Интернет-журнал «Науковедение» ISSN 2223-5167 http://naukovedenie.ru/

Том 8, №5 (2016) http://naukovedenie.ru/index.php?p=vol8-5

URL статьи: http://naukovedenie.ru/PDF/51TVN516.pdf

DOI: 10.15862/51TVN516 (http://dx.doi.org/10.15862/51TVN516)

Статья опубликована 31.10.2016.

Ссылка для цитирования этой статьи:
Сахаров М.К., Карпенко А.П., Иваньков И.Ф. Исследование эффективности алгоритма эволюции разума в задаче глобальной оптимизации // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 8, №5 (2016) http://naukovedenie.ru/PDF/51TVN516.pdf (доступ свободный). Загл. с экрана. Яз. рус., англ.

Работа поддержана РФФИ (проект № 16-07-00287)

УДК 519.6

Сахаров Максим Константинович
ФГБОУ ВПО «Московский государственный технический университет им. Н.Э. Баумана», Россия, Москва
Старший преподаватель
E-mail: Max.sfn90@gmail.com
ORCID: http://orcid.org/0000-0002-1527-7112

Карпенко Анатолий Павлович
ФГБОУ ВПО «Московский государственный технический университет им. Н.Э. Баумана», Россия, Москва
Заведующий кафедрой
Доктор физико-математических наук, профессор
E-mail: apkarpenko@mail.ru

Иваньков Игорь Федорович
ФГБОУ ВПО «Московский государственный технический университет им. Н.Э. Баумана», Россия, Москва
Ассистент
E-mail: ivankov.i.f@gmail.com

Исследование эффективности алгоритма эволюции разума в задаче глобальной оптимизации

Аннотация. В работе представлен простой алгоритм эволюции разума, являющийся каноническим алгоритмом класса алгоритмов эволюции разума. Авторами разработана программное обеспечение, реализующее данный алгоритм, в последовательном и параллельном вариантах. В рамках работы выполнен широкий вычислительный эксперимент по исследованию эффективности разработанного алгоритмического и программного обеспечения. Для исследования использованы многомерные овражная функция Розенброка и многоэкстремальные функции Экли и Растригина. Кроме того, рассмотрена практически значимая четырехмерная задача оптимизации сварной конструкции балки, отличительной особенностью которой является наличие значительного числа сложных ограничений на вектор варьируемых параметров. Исследование эффективности параллельной реализации алгоритма выполнено с использованием многомерных функций Розенброка и Растригина; представлена оценка ускорения вычислений за счет распараллеливания.

Ключевые слова: глобальная оптимизация; эволюционные алгоритмы; алгоритм эволюции разума; параллельные алгоритмы

Введение

Во многих областях человеческой деятельности возникают задачи глобальной оптимизации, принципиальной особенностью которых является наличие локально-оптимальных решений. Известно значительное число различных алгоритмы решения таких задач. На верхнем уровне иерархии вслед за работой [1], разделяем алгоритмы глобальной оптимизации на классические и не классические (популяционные). *Популяционные* алгоритмы в сравнении с классическими имеют неоспоримые преимущества при решении мультимодальных задач высокой размерности. Важно также, что популяционные алгоритмы позволяют эффективнее классических алгоритмов отыскивать субоптимальные (близкие к оптимальному) решения. В практически значимых задачах оптимизации часто достаточным является именно такое решение.

Выделяем следующие классы популяционных алгоритмов глобальной оптимизации: эволюционные алгоритмы, включая генетические; популяционные алгоритмы, вдохновленные живой природой; алгоритмы, вдохновленные неживой природой; алгоритмы, инспирированные человеческим обществом; прочие алгоритмы [2].

Алгоритмы эволюции разума (Mind Evolutionary Computation, MEC) [3] относятся к числу алгоритмов, инспирированных человеческим обществом. Алгоритмы этого класса моделирует некоторые аспекты поведения человека в обществе. Индивид в алгоритмах MEC рассматривается как разумный агент, функционирующий в группе аналогичных индивидов. При принятии решений данный индивид учитывает влияние со стороны членов своей группы, а также со стороны членов других групп. При этом моделируется следующая логика поведения индивида. Для того чтобы достичь высокого положения в группе, ему нужно учиться у наиболее успешных индивидов этой группы. С другой стороны, чтобы группа, которой принадлежит данный индивид, становилась более успешной по сравнению с другими группам, этот индивид, как и все индивиды его группы, должен руководствоваться тем же принципом в межгрупповой конкуренции.

Представленную концепцию алгоритмы MEC реализуют с помощью операций *локальных состязаний* и *диссимиляции*, ответственных за локальный и глобальный поиск соответственно. *Доски объявлений* используются алгоритмом для хранения информации об истории эволюции популяции. На основе этой информации осуществляется управление процессом оптимизации.

В работе рассматриваем алгоритм семейства MEC, называемый *простым алгоритмом* эволюции разума (Simple MEC, SMEC). Алгоритм выбран для исследования в силу следующих причин: в настоящее время этот алгоритм и его модификации успешно используются для решения широкого класса задач оптимизации; алгоритм имеет высокий потенциал развития для организации параллельных вычислений, особенно, в слабосвязанных системах [4]; алгоритм недостаточно изучен - известно относительно небольшое число его модификаций (тогда как для известного популяционного алгоритма роя частиц, например, известны десятки модификаций [1]).

Основными являются следующие модификации алгоритма SMEC.

Расширенный алгоритм эволюции разума EMEC (Extended Mind Evolutionary Computation) [5] развивает алгоритм SMEC путем использования модифицированных операций локальных состязаний и диссимиляции, а также новой операции *кооперации* (cooperation). Суть этой операции состоит в том, что в процессе эволюции каждая из групп на основе информации с *глобальной доски объявлений* находит соседнюю «лучшую» группу и изменяет свое положение в пространстве поиска, перемещаясь в направлении этой группы.

Улучшенный SMEC-алгоритм (Improved Mind Evolutionary Computation, IMEC) [6] имеет следующие основные особенности: при инициализации групп используется стратегия с разделением области поиска на подобласти; операция локальных состязаний реализована с помощью самоадаптирующегося алгоритма; операция диссимиляции расширена средствами нишевания.

Так называемая, *хаотическая* модификация SMEC-алгоритма (Chaotic Mind Evolutionary Computation, CMEC) [7] представляет собой гибридизацию алгоритма SMEC и хаотического алгоритма локального поиска, который вместо классических паевдослучайных чисел использует хаотические последовательности.

Отметим также мультимемеевую модификацию алгоритма SMEC, названную авторами HMEC [8].

Целью работы является исследование эффективности последовательного и параллельного вариантов алгоритма SMEC.

1. Постановка задачи

Рассматриваем детерминированную задачу глобальной безусловной максимизации

$$\max_{X \in R^{|X|}} f(X) = f(X^*) = f^*, \qquad (1)$$

где: $f(X)$ - скалярная целевая функция; $X = (x_1, x_2, ..., x_{|X|})$ - $|X|$-мерный вектор варьируемых параметров; $R^{|X|}$ - $|X|$-мерное арифметическое пространство; X^* - искомое решение задачи.

Решение задачи отыскиваем в области

$$D = \left\{X \mid x_i^{\min} \le x_i \le x_i^{\max},\ i \in [1:|X|]\right\} \subset R^{|X|}, \qquad (2)$$

представляющей собой параллелепипед в пространстве $R^{|X|}$.

2. Алгоритм SMEC

Популяция алгоритм SMEC состоит из лидирующих групп $\mathbf{S}^b = (S_1^b, S_2^b, ..., S_{|\mathbf{S}^b|}^b)$ и отстающих групп $\mathbf{S}^w = (S_1^w, S_2^w, ..., S_{|\mathbf{S}^w|}^w)$, число индивидов в которых полагаем одинаковым и равным $|S|$. Локальные доски объявлений групп S_i^b, S_j^w обозначаем C_i^b, C_j^w соответственно. Глобальную доску объявлений обозначаем C^g. Отсутствие индексов b, w в обозначениях групп и соответствующих локальных досок объявлений означает произвольную из указанных групп и их досок объявлений.

Алгоритм SMEC построен на основе операций инициализации групп, локальных состязаний (*similar-taxis*) и диссимиляции (*dissimilation*).

Операция инициализации групп создает группы $\mathbf{S}^b$, $\mathbf{S}^w$ и размещает их в области поиска. Рассмотрим схему операции на примере группы S_i.

1) Генерируем случайный вектор $X_{i,1}$, компоненты которого равномерно распределены в области поиска *D*. Отождествляем этот вектор с индивидом $s_{i,1}$ группы S_i.

2) Определяем начальные координаты остальных индивидов данной группы по формуле

$$X_{i,j} = X_{i,1} + N_{|X|}(0,\sigma),\ j \in [2:|S|], \tag{3}$$

где $N_{|X|}(0,\sigma)$ - $(|X| \times 1)$-вектор независимых вещественных случайных чисел, распределенных по нормальному закону с математическим ожиданием и средним квадратичным отклонением, равными *0* и σ соответственно.

Операция локальных состязаний реализует локальный поиск максимума фитнесс-функции каждой из групп $\mathbf{S}^b$, $\mathbf{S}^w$. Схема этой операции для группы S_i имеет следующий вид.

1) С доски объявлений C_i берем информацию о текущем индивиде-победителе группы S_i. Пусть это будет индивид s_{i,j_b}, $j_b \in [1:|S|]$.

2) Определяем новые координаты остальных индивидов $s'_{i,j}$, $j \in [1:|S|]$, $j \neq j_b$ данной группы по правилу вида (3), то есть размещаем случайным образом вокруг победителя в соответствии с нормальным законом распределения.

3) Вычисляем *счета* (*scores*) агентов группы $\varphi'_{i,j} = \varphi(X'_{i,j})$; $j \in [1:|S|]$.

4) Определяем нового победителя группы s'_{i,k_b}, $k_b \in [1:|S|]$, как индивида данной группы, который имеет максимальный текущий счет.

5) Заносим информацию о новом победителе группы s'_{i,k_b} на доски объявлений C_i, C^g.

Операция диссимиляции управляет глобальным поиском. Схема операции имеет следующий вид.

1) С глобальной доски объявлений C^g считываем текущие счета победителей групп $\mathbf{S}^b$, $\mathbf{S}^w$.

2) Выполняем сравнение указанных счетов между собой. Если счет некоторой лидирующей группы S_i^b меньше счета одной из отстающих групп S_j^w, то последняя группа занимает место группы S_i^b в наборе лидирующих групп $\mathbf{S}^b$, а группа S_i^b - место группы S_j^w среди отстающих групп $\mathbf{S}^w$. Если счет группы S_j^w ниже счетов всех лидирующих групп, то удаляем группу S_j^w из популяции.

3) С помощью операции инициализации взамен каждой из удаленных групп инициализируем новую группу.

Операции локальных состязаний и диссимиляции итерационно повторяем до тех пор, пока имеет место увеличение максимального счета лидирующих групп. При прекращении роста этого показателя, решение задачи, соответствующее победителю лучшей из лидирующих групп, объявляем искомым приближением к решению задачи (1).

3. Вычислительный эксперимент

Разработанное программное обеспечение, реализующее алгоритм SMEC, функционирует под управлением операционной системы mac OS X. При разработке использована интегрированная среда разработки CLion и язык программирования высокого уровня C++.

Характер сходимости алгоритма для сферической целевой функции иллюстрирует рисунок 1. Генерация начальной популяции выполнена в параллелепипеде (2) при $x_i^{\min} = -10$, $x_i^{\max} = 10$; $i \in [1:|X|]$. Здесь и далее полагаем, что фитнесс-функция обратна целевой функции, то есть $\varphi(X) = -f(X)$. Вычислительный эксперимент во всех случаях выполнен при следующих значениях свободных параметров алгоритма: $|S^b| = |S^w| = |S| = 50$; σ=0,001. В качестве условия окончания итераций использовано условие $|\tilde{\varphi}^*_{k+1} - \tilde{\varphi}^*_k)| < \varepsilon = 10^{-3}$, где $\tilde{\varphi}^*_k$ - лучшее значение фитнесс-функции, достигнутое на *k*-ой итерации.

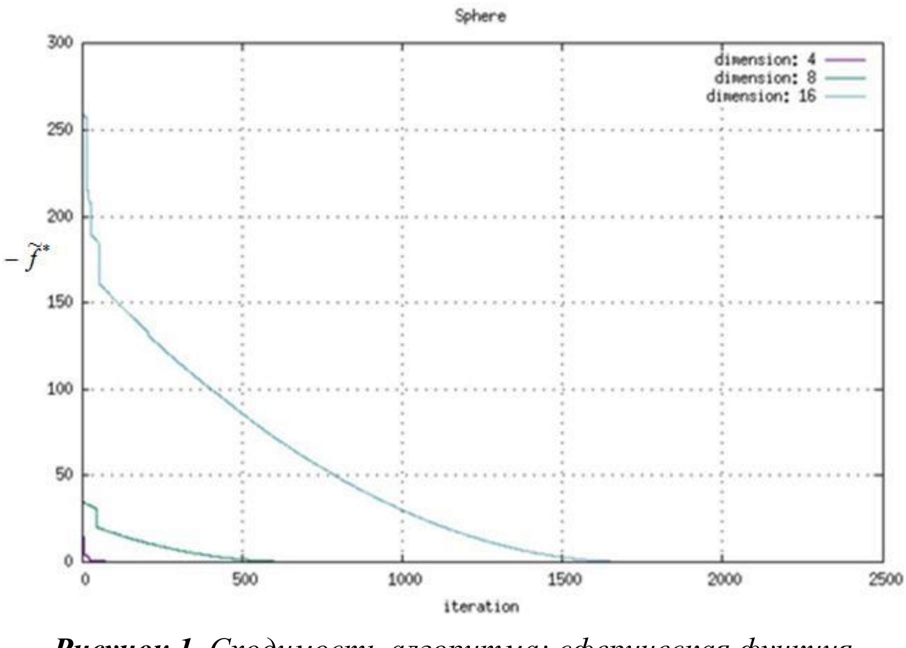

Рисунок 1. *Сходимость алгоритма: сферическая функция*

Овражная функция. Исследование эффективности алгоритма для овражных функций выполнено на примере функции Розенброка [1] в области $D = \{X \mid -2 \le x_i \le 2, i \in [1:|X|]\}$. Поскольку алгоритм SMEC является стохастическим, использован метод мультистарта с числом запусков, равным 50. Некоторые результаты исследования представлены в таблице 1 и на иллюстрирующем ее рисунке 2. В таблице 1 и в следующих аналогичных таблицах приняты обозначения: $\tilde{f}^{*}$ - лучшее достигнутое значение фитнесс функции; $\bar{k}_{cx}$ - средняя скорость сходимости; $\tilde{P}$ - оценка вероятности локализации глобального экстремума с заданной точностью $\delta = 10^{-3}$.

Таблица 1

Результаты исследования: функция Розенброка

$\|X\|$	$\tilde{f}^{*}$	$\bar{k}_{cx}$	$\tilde{P}$
4	2,2E-06	404	1,00
8	3,2E-04	603	0,95
16	5,3E-03	871	0,90

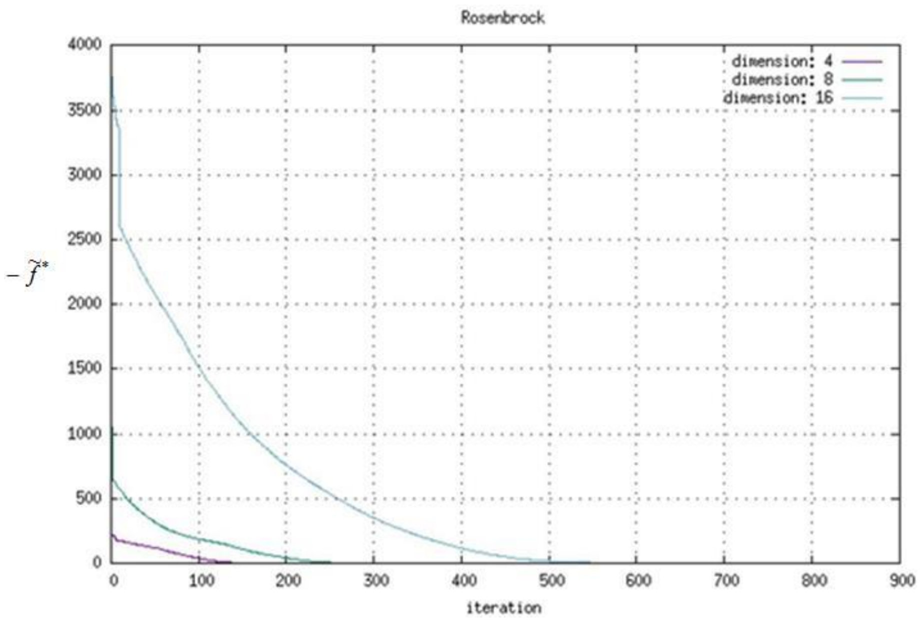

Рисунок 2. *Сходимость алгоритма: функция Розенброка*

Данные, представленные в таблице 1 и на рисунке 2, показывают, что для овражной функции Розенброка алгоритм SMEC обеспечивает достаточно высокую скорость сходимости и может быть использован для решения подобных задач без своей модификации.

Многоэкстремальные функции. Рассматриваем известные тестовые многоэкстремальные функции Экли в области $D = \{X \mid -5 \le x_i \le 5,\ i \in [1:|X|]\}$ и Растригина в области $D = \{X \mid -5{,}12 \le x_i \le 5{,}12,\ i \in [1:|X|]\}$ [1]. Результаты вычислительного эксперимента представлены в таблицах 2, 3 и на рисунках 3, 4.

Таблица 2

Результаты исследования: функция Экли

$\|X\|$	$\tilde{f}^{*}$	$\bar{k}_{cx}$	$\tilde{P}$
4	3,2E-05	446	0,95
8	8,2E-05	3187	0,85
16	3,0E-04	2575	0,70

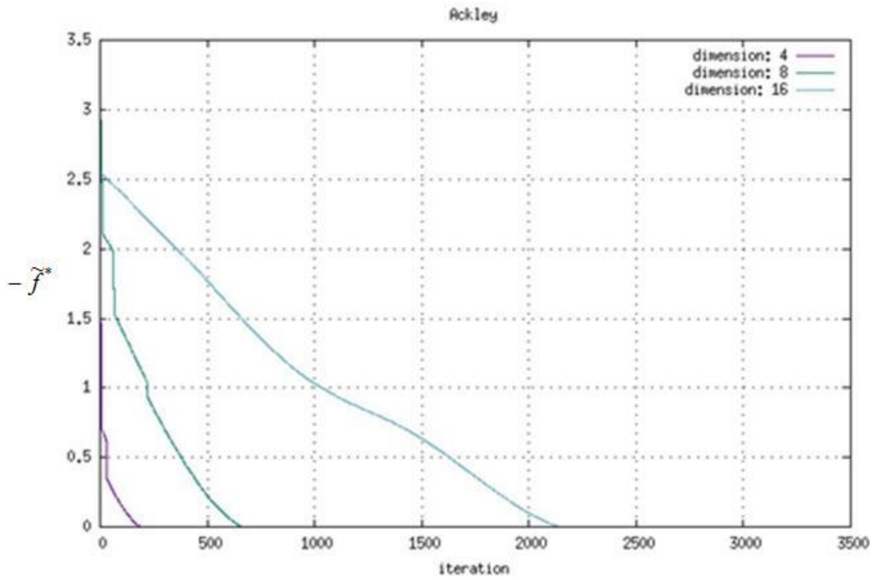

Рисунок 3. *Сходимость алгоритма: функция Экли*

Таблица 3

Результаты исследования: функция Растригина

$\left\|X\right\|$	$\tilde{f}^*$	$\bar{k}_{cx}$	$\tilde{P}$
4	4,06E-08	847	0,3
8	2,69E-06	662	0,15
16	1,56E-05	1542	0,05

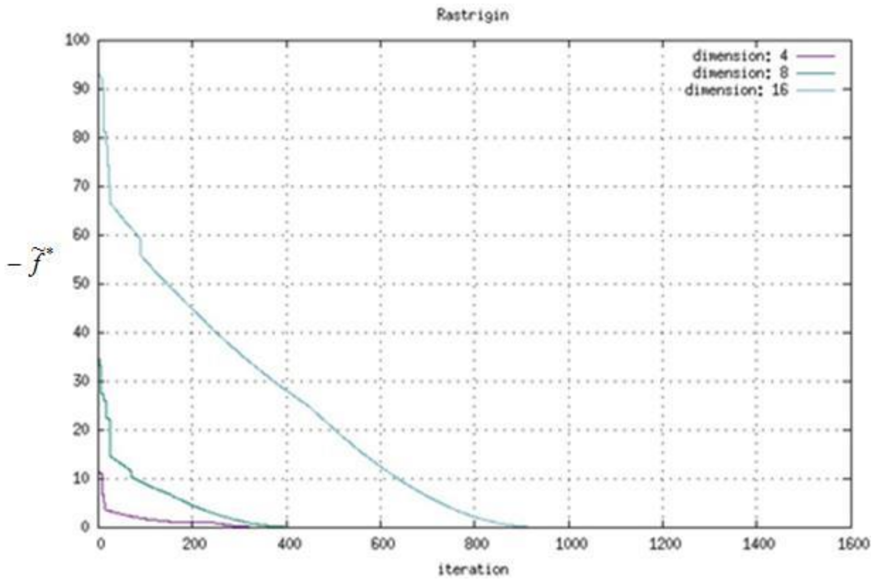

Рисунок 4. *Сходимость алгоритма: функция Растригина*

Таблицы 2, 3 и рисунки 3, 4 показывают, что в комбинации с методом мультистарта алгоритм SMEC может обеспечить отыскание глобального экстремума даже таких сложных мультимодальных функций, как функции Экли и Растригина. Подчеркнем, что при высокой размерности вектора варьируемых параметров функция Растригина в исследуемой области имеет очень большое число экстремумов, так что невысокая вероятность локализации глобального экстремума этой функции вполне ожидаема (таблица 3).

Оптимизация сварной конструкции балки. Рассматриваем практически значимую задачу оптимизации сварной конструкции балки (рисунок 5) [9]. Целью оптимизации является минимизация стоимости материала, расходуемого на изготовление этой конструкции.

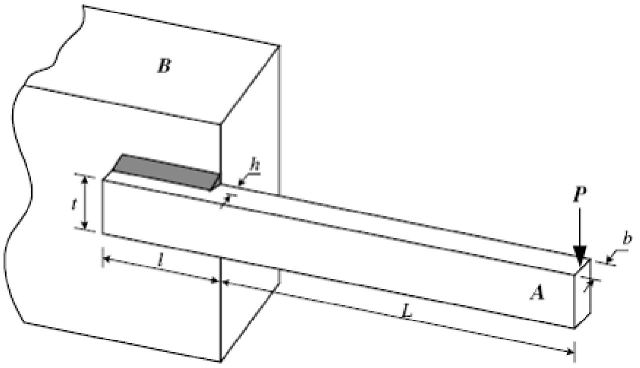

Рисунок 5. *Схема сварной конструкции балки [9]*

Длина балки фиксирована и равна $L = 14$ дюймов, величина сосредоточенной нагрузки на конец балки равна $P = 6000$ фунтов. Целевая и ограничивающие функции имеют следующий вид:

$$f(X) = 1{,}10471\ h^2 l + 0{,}04811\ t b (L + l);$$

$$g_1(X) = \tau_d - \tau(X) \ge 0;$$

$$g_2(X) = \sigma_d - \sigma(X) \ge 0;$$

$$g_3(X) = b - h \ge 0;$$

$$g_4(X) = P_c(X) - P \ge 0;$$

$$g_5(X) = 0{,}25 - \delta(X) \ge 0.$$

Здесь принято, что $\tau_d = 13600$ фунтов на кв. дюйм - допустимое напряжение сварного шва;

$$\tau(X) = \sqrt{(\tau'(X))^2 + (\tau''(X))^2 + \frac{l\tau'(X)\tau''(X)}{\sqrt{0{,}25\ (l^2 + (h+t)^2)}}}$$

- касательное напряжение в сварном шве; $\sigma_d = 30600$ фунтов на кв. дюйм - допустимый предел текучести материала конструкции; $\sigma(X) = \frac{504000}{t^2 b}$ - максимальное напряжение изгиба;

$$P_c(X) = 64746\ (1 - 0{,}0282346\ t)\ t b^2$$

- критическая нагрузка; $\delta(X) = \frac{2{,}1952}{t^2\ b}$ - прогиб конца балки. Величины $\tau'(X)$, $\tau''(X)$ определяют выражения

$$\tau'(X) = \frac{6000}{l\ h\ \sqrt{2}}, \quad \tau''(X) = \frac{6000\ (14 + 0{,}5\ l)\sqrt{0{,}25\ (l^2 + (h+t)^2)}}{2\left(0{,}707\ h\ l\left(\frac{l^2}{12} + 0{,}25\ (h+t)^2\right)\right)}.$$

Компонентами вектор варьируемых параметров X являются параметры конструкции b, h, l, t , подчиненные ограничениям

$$0{,}12 \le h \le 5{,}0 ; \; 0{,}1 \le l ; \; t \le 10{,}0 ; \; 0{,}1 \le b \le 5{,}0 . \qquad (4)$$

Для перехода от поставленной задаче условной оптимизации к задаче безусловной оптимизации используем метод штрафных функций, в качестве которых используем функции вида

$$p(X) = \begin{cases} 0, & g(X) \ge 0, \\ g^2(X), & g(X) < 0. \end{cases}$$

Результаты решения поставленной задачи представлены в таблице 4, в которой приведены также результаты решения этой задачи, полученные другими авторами [9].

Таблица 4

Результаты оптимизации сварной конструкции

Параметр	Алгоритм оптимизации				
	APPROX	GP	SIMPLEX	RANDOM	SMEC
b	0,2444	0,2455	0,2792	0,4575	0,2489
h	6,2189	6,1960	5,6256	4,7313	6,1730
l	8,2915	8,2730	7,7512	5,0853	8,1789
t	0,2444	0,2455	0,2796	0,66	0,2533
$\tilde{f}^*$	2,38	2,39	2,53	4,12	2,43

Таблица 4 показывает, что даже простейший алгоритм эволюции разума SMEC является вполне конкурентно-способным по сравнению с представленными алгоритмами: алгоритм позволил получить третий результат с точки зрения достигнутого значения целевой функции. Отметим высокую сложность фитнесс-функции в данной задаче, которая обусловлена использованием метода штрафных функций, приводящего, как известно, к овражности даже в простейших задачах оптимизации.

4. Распараллеливание алгоритма SMEC

Разработанная параллельная программа реализует модель параллельных вычислений типа «хозяин-раб» (*master-slave*). Мастер-процесс выполняет собственно алгоритм SMEC. Каждый из подчиненных (рабочих) процессов назначаем на выполнение одному из потоков. Функциональность рабочих-процессов ограничена вычислением значений фитнес-функции. Для синхронизации этих вычислений используем глобальную барьерную синхронизацию, так что очередную итерацию вычислительного процесса мастер-процесс начинает только после получения информации обо всех агентах популяции.

Вычислительный эксперимент с разработанным параллельным программным обеспечением выполнен с использованием процессора Intel Core i5-5257U, который представляет собой быстрый двухъядерный процессор со встроенным графическим чипом Iris Graphics 6100. Поддержка технологии Hyper-Threading позволяет процессору работать с четырьмя вычислительными потоками посредством всего двух физических ядер.

Эффективность разработанного параллельного программного обеспечения иллюстрируют таблицы 5, 6, где T_{seq}, T_{par} - времена последовательного и параллельного решения задачи соответственно; $S = T_{seq} / T_{par}$ - ускорение вычислений.

Таблица 5

Ускорение параллельной программы: функция Розенброка

$\lvert X \rvert$	T_{seq}, с	T_{par}, с	S
4	0,37	0,21	1,76
8	1,72	0,87	1,97
16	5,54	2,79	1,98

Таблица 6

Ускорение параллельной программы: функция Растригина

$\lvert X \rvert$	T_{seq}, с	T_{par}, с	S
4	2,11	1,16	1,81
8	4,66	2,79	1,67
16	7,27	3,65	1,99

Таблицы показывают, что для обеих рассмотренных тестовых задач ускорение вычислений достигает величины, равной примерно двум (при асимптотическом ускорении, равном приблизительно четырем). В силу невысокой вычислительной сложности указанных функций этот результат следует считать хорошим. В практически значимых задачах вычислительная сложность целевых функций может быть на несколько порядков выше, что должно обеспечить значительно более высокое ускорение.

Заключение

В работе представлен простой алгоритм эволюции разума (Simple MEC, SMEC), являющийся каноническим алгоритмом класса алгоритмов эволюции разума MEC.

Разработано программное обеспечение, реализующее алгоритм SMEC, в последовательном и параллельном вариантах. Последовательная версия программного обеспечения функционирует под управлением операционной системы mac OS X, при разработке использована интегрированная среда разработки CLion и язык программирования высокого уровня C++. Параллельная версия программы реализует модель параллельных вычислений типа «хозяин-раб» (*master-slave*) и ориентирована на использование многоядерных процессоров, таких как использованный в работе двух ядерный процессор Intel Core i5-5257U, поддерживающий технологию Hyper-Threading, которая позволяет процессору работать с четырьмя вычислительными потоками.

Выполнен широкий вычислительный эксперимент по исследованию эффективности разработанного алгоритмического и программного обеспечения. Для исследования использованы овражная функция Розенброка и многоэкстремальные функции Экли и Растригина, размерности которых менялись от четырех до 16. Кроме того, рассмотрена практически значимая четырехмерная задача оптимизации сварной конструкции балки, отличительной особенностью которой является наличие значительного числа сложных ограничений на вектор варьируемых параметров.

Результаты исследования последовательной реализации алгоритма SMEC показывают, что для овражных функций алгоритм обеспечивает достаточно высокую скорость сходимости. Для многомерных мультимодальных функций алгоритм SMEC лишь в комбинации с методом мультистарта может обеспечить отыскание глобального экстремума с достаточно высокой вероятностью. При решении указанной практически значимой задачи алгоритм показал себя как вполне конкурентно-способный по сравнению с алгоритмами, с помощью которых эта задача решалась ранее. Подчеркнем, что данная задача имеет сложный ландшафт целевой функции, обусловленный ее многомодальностью и сильной овражностью (вследствие использования метода штрафных функций).

Исследование эффективности параллельной реализации алгоритма выполнено с использованием многомерных функций Розенброка и Растригина. Результаты исследования показывают, что в обоих случаях ускорение вычислений достигает величины, равной примерно двум (на двух ядерном процессоре, напомним). Поскольку в практически значимых задачах вычислительная сложность целевых функций может быть на несколько порядков выше, в этих условиях следует ожидать значительно более высокого ускорения.

В развитии работы авторы планирует реализацию и исследование эффективности алгоритмов, построенных на основе алгоритмов класса MEC, при параллельном решении задач глобальной оптимизации на слабосвязанных параллельных вычислительных системах, а также при решении задач многокритериальной оптимизации методом, который предполагает предварительное построение дискретной аппроксимации множества (фронта) Парето этой задачи.

ЛИТЕРАТУРА

1. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы вдохновленные природой. – Москва: Издательство МГТУ им. Н.Э. Баумана. - 2014. – 446 с.
2. Gill P.E., Murray W., Wright M.H. Practical Optimization. - Emerald Publishing. - 2002. - P. 314-319.
3. Chengyi S., Yan S., Wanzhen W. A Survey of MEC: 1998-2001. IEEE International Conference on Systems 2002. Man and Cybernetics. - IEEE Publ., 2002. - Vol. 6. - P. 445-453.
4. Велисевич Я.И., Карпенко А.П., Сахаров М.К. Мультимемеевый алгоритм эволюции разума для слабосвязанных систем на основе персональных компьютеров // Наука и образование: электронное научно- техническое издание. – 2015. - №10 (http://technomag.bmstu.ru/doc/814435.html).
5. Jing Jie, Chongzhao Han, Jianchao Zeng. An Extended Mind Evolutionary Computation Model for Optimizations // Applied Mathematics and Computation. -2007. - 185(2). - P. 1038 – 1049.
6. Jing Jie, Jianchao Zeng. Improved Mind Evolutionary Computation for Optimizations / Proceedings of 5th World Congress on Intelligent Control and Automation, Hang Zhou, China, 2004. - P. 72-93.
7. Jianxia Liu, Nan Li, Keming Xie. Application of Chaos Mind Evolutionary Algorithm in Antenna Arrays Synthesis // Journal of computers. - May 2010. - Vol. 5. - No. 5. - P. 717-724.
8. Карпенко А.П., Сахаров М.К. Мультимемевая глобальная оптимизация на основе алгоритма эволюции разума // Информационные технологии.- 2014. - №7. - С. 22-30.
9. Lee K.S., Geem Z.W. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice // Computation Methods Applied Mechanics and Engineering. - 2005. – Vol. 194. - P. 3902–3933.

Sakharov Maxim Konstantinovich
Bauman Moscow state technical university, Russia, Moscow
E-mail: Max.sfn90@gmail.com

Karpenko Anatoly Pavlovich
Bauman Moscow state technical university, Russia, Moscow
E-mail: apkarpenko@mail.ru

Ivankov Igor Fedorovich
Bauman Moscow state technical university, Russia, Moscow
E-mail: ivankov.i.f@gmail.com

Performance investigation of mind evolutionary computation algorithm in the context of global optimization

Abstract. This paper presents a simple Mind Evolutionary Computation Algorithm which is a canonical algorithm within a class of Mind Evolutionary Algorithms. Authors developed software implementation of that algorithm both in sequential and parallel versions. A large number of computational experiments was carried out in order to investigate the efficiency of both algorithm and its software implementation. In order to estimate the efficeince of a sequential version, a set of multidimensional benchmark functions was used, namely ravine Rosenbrock function and multiextremal Rastrigin and Ackley functions. In addition, a real-world problem was considered which deals with an optimization of welded beam's structure; the distinct feature of that task is a significant number of complex restrictions imposed on the vector of variables. Performance investigation of a parallel version was carried out with a use of multidimensional Rosenbrock and Rastrigin functions; computational speedup was also estimated.

Keywords: global optimization; evolutionary computation; mind evolutionary computation; parallel algorithms