 2020, vol. 13, no. 5, art. 108. DOI: <https://doi.org/10.3390/a13050108>
- [39] Vakhnin A., Sopov E. Investigation of improved cooperative coevolution for large-scale global optimization problems. *Algorithms*, 2021, vol. 14, no. 5, art. 146. DOI: <https://doi.org/10.3390/a14050146>
- [40] Polakova R., Bujok P. Adaptation of population size in differential evolution algorithm: an experimental comparison. *Proc. IWSSIP*, 2018. DOI: <https://doi.org/10.1109/IWSSIP.2018.8439374>

Вахнин Алексей Валерьевич — аспирант кафедры системного анализа и исследования операций СибГУ им. М.Ф. Решетнева (Российская Федерация, 660037, г. Красноярск, пр-т имени газеты «Красноярский рабочий», д. 31).

Сопов Евгений Александрович — д-р техн. наук, профессор кафедры системного анализа и исследования операций СибГУ им. М.Ф. Решетнева (Российская Федерация, 660037, г. Красноярск, пр-т имени газеты «Красноярский рабочий», д. 31); профессор кафедры информационных систем СФУ (Российская Федерация, 660041, г. Красноярск, Свободный пр-т, д. 79).

Рурич Мария Александровна — студентка магистратуры кафедры системного анализа и исследования операции СибГУ им. М.Ф. Решетнева (Российская Федерация, 660037, г. Красноярск, пр-т имени газеты «Красноярский рабочий», д. 31).

Просьба ссылаться на эту статью следующим образом:

Вахнин А.В., Сопов Е.А., Рурич М.А. Гибридный эволюционный алгоритм для решения задач глобальной оптимизации сверхбольшой размерности. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2023, № 2 (143), с. 51–73.

DOI: <https://doi.org/10.18698/0236-3933-2023-2-51-73>

HYBRID EVOLUTIONARY ALGORITHM FOR SOLVING THE LARGE-SCALE GLOBAL OPTIMIZATION PROBLEMS

A.V. Vakhnin¹

E.A. Sopov^{1, 2}

M.A. Rurich¹

alexeyvah@gmail.com

evgenysopov@gmail.com

mariar8@yandex.ru

¹ Reshetnev Siberian State University of Science and Technology,
Krasnoyarsk, Russian Federation

² SibFU, Krasnoyarsk, Russian Federation

Abstract

When solving applied problems in various areas of human activity, the need appears to find the best set of parameters according to the given criterion. Usually such a problem is being formulated as a parametric optimization problem. The paper considers optimization problems represented by the black-box model. As such problems dimension grows, it becomes difficult to find a satisfactory solution for many traditional optimization approaches even with a significant increase in the number of objective function calculations. A new hybrid evolutionary method in coordinating the self-adjusting coevolution algorithms with the COSACC-LS1 local search is proposed to solve the problems of global material optimization of the extra-large dimension. COSACC-LS1 is based on the idea of the computing resources automatic allocation between a group of self-tuning differential evolution algorithms based on coevolution and local search algorithm. Effectiveness of the proposed algorithm was evaluated on 15 reference test problems from the LSGO CE 2013 set. Results of the COSACC-LS1-based algorithm were compared with a number of modern metaheuristic algorithms that were designed specifically for solving the very large-scale optimization problems and were the winners and prize-winners in the optimization competitions conducted within the framework of the IEEE CEC. With the help of numerical experiments, it is demonstrated that the proposed algorithm is better than most other popular algorithms according to the average accuracy criterion of the solution found

Keywords

Large-scale global optimization, self-tuning evolutionary algorithm, differential evolution, cooperative coevolution

Received 22.12.2022

Accepted 13.02.2023

© Author(s), 2023

The study was supported by the Ministry of Education and Science of Russian Federation (grant no. 075-15-2022-1121)

REFERENCES

- [1] Karpenko A.P. *Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy* [Modern search optimization algorithms. Algorithms inspired by nature]. Moscow, Bauman MSTU Publ., 2021.
- [2] Del Ser J., Osaba E., Molina D., et al. Bio-inspired computation: where we stand and what's next. *Swarm Evol. Comput.*, 2019, vol. 48, pp. 220–250. DOI: <https://doi.org/10.1016/j.swevo.2019.04.008>
- [3] Tang K. Summary of results on CEC'08 competition on large-scale global optimization. China, USTC, 2008.
- [4] Tang K., Yao X., Suganthan P.N., et al. Benchmark functions for the CEC'2008 special session and competition on large-scale global optimization. China, USTC, 2007.
- [5] Tang K., Li X., Suganthan P.N., et al. Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization China, USTC, 2008.
- [6] Li X., Tang K., Omidvar M.N., et al. Benchmark functions for the CEC'2013 special session and competition on large-scale global optimization. Australia, RMIT University, 2013.
- [7] Molina D., La Torre A. Toolkit for the automatic comparison of optimizers: comparing large-scale global optimizers made easy. *Proc. IEEE CEC*, 2018. DOI: <https://doi.org/10.1109/CEC.2018.8477924>
- [8] Mahdavi S., Shiri M.E., Rahnamayan S. Metaheuristics in large-scale global continues optimization: a survey. *Inf. Sc.*, 2015, vol. 295, pp. 407–428. DOI: <https://doi.org/10.1016/j.ins.2014.10.042>
- [9] Singh A., Dulal N. A survey on metaheuristics for solving large-scale optimization problems. *Int. J. Comput. Appl.*, 2017, vol. 170, no. 5, pp. 1–7. DOI: <https://doi.org/10.5120/ijca2017914839>
- [10] Omidvar M.N., Li X., Yao X. A review of population-based metaheuristics for large-scale black-box global optimization. Part I. *IEEE Trans. Evol. Comput.*, 2022, vol. 26, no. 5, pp. 802–822. DOI: <https://doi.org/10.1109/TEVC.2021.3130838>
- [11] Omidvar M.N., Li X., Yao X. A review of population-based metaheuristics for large-scale black-box global optimization. Part II. *IEEE Trans. Evol. Comput.*, 2022, vol. 26, no. 5, pp. 823–843. DOI: <https://doi.org/10.1109/TEVC.2021.3130835>
- [12] Molina D., La Torre A., Herrera F. SHADE with iterative local search for large-scale global optimization. *Proc. IEEE CEC*, 2018. DOI: <https://doi.org/10.1109/CEC.2018.8477755>
- [13] La Torre A., Muelas S., Pena J.-M. Multiple offspring sampling in large-scale global optimization. *Proc. IEEE CEC*, 2012. DOI: <https://doi.org/10.1109/CEC.2012.6256611>
- [14] Zhao S.Z., Liang J.J., Suganthan P.N., et al. Dynamic multi-swarm particle swarm optimizer with local search for large-scale global optimization. *Proc. IEEE CEC*, 2008, pp. 3845–3852. DOI: <https://doi.org/10.1109/CEC.2008.4631320>

- [15] Marcelino C., Almeida P., Pedreira C., et al. Applying C-DEEPSO to solve large-scale global optimization problems. *Proc. IEEE CEC*, 2018.
DOI: <https://doi.org/10.1109/CEC.2018.8477854>
- [16] Lopez E., Puris A., Bello R. VMODE: a hybrid metaheuristic for the solution of large-scale optimization problems. *Investig. Operacional*, 2015, vol. 36, no. 3, pp. 232–239.
- [17] Tseng L.-Y., Chen C. Multiple trajectory search for large-scale global optimization. *Proc. IEEE CEC*, 2018, pp. 3052–3059.
DOI: <https://doi.org/10.1109/CEC.2008.4631210>
- [18] Molina D., Herrera F. Iterative hybridization of DE with local search for the CEC'2015 special session on large-scale global optimization. *Proc. IEEE CEC*, 2015, pp. 1974–1978. DOI: <https://doi.org/10.1109/CEC.2015.7257127>
- [19] Qin A.K., Huang V.L., Suganthan P.N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.*, 2009, vol. 13, no. 2, pp. 398–417. DOI: <https://doi.org/10.1109/TEVC.2008.927706>
- [20] Morales J., Nocedal J. Remark on “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization”. *ACM Trans. Math. Softw.*, 2011, vol. 38, no. 1, pp. 1–4. DOI: <https://doi.org/10.1145/2049662.2049669>
- [21] La Torre A., Muelas S., Pena J.-M. Large-scale global optimization: experimental results with MOS-based hybrid algorithms. *Proc. IEEE CEC*, 2013, pp. 2742–2749.
DOI: <https://doi.org/10.1109/CEC.2013.6557901>
- [22] Tanabe R., Fukunaga A. Evaluating the performance of SHADE on CEC'2013 benchmark problems. *Proc. IEEE CEC*, 2013, pp. 1952–1959.
DOI: <https://doi.org/10.1109/CEC.2013.6557798>
- [23] Bolufe-Rohler A., Fiol-Gonzalez S., Chen S. A minimum population search hybrid for large-scale global optimization. *Proc. IEEE CEC*, 2015, pp. 1958–1965.
DOI: <https://doi.org/10.1109/CEC.2015.7257125>
- [24] Potter M., De Jong K. A cooperative coevolutionary approach to function optimization. *PPSN 1994*. Berlin, Springer Verlag, 1994, pp. 245–257.
DOI: https://doi.org/10.1007/3-540-58484-6_269
- [25] Yang Z., Tang K., Yao X. Large-scale evolutionary optimization using cooperative coevolution. *Inf. Sc.*, 2008, vol. 178, no. 15, pp. 2985–2999.
DOI: <https://doi.org/10.1016/j.ins.2008.02.017>
- [26] Yang Z., Tang K., Yao X. Self-adaptive differential evolution with neighborhood search. *Proc. IEEE CEC*, 2008, pp. 1110–1116.
DOI: <https://doi.org/10.1109/CEC.2008.4630935>
- [27] Hadi A., Wagdy A., Jambi K. LSHADE-SPA memetic framework for solving large-scale optimization problems. *Complex Intell. Syst.*, 2019, vol. 5, no. 1, pp. 25–40.
DOI: <https://doi.org/10.1007/s40747-018-0086-8>
- [28] Omidvar M.N., Li X., Yao X. Cooperative coevolution with delta grouping for large-scale non-separable function optimization. *Proc. IEEE CEC*, 2010, pp. 1–8.
DOI: <https://doi.org/10.1109/CEC.2010.5585979>

- [29] Omidvar M.N., Li X., Mei Y., et al. Cooperative coevolution with differential grouping for large-scale optimization. *IEEE Trans. Evol. Comput.*, 2014, vol. 18, no. 3, pp. 378–393. DOI: <https://doi.org/10.1109/TEVC.2013.2281543>
- [30] Omidvar M.N., Yang M., Mei Y., et al. DG2: a faster and more accurate differential grouping for large-scale black-box optimization. *IEEE Trans. Evol. Comput.*, 2017, vol. 21, no. 6, pp. 929–942. DOI: <https://doi.org/10.1109/TEVC.2017.2694221>
- [31] Li L., Fang W., Wang Q., et al. Differential grouping with spectral clustering for large-scale global optimization. *IEEE Trans. Evol. Comput.*, 2019, pp. 334–341. DOI: <https://doi.org/10.1109/CEC.2019.8790056>
- [32] Liu J., Tang K. Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution. *IDEAL 2013*. Berlin, Springer Verlag, 2013, pp. 350–357. DOI: https://doi.org/10.1007/978-3-642-41278-3_43
- [33] Mahdavi S., Rahnamayan S., Shiri M. Cooperative coevolution with sensitivity analysis-based budget assignment strategy for large-scale global optimization. *Appl. Intell.*, 2017, vol. 47, no. 3, pp. 888–913. DOI: <https://doi.org/10.1007/s10489-017-0926-z>
- [34] Ge H., Zhao M., Hou Y., et al. Bi-space interactive cooperative coevolutionary algorithm for large-scale black-box optimization. *Appl. Soft Comput.*, 2020, vol. 97-A, art. 106798. DOI: <https://doi.org/10.1016/j.asoc.2020.106798>
- [35] Liu W., Zhou Y., Li B., et al. Cooperative coevolution with soft grouping for large-scale global optimization. *Proc. IEEE CEC*, 2019, pp. 318–325. DOI: <https://doi.org/10.1109/CEC.2019.8790053>
- [36] Sun Y., Kirley M., Halgamuge S. A recursive decomposition method for large-scale continuous optimization. *IEEE Trans. Evol. Comput.*, 2018, vol. 22, no. 5, pp. 647–661. DOI: <https://doi.org/10.1109/TEVC.2017.2778089>
- [37] Sun Y., Li X., Ernst A., et al. Decomposition for large-scale optimization problems with overlapping components. *Proc. IEEE CEC*, 2019, pp. 326–333. DOI: <https://doi.org/10.1109/CEC.2019.8790204>
- [38] Vakhnin A., Sopov E. Investigation of the iCC framework performance for solving constrained LSGO problems. *Algorithms*, 2020, vol. 13, no. 5, art. 108. DOI: <https://doi.org/10.3390/a13050108>
- [39] Vakhnin A., Sopov E. Investigation of improved cooperative coevolution for large-scale global optimization problems. *Algorithms*, 2021, vol. 14, no. 5, art. 146. DOI: <https://doi.org/10.3390/a14050146>
- [40] Polakova R., Bujok P. Adaptation of population size in differential evolution algorithm: an experimental comparison. *Proc. IWSSIP*, 2018. DOI: <https://doi.org/10.1109/IWSSIP.2018.8439374>

Vakhnin A.V. — Post-Graduate Student, Department of System Analysis and Operation Research, Reshetnev Siberian State University of Science and Technology (Imeni gazety “Krasnoyarskiy rabochiy” prospekt 31, Krasnoyarsk, 660037 Russian Federation).

Sopov E.A. — Dr. Sc. (Eng.), Professor, Department of System Analysis and Operation Research, Reshetnev Siberian State University of Science and Technology (Imeni gazety “Krasnoyarskiy rabochiy” prospekt 31, Krasnoyarsk, 660037 Russian Federation); Professor, Department of Information Systems, SibFU (Svobodnyi prospekt 79, Krasnoyarsk, 660041 Russian Federation).

Rurich M.A. — Master’s Degree Student, Department of System Analysis and Operation Research, Reshetnev Siberian State University of Science and Technology (Imeni gazety “Krasnoyarskiy rabochiy” prospekt 31, Krasnoyarsk, 660037 Russian Federation).

Please cite this article in English as:

Vakhnin A.V., Sopov E.A., Rurich M.A. Hybrid evolutionary algorithm for solving the large-scale global optimization problems. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2023, no. 2 (143), pp. 51–73 (in Russ.). DOI: <https://doi.org/10.18698/0236-3933-2023-2-51-73>