Моделирование и анализ данных
2020. Том 10. № 2. С. 62–73
DOI: https://doi.org/10.17759/mda.2020100205
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2020. Vol 10. no. 2. pp. 62–73
DOI: https://doi.org/10.17759/mda.2020100205
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

◇◇◇◇◇◇◇◇ МЕТОДЫ ОПТИМИЗАЦИИ ◇◇◇◇◇◇◇◇

УДК 517.8

Разработка модифицированного самоорганизующегося миграционного алгоритма оптимизации (MSOMA)

Пантелеев А.В.*
Московский авиационный институт
(национальный исследовательский университет),
г. Москва, Российская Федерация
ORCID: https://orcid.org/0000–0003–2493–3617
e-mail: avpanteleev@inbox.ru

Ракитянский В.М.**
Московский авиационный институт (национальный исследовательский университет), г. Москва, Российская Федерация
ORCID: https://orcid.org/0000–0001–7894–7462
e-mail: rymbelv@gmail.com

В статье рассматривается модифицированный самоорганизующийся миграционный алгоритм (MSOMA), созданный на основе самоорганизующегося миграционного алгоритма (SOMA). Сформирован алгоритм решения задачи нахождения глобального условного экстремума функции многих переменных на заданном параллелепипедном множестве допустимых решений. Приведены примеры, иллюстрирующие применение созданного алгоритма и соответствующего программного обеспечения.

Ключевые слова: алгоритм глобальной оптимизации, миграционный цикл, популяция, индивид, тестовые задачи

Для цитаты:
Пантелеев А.В., Ракитянский В.М. Разработка модифицированного самоорганизующегося миграционного алгоритма оптимизации (MSOMA) // Моделирование и анализ данных. 2020. Том 10. № 2. С. 62–73. DOI:10.17759/ mda.2020100205

****Пантелеев Андрей Владимирович,*** доктор физико-математических наук, профессор, заведующий кафедрой математической кибернетики института «Информационные технологии и прикладная математика» Московского авиационного института (национального исследовательского университета), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000–0003–2493–361, e-mail: avpanteleev@inbox.ru

*****Ракитянский Владислав Максимович,*** студент бакалавриата института «Информационные технологии и прикладная математика» Московского авиационного института (национального исследовательского университета), г. Москва, Российская Федерация, ORCID: https://orcid.org/0000–0001–7894–7462, e-mail: rymbelv@gmail.com

1. ВВЕДЕНИЕ

При разработке модифицированного самоорганизующегося миграционного алгоритма оптимизации (MSOMA) использовалась базовая версия алгоритма SOMA [1]. SOMA – самоорганизующийся миграционный алгоритм, который может быть классифицирован, как эволюционный алгоритм оптимизации, основанный на самоорганизующемся поведении групп индивидов в социальном окружении.

Алгоритм реализуется следующим образом:

1) индивиды (потенциальные решения задачи) случайно генерируются согласно заданным ограничениям, задающим множество допустимых решений;
2) реализуется миграционный цикл – изменение позиций индивидов в пространстве поиска относительно индивида-лидера с наилучшим значением целевой функции.

Циклы продолжаются, пока не будут достигнуты критерии остановки.

Начиная с 1999 года, когда SOMA был представлен [1–2], были проведены различные исследования и найдены следующие области применения алгоритма [3]: создание и контроль химических реакций, контроль реакций плазмы, проектирование крыльев самолета, оптимизация хаотических систем и др.

Модификации заключаются в выделении среди индивидов, образующих популяцию, трех лидеров. Для каждого из членов популяции генерируются два клона с той же позицией. Тем самым порождаются три популяции, каждая из которых реализует миграционный цикл относительно своего лидера. Для всех членов популяции находятся наилучшие положения, достигнутые в течение цикла. В процессе поиска популяция регулярно обновляется за счет новых индивидов, генерируемых на множестве допустимых решений. Они замещают выбывающих индивидов с наихудшими значениями целевой функции. После выполнения условий окончания производится уточняющий поиск (миграционный цикл), в котором участвуют три оставшихся лидера популяции. В качестве решения предъявляется наилучший результат.

Модификации соответствуют основным идеям, применяемым в известных метаэвристических алгоритмах глобальной оптимизации и их приложениях при решении задач оптимального управления [4, 5].

2. ПОСТАНОВКА ЗАДАЧИ

Дана целевая функция $f(x) = f(x_1, x_2, \ldots, x_n)$, определенная на множестве допустимых решений $D \subseteq R^n$.

Требуется найти глобальный условный минимум функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что

$$f(x^*) = \min_{x \in D} f(x),$$

где $x = (x_1, x_2, \ldots, x_n)^T$, $D = \{x | x_i \in [a_i, b_i], i = 1, 2, \ldots, n\}$.

Задача поиска глобального условного максимума функции $f(x)$ на множестве D сводится к задаче поиска минимума путем замены знака перед функцией на противоположный.

3. СТРАТЕГИЯ ПОИСКА РЕШЕНИЯ

При решении задачи используются конечные наборы $I = \left\{ x^j = \left(x_1^j, x_2^j, \ldots, x_n^j \right)^T, j = 1, 2, \ldots, Np \right\} \subset D$ возможных решений, называемые популяциями, где x^j – индивид с номером j, Np – размер популяции.

Самоорганизующийся миграционный алгоритм имитирует эволюцию начальной популяции $I_0 = \{ x^j,\ j = 1, 2, \ldots, Np | x^j = \left(x_1^j, x_2^j, \ldots, x_n^j \right)^T \in D \}$ и представляет собой итерационный процесс, исследующий множество D.

Начальная популяция создается из индивидов со случайно сгенерированными координатами x_i из промежутка $\left[a_i, b_i \right], i = 1, \ldots, n$.

Далее реализуются миграционные циклы. Перед каждым циклом индивиды упорядочиваются и выявляются три лидера – индивиды с наилучшим значением целевой функции $f(x)$. У каждого индивида создаются по два индивида-клона с такими же координатами. Таким образом, создается 3 популяции суммарного размера $K = 3 * Np$. Каждая из популяций движется относительно одного из трех лидеров. В этих циклах все индивиды последовательно с некоторой величиной шага двигаются к своему лидеру и за него на некоторое расстояние. При этом лидер остается неподвижным. Затем выбирается та позиция индивида, которой соответствует наилучшее значение целевой функции, полученное в процессе поиска. Эта позиция записывается в следующую популяцию размера K. После записи всех индивидов в новую популяцию из нее удаляются $2 * Np + \left[\frac{1}{3} * Np \right]$ индивидов с худшим значением целевой функции $f(x)$. Затем генерируются $\left[\frac{1}{3} * Np \right]$ новых индивидов по правилам формирования начальной популяции для дополнения популяции до изначального размера Np с целью обновления популяции и исследования новых областей в множестве допустимых решений.

Алгоритм продолжается до тех пор, пока среднее отклонение по величине функции между тремя лидерами популяции не станет меньше заранее заданного значения или пока не реализуется максимальное число миграционных циклов. Тогда проводится уточняющий миграционный цикл для трех лидеров популяции относительно абсолютного лидера и возвращается лучшее значение, найденное в процессе поиска.

4. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

Шаг 1. *Задать параметры метода:*

1) *NStep* – контролирующий параметр, определяющий количество шагов до окончания движения.
2) *PRT* – контролирующий параметр, определяющий, будет ли индивид двигаться по выбранной координате к лидеру. Рекомендуемое значение 0,1.
3) *Np* – контролирующий параметр, определяющий размер популяции особей. Рекомендуемое значение *Np* > 10.
4) *Migration* – останавливающий параметр, показывающий максимальное количество итераций. Рекомендуемое значение *Migration* > 10.

5) *MinDist* – останавливающий параметр, отражающий величину среднего отклонения между тремя лидерами популяции по величине функции. Если эта величина меньше заданной, алгоритм остановится. Возможно задание любого значения (при отрицательном значении условие не выполнится, и поиск остановится по достижении максимального числа миграционных циклов).
6) *MCount* – счетчик итераций, необходимый для остановки алгоритма при достижении им числа *Migration*. Положить его равным нулю (*MCount* = 0).

Шаг 2. *Создание начальной популяции.*

Создание популяции индивидов со случайно сгенерированными координатами x_i из промежутка $\left[a_i, b_i\right]$:

$$x_i^j = a_i + rand_i\left[0,1\right] * \left(b_i - a_i\right),\ j = 1,\ldots,Np;\ i = 1,\ldots,n.$$

Шаг 3. *Миграционный цикл.*

Шаг 3.1. Каждый индивид оценивается с помощью целевой функции и выбираются лидеры (особи с лучшими значениями). Для этого популяция сортируется в порядке неубывания целевой функции: $\left\{x^1,\ldots,x^{Np}\right\}$. Затем из нее выбираются первые три особи с наименьшим значением целевой функции.

Шаг 3.2. Для всех индивидов создаются два «клона» – индивида с такими же координатами:

$$\text{для всех}\ \ j = 1,\ldots,Np : x_i^{Np+j} = x_i^j,\ x_i^{2*Np+j} = x_i^j, i = 1,\ldots,n.$$

Шаг 3.3. Перед началом движения каждого индивида к одному из лидеров генерируется случайное число для каждой координаты индивида, а затем сравнивается с *PRT*:

$$\text{если}\ \ rnd_i < PRT, \text{то}\ \ PRTVector_i = 1\ \text{иначе}\ \ , i = 1,\ldots,n.$$

Шаг 3.4. Затем все остальные особи начинают движение к соответствующему лидеру. Движение происходит по шагам, пока не будет достигнута конечная позиция на итерации под номером $NStep$, по следующей формуле:

– для первого лидера:

$$x^{j,m} = x^j + \frac{\left(x^1 - x^j\right)}{2 * NStep} m_1 PRTVector,\ m_1 = 0,1,\ldots,4 * NStep,\ j = 1,\ldots,Np\ ;$$

– для второго лидера:

$$x^{j,m} = x^j + \frac{\left(x^2 - x^j\right)}{Nstep} m_2 PRTVector,\ m_2 = 0,1,\ldots,2 * NStep,\ \ j = \left(Np + 1\right),\ldots,\left(2 * Np\right);$$

– для третьего лидера:

$$x^{j,m} = x^j + \frac{\left(x^3 - x^j\right)}{\left\lceil \frac{NStep}{2} \right\rceil} m_3 PRTVector,\ m_3 = 0,1,\ldots,NStep,\ j = \left(2 * Np + 1\right),\ldots,K\ ,$$

где $K = 3 * Np$.

Шаг 3.5. После всех выполненных шагов для каждого индивида находится наилучшая позиция (на которой значение целевой функции было наименьшим), а индивид становится на эту позицию, присваивая себе соответствующие значения координат, после чего переходит в следующую популяцию:

– для первого лидера:

$$x^{j,New} = \operatorname*{argmin}_{m_1=0,1,\ldots,4*NStep} f\left(x^{j,m_1}\right), \ j=1,\ldots,Np \ ;$$

для второго лидера:

$$x^{j,New} = \operatorname*{argmin}_{m_2=0,1,\ldots,2*NStep} f\left(x^{j,m_2}\right), \ j=\left(Np+1\right),\ldots,\left(2*Np\right);$$

– для третьего лидера:

$$x^{j,New} = \operatorname*{argmin}_{m_3=0,1,\ldots,NStep} f\left(x^{j,m_3}\right), \ j=\left(2*Np+1\right),\ldots,K \ ,$$

где $K=3*Np$.

Иллюстрация миграционного цикла приведена на рис. 1.

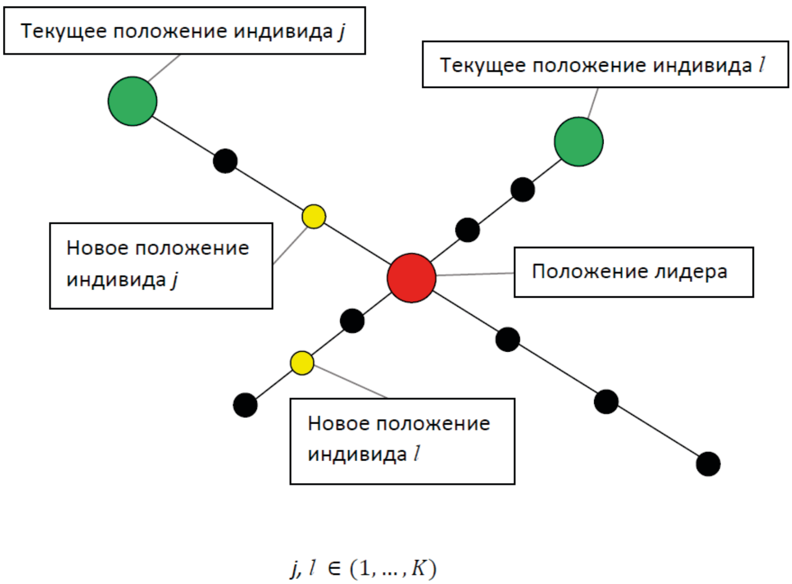

Рис. 1. Миграционный цикл

Шаг 4. *Тест на остановку алгоритма.*

Если $\sqrt{\frac{1}{2}\sum_{j=2}^{3}\left[f\left(x^j\right)-f\left(x^1\right)\right]^2} \geq MinDist$ и $MCount < Migration$, т.е. предельное число миграций (*Migration*) не достигнуто, то необходимо перейти к шагу 5, иначе перейти к шагу 6.

Шаг 5. *Обновление популяции.*

Шаг 5.1. Отсортировать всех индивидов по неубыванию целевой функции:
$\{x^1,\ldots,x^K | f(x^k \leq f\left(x^{k+1}\right)\}, k=1,\ldots,K-1.$

Шаг 5.2. Удалить последних $2*Np+\left[\frac{1}{3}*Np\right]$ особей, тем самым оставить только $\{x^1,\ldots,x^{\left[\frac{2}{3}*Np\right]}\}$.

Шаг 5.3. Создать $\left[\frac{1}{3}*Np\right]$ новых особей:

$$x_i^j=a_i+rand_i\left[0,1\right]*\left(b_i-a_i\right),\ j=\left[\frac{2}{3}*Np\right],\ldots,Np;\ i=1,\ldots,n.$$

Шаг 5.4. Увеличить счетчик числа итераций: $MCount=MCount+1$.

Шаг 5.5. Перейти к шагу 3.

Шаг 6. *Уточнение и остановка алгоритма.*

Шаг 6.1. Увеличить параметр *NStep* и реализовать миграционный цикл для второго и третьего лидеров относительно первого лидера:

$$NStep=10*NStep,$$

$$x^{j,m}=x^j+\frac{\left(x^1-x^j\right)}{\left[\frac{NStep}{2}\right]}mPRTVector\ ,\ \ j=1,2,3.$$

Шаг 6.2. Найти новые позиции лидеров:

$$x^{j,New}=\underset{m=0,1,\ldots,NStep}{\operatorname{argmin}}\ f\left(x^{j,m}\right),j=1,2,3.$$

Шаг 6.3. Возвращение лучшего решения из найденных в процессе поиска:

$$x^{ans}=\underset{j=1,2,3}{\operatorname{argmin}}\,f\left(x^j\right).$$

5. ПРИМЕРЫ ДЛЯ ТЕСТОВЫХ ФУНКЦИЙ

Предложенный алгоритм MSOMA реализован на языке python (ссылка на реализацию: https://github.com/Vradislav/MSOMA). Для тестирования использован стандартный набор тестовых функций со сложной структурой линий уровня.

1. Функция Изома

Целевая функция $f\left(x\right)=-\cos\left(x_1\right)\cos\left(x_2\right)\exp\left(-\left(\left(x_1-\pi\right)^2+\left(x_2-\pi\right)^2\right)\right)$, множество допустимых решений $x_1,x_2\in\left[-100,100\right]$, глобальный минимум: $x^*=\left(\pi;\pi\right)^T$, $f\left(x^*\right)=-1$.

Лучшее приближение:

```
min f(x) =  -1.0
x* =  [3.14159266 3.14159265]
loop =  9
--- run time 1.2546403408050537 seconds ---
```

Таблица 1.1

Статистика по 100 запускам MSOMA:

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migration*	*MinDist*			
30	0.4	800	40	10^{-6}	-0.604447390346022	-1.0	0.4671167934008718
50	0.4	1000	30	10^{-6}	-0.767532782657726	-1.0	0.4102030607415138
60	0.4	1500	40	10^{-6}	-0.851652624382647	-1.0	0.2924270375565159
60	0.4	2000	40	10^{-6}	-0.969999953334970	-1.0	0.2686198843558043
40	0.4	3000	40	10^{-10}	-0.96999999966223	-1.0	0.2306567303311588
100	0.6	3000	40	10^{-15}	**-1.0**	**-1.0**	**0**

Среднее время одного запуска MSOMA с параметрами, соответствующими наилучшему значению в таблице 1.1 (выделено полужирным) равно 127.7654502153396 с.

Таблица 1.2

Статистика по 100 запускам SOMA с близкими к наилучшим параметрами:

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migration*	*MinDist*			
600	0.6	12000	50	10^{-8}	-0.76999999966223	-0.9999998466342431	0.2534421942959418

Среднее время одного запуска SOMA из таблицы 1.2 равно 542.9886940987 с.

2. Функция «Птица»

Целевая функция

$f(x)=\sin x_1 \exp\left(\left(1-\cos x_2\right)^2\right)+\cos x_2 \exp\left(\left(1-\sin x_1\right)^2\right)+\left(x_1-x_2\right)^2$, множество допустимых решений $x_1, x_2 \in [-2\pi, 2\pi]$, глобальный минимум: $x^* = (-1,5708;-3,1416)^T$ или $x^* = (4,7124;3,1416)^T$, $f(x^*) = -106,7645$.

Лучшее приближение:

```
min f(x) =  -106.76453675
x* =  [4.70104503 3.15293953]
loop =  8
--- run time 0.18852782249450684 seconds ---

min f(x) =  -106.76229937
x* =  [-1.57810686 -3.13024376]
loop =  9
--- run time 0.3041861057281494 seconds ---
```

Таблица 2.1

Статистика по 100 запускам MSOMA:

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migration*	*MinDist*			
3	0.4	6	30	10^{-6}	-95.16180267326564	-106.7644859888339	19.2278631431097
5	0.4	10	30	10^{-6}	-103.3882558519979	-106.76453674926475	5.76252909071151
9	0.4	15	30	10^{-6}	-106.41919714064015	-106.76453674926475	3.99687148995614
9	0.5	15	30	10^{-6}	-106.64192928875836	-106.76453674926476	0.22181315811163
12	0.5	15	30	10^{-6}	-106.68484921251411	-106.76453674926476	0.14187788626850
12	0.7	20	30	10^{-10}	-106.74915552859784	-106.76453674926476	0.07380542967686
17	0.6	26	30	10^{-10}	-106.76422123629065	-106.76453674926476	0.00381473512963
20	0.7	30	40	10^{-12}	**-106.76453574967775**	**-106.76453674926478**	**1.293234249***10^{-6}

Среднее время одного запуска MSOMA с параметрами для лучшего значения в таблице 2.1 (выделено полужирным) равно 0.385252666 с.

Таблица 2.2

Статистика по 100 запускам SOMAс близкими к наилучшим параметрами:

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migration*	*MinDist*			
40	0.6	5000	40	10^{-6}	-105.728448775396	-106.764834888339	4.809178640550839

Среднее время одного запуска SOMA с параметрами из таблицы 2.2 равно 20.35154864717с.

3. Трехгорбая функция

Целевая функция $f(x)=2x_1^2-1,05x_1^4+\frac{1}{6}x_1^6+x_1x_2+x_2^2$, множество допустимых решений $x_1,x_2\in[-5,5]$, глобальный минимум: $x^*=(0;0)^T$, $f(x^*)=0$.

Лучшее приближение:

```
min f(x) =  0.0
x* =  [ 0. 0.]
loop =  14
--- run time 0.7091007232666016 seconds ---
```

Таблица 3.1

Статистика по 100 запускам MSOMA:

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migration*	*MinDist*			
3	0.4	3	10	10^{-6}	0.0138395554668576	1.57691322573e-05	0.0358463274474092
5	0.6	10	10	10^{-6}	0.0021472583344410	4.40027999425597e-17	0.0095383475872818
10	0.6	20	10	10^{-6}	2.61652408225*10^{-5}	3.356176441003*10^{-19}	2.61275996424*10^{-5}

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migration*	*Min-Dist*			
10	0.4	20	15	10^{-10}	$5.51295563145*10^{-5}$	$1.151300814085*10^{-18}$	0.0001630893051336
30	0.6	25	20	10^{-15}	**0.0**	**0.0**	**0.0**

Среднее время одного запуска MSOMA с параметрами для лучшего значения в таблице 3.1 (выделено полужирным) равно 0.2378336048126 с.

Таблица 3.2

Статистика по 100 запускам SOMA с близкими к наилучшим параметрами:

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migration*	*Min-Dist*			
10	0.7	15	20	0.01	0.0	0.0	0.0

Среднее время одного запуска SOMA с параметрами из таблицы 3.2 равно 0.00767917633056 с.

4. Функция Гольдман-Прайса

Целевая функция $f(x)=\left[1+\left(x_1+x_2+1\right)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]\times$ $\times\left[30+\left(2x_1-3x_2\right)^2\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$, множество допустимых решений $x_1,x_2\in[-2,2]$, глобальный минимум: $x^*=(0;-1)^T$, $f\left(x^*\right)=3$.

Лучшее приближение:

```
min f(x) =  3.0
x* =  [ 0. -1.]
loop =  9
--- run time 1.6824641227722168 seconds ---
```

Таблица 4.1

Статистика по 100 запускам MSOMA:

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migration*	*Min-Dist*			
8	0.6	8	25	10^{-6}	3.6754158626861346	3.000000000239916	3.3440062810127555
10	0.6	15	25	10^{-6}	3.0002451846253244	3.0000000000001856	3.946705398904461
20	0.6	20	40	10^{-6}	3.000012124599443	3.0000000000008584	0.00011742057057133
20	0.7	26	40	10^{-10}	3.000014048213801	3.0000000000006817	$1.400602458523*10^{-5}$
25	0.7	27	200	10^{-15}	3.0000029512408757	3.000000000000059	$9.141584157428*10^{-6}$
40	0.7	50	200	10^{-15}	**3.0**	**3.0**	**0.0**

Среднее время одного запуска MSOMA с параметрами для лучшего значения в таблице 4.1 (выделено полужирным) равно 3.114569830894с.

Таблица 4.2

Статистика по 100 запускам SOMA:

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migra-tion*	*Min-Dist*			
50	0.7	120	1000	10^{-12}	3.0000005148914557	3.0000000000431856	$1.5256024585*10^{-5}$

Среднее время одного запуска SOMA с параметрами из таблицы 4.2 равно 39.599019289с.

5. Функция Экли

Целевая функция

$f\left(x\right)=e-20\exp\left(-\sqrt{\dfrac{x_1^2+x_2^2}{50}}\right)-\exp\left(\dfrac{1}{2}\left(\cos\left(2\pi x_1\right)+\cos\left(2\pi x_2\right)\right)\right)$, множество допустимых решений $x_1, x_2 \in \left[-10,10\right]$, глобальный минимум: $x^* = \left(0;0\right)^T$, $f\left(x^*\right)=-20$.

Лучшее приближение:

```
min f(x) =  -20.0
x* =  [ 0. 0.]
loop =  14
--- run time 0.7051131725311279 seconds ---
```

Таблица 5.1

Статистика по 100 запускам MSOMA:

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migra-tion*	*Min-Dist*			
5	0.4	5	20	0.00001	-18.57383939296661	-19.999589170913417	2.200696233727464
10	0.4	10	20	0.00001	-19.86717335233902	-19.999999830888818	0.4567496220861112
10	0.5	15	20	0.0001	-19.98560628199538	-19.999999536804992	0.2477241407268142
15	0.5	20	100	0.0001	-19.97390653428714	-19.999998645676598	0.0259738220760433
20	0.6	30	100	0.0001	-19.99987555999442	-19.99999992744094	0.0154408047129368
20	0.6	30	100	10^{-10}	**-19.99996144678856**	**-20.0**	**5.1107563787***10^{-5}

Среднее время одного запуска MSOMA с параметрами для лучшего значения в таблице 5.1 (выделено полужирным) равно 0.688916547с.

Таблица 5.2

Статистика по 100 запускам SOMA с близкими к наилучшим параметрами:

Параметры					$\overline{f}$	Наименьшая величина f	$\overline{\sigma_f}$
NStep	*PRT*	*NP*	*Migra-tion*	*Min-Dist*			
10	0.7	15	20	0.01	-20.0	-20.0	0.0

Среднее время одного запуска SOMA с параметрами из таблицы 5.2 равно 0.009973287582397с.

Безусловно, время работы программы зависит как от реализации алгоритма, так и от конфигурации компьютера и поэтому является относительной величиной. Оно представлено только для сравнения скорости нахождения решения задачи оптимизации по сравнению с алгоритмом SOMA. Так как сложности алгоритмов SOMA и MSOMA почти одинаковы, точность найденных наилучших значений и среднее время работы одного запуска, соответствующие этому значению, показывают, какой из алгоритмов лучше справляется с конкретной тестовой функцией. Исходя из представленных результатов тестирования, можно сделать вывод, что алгоритм MSOMA быстрее и точнее работает с теми функциями, которые для базовой версии алгоритма SOMA являются более сложными. Это преимущество возникает из-за разного способа движения (движение в MSOMA покрывает большую область из-за движения вокруг трех разных лидеров и добавления новых особей) и разных условий остановки. Но одновременно, это различие в способах движения особей заставляет алгоритм MSOMA дольше работать с теми функциями, с которыми базовая версия алгоритма SOMA справляется достаточно быстро и точно. Таким образом, алгоритм MSOMA лучше показывает себя на тех функциях, на которых требуется большее покрытие множества допустимых решений для нахождения экстремума.

6. ЗАКЛЮЧЕНИЕ

Сформирован модифицированный алгоритм MSOMA для решения задачи условной глобальной оптимизации. На его основе создана программа на языке Python, работоспособность которой продемонстрирована на тестовых примерах. Проведен сравнительный анализ эффективности алгоритма по сравнению с базовым алгоритмом SOMA.

Литература

1. *Zelinka I., Lampinen J.* SOMA–Self-Organizing Migrating Algorithm // *Proceedings of the 6th International Conference on Soft Computing (Mendel 2000)*, Brno, Czech Republic, pp. 177–187.
2. *Zelinka I., Lampinen J., Noulle L.* On the theoretical proof of convergence for a class of SOMA search algorithms // *Proceedings of 7th International Conference on Soft Computing (Mendel 2001)*, Brno, Czech Republic, pp. 103–110.
3. *Davendra D., Zelinka I.* Self-Organizing Migrating Algorithm. Methodology and Implementation // Studies in Computational Intelligence, Vol. 626. Springer. 2016. V. 626.
4. *Пантелеев А.В.*, *Скавинская Д.В.* Метаэвристические алгоритмы глобальной оптимизации. – М.: Вузовская книга, 2019.
5. *Пантелеев А.В*. Метаэвристические алгоритмы оптимизации законов управления динамическими системами. – М.: Факториал, 2020.

Development of a Modified Self-Organizing Migration Optimization Algorithm (MSOMA)

Andrei V. Panteleev *
MAI (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000–0003–2493–3617
e-mail: avpanteleev@inbox.ru

Vladislav M. Rakitianskii**
MAI (National Research University), Moscow, Russia
ORCID: https://orcid.org/0000–0001–7894–7462
e-mail: rymbelv@gmail.com

The modified self-organizing migration optimization algorithm (MSOMA) based on a self-organizing migration algorithm (SOMA) is suggested. An algorithm for solving the problem of finding the global conditional extremum of the objective function on a given set is developed. Examples illustrating the application of the algorithm and created software are given.

Keywords: global optimization algorithm, migration cycle, population, individual, benchmark problems.

For citation:
Panteleev A.V., Rakitianskii V.M. Development of a Modified Self-Organizing Migration Optimization Algorithm (MSOMA). *Modelirovanie i analiz dannykh = Modelling and Data Analysis*, 2020. Vol. 10, no. 2, pp. 62–73. DOI:10.17759/ mda.2020100205 (In Russ., abstr. in Engl.).

References

1. Zelinka I., Lampinen J. SOMA–Self-Organizing Migrating Algorithm // *Proceedings of the 6th International Conference on Soft Computing (Mendel 2000)*, Brno, Czech Republic, pp. 177–187.
2. Zelinka I., Lampinen J., Noulle L. On the theoretical proof of convergence for a class of SOMA search algorithms // *Proceedings of 7th International Conference on Soft Computing (Mendel 2001)*, Brno, Czech Republic, pp. 103–110.
3. Davendra D., Zelinka I. Self-Organizing Migrating Algorithm. Methodology and Implementation // Studies in Computational Intelligence, Vol. 626. Springer. 2016. V. 626.
4. Пантелеев А.В., Скавинская Д.В. Метаэвристические алгоритмы глобальной оптимизации. – М.: Вузовская книга, 2019.
5. Пантелеев А.В. Метаэвристические алгоритмы оптимизации законов управления динамическими системами. – М.: Факториал, 2020.

****Panteleev Andrei Vladimirovich,*** Doctor of Physics and Mathematics, Professor, Head of the Department of Mathematics and Cybernetics, Institute of Information Technologies and Applied Mathematics, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000–0003–2493–361, e-mail: avpanteleev@inbox.ru

*****Rakitianskii Vladislav Maksimovich,*** Undergraduate Student of the Institute of Information Technology and Applied Mathematics, Moscow Aviation Institute (National Research University), Moscow, Russia, ORCID: https://orcid.org/0000–0001–7894–7462, e-mail: rymbelv@gmail.com.