УДК 519.85, 517.972.8

МОДУЛЬНЫЙ ГИБРИДНЫЙ МЕМЕТИЧЕСКИЙ АЛГОРИТМ ПОИСКА УСЛОВНОГО ГЛОБАЛЬНОГО ЭКСТРЕМУМА ФУНКЦИЙ МНОГИХ ПЕРЕМЕННЫХ

А.В. ПАНТЕЛЕЕВ, В.А. ПИСЬМЕННАЯ

В работе предложен гибридный меметический алгоритм поиска условного глобального экстремума функций. Данный алгоритм сочетает в себе свойства модульности и адаптивности, что обеспечивает алгоритму гибкость, настраиваемость и позволяет уменьшить степень влияния параметров. На основе предложенного алгоритма реализован комплекс программ на языке C#. Его эффективность продемонстрирована на широко распространенных тестовых задачах поиска глобального условного экстремума функций многих переменных.

Ключевые слова: меметический алгоритм, условный глобальный экстремум, мем, гибридный алгоритм, модульный алгоритм, гармонический поиск, дифференциальная эволюция, алгоритм опыления цветов.

ВВЕДЕНИЕ

Меметические алгоритмы (МА) представляют собой гибрид эволюционных методов оптимизации и улучшающих процедур локального поиска. Они основаны на понятии «мема», введенного Р. Докинзом и определенного как «единица передачи культурной информации, распространяемая от одной особи к другой посредством имитации, научения и др.» [1]. Термин «меметический алгоритм» (МА) был впервые предложен P. Moscato [2], где он рассматривал МА как гибрид генетического алгоритма и процедуры индивидуального обучения для уточнения решения задачи.

В работе представлен гибридный меметический алгоритм, который сочетает в себе свойства модульности и адаптивности, что позволяет обеспечивать гибкость и настраиваемость алгоритма, а также снизить степень влияния параметров алгоритма на его сходимость. Разработан комплекс программ, реализующий модульный гибридный меметический алгоритм, который для решения задачи локального улучшения использует методы дифференциальной эволюции [3, 4], алгоритмы опыления цветков [5] и гармонического поиска [6, 7]. С помощью разработанного комплекса решены распространенные задачи поиска условного глобального экстремума функций многих переменных.

В рамках решения сложных прикладных задач условной оптимизации меметические алгоритмы широко используются в различных областях и, как правило, показывают более точные результаты, чем классические эволюционные методы [8].

1. ПОСТАНОВКА ЗАДАЧИ

Дана целевая функция $f(x) = f(x_1, x_2, ..., x_n)$, определенная на множестве допустимых решений $D \subseteq R^n$. Требуется найти условный глобальный минимум функции $f(x)$ на множестве D, т.е. такую точку $x^* \in D$, что $f(x^*) = \min\limits_{x \in D} f(x)$, где $x = (x_1, x_2, ..., x_n)^T$, $D = \{x \mid x_i \in [a_i, b_i],\ i = 1, 2, ..., n\}$.

Задача поиска максимума функции $f(x)$ сводится к задаче поиска минимума путем замены знака перед функцией на противоположный: $f(x^*) = \max\limits_{x \in D} f(x) = -\min\limits_{x \in D}[-f(x)]$.

Функция $f(x)$ может быть многоэкстремальной, поэтому искомое решение в общем случае неединственное.

2. СТРАТЕГИЯ ПОИСКА РЕШЕНИЯ

Гибридный меметический алгоритм представляет собой синтез нескольких методов оптимизации, объединенных в адаптивный оптимизационный кластер. Основу описанного алгоритма составляют следующие модули:

1) модуль генерации особей, необходимый для создания множества начальных особей (точек, решений), на основе которого далее будут созданы популяции;

2) модуль группировки, разделяющий множество начальных особей на популяции;

3) модуль сдвига; в процессе функционирования данного модуля происходит развитие популяций согласно алгоритмам методов оптимизации, которые ассоциированы с каждой из популяций. В текущей версии алгоритма предлагается использовать следующие популяционные алгоритмы глобальной условной оптимизации для реализации модуля сдвига: гармонический поиск [6, 7], дифференциальная эволюция [3, 4, 8], алгоритм опыления цветов [5];

4) модуль обучения; данный модуль оперирует текущим положением особи в популяции для обучения остальных особей. В процессе работы модуля обучаемая особь сдвигается в сторону учителя (обучающей особи) с учетом заданного коэффициента обучения;

5) модуль имитации; данный модуль оперирует множеством векторов сдвига, полученных в результате реализации модуля сдвига и описывающих перемещение особи, для имитации поведения особей. В процессе работы модуля имитирующая особь повторяет сдвиг имитируемой особи с учетом заданного параметра имитации;

6) модуль локальной настройки; данный модуль позволяет выводить особь (решение) из области притяжения локального экстремума за счет реализации процесса обмена особями между популяциями;

7) модуль глобальной настройки; позволяет выводить текущие решения из области притяжения локального экстремума за счет смешивания и перегруппировки популяций.

3. АЛГОРИТМ

Шаг 1. Задание параметров алгоритма.

Задать параметры алгоритма: $ITER_{\max}$ – максимальное число итераций; OP_A – число используемых алгоритмов оптимизации, каждый из которых ассоциирован с популяцией $Pop^{id} = \left\{ x^{id,i} = \left(x_1^{id,i}, x_2^{id,i}, \ldots, x_n^{id,i} \right)^T \right\}_{i=1}^{N_A^{id}}$, где $id = 1,\ldots,OP_A$ – номер популяции, N_A^{id} – количество особей в популяции с номером id; N_{ITER}^{i}, $i = 1,\ldots,OP_A$ – число итераций алгоритма оптимизации, ассоциированного i-й популяцией.

Задать параметры модуля обучения (Teaching): $pr_T \in [0;1]$ – вероятность обучения; $\xi_T \in [0;1]$ – коэффициент обучения.

Задать параметры модуля имитации (Imitation): $pr_I \in [0;1]$ – вероятность имитации; $\xi_I \in [0;1]$ – коэффициент имитации.

Задать параметры модуля локальной настройки (Local Tuning): $freq_L \in \mathbb{N}$ – частоту; $b_L \in [0;1]$ – порог.

Задать параметры модуля глобальной настройки (Global Tuning): $freq_G \in \mathbb{N}$ – частоту; $b_G \in [0;1]$ – порог. Положить множество значений средней приспособленности особей всех популяций $Aver_{total} = \varnothing$ и средней приспособленности в популяции с номером id: $Aver_{gr}^{id} = \varnothing$, $id = 1,\ldots,OP_A$.

Шаг 2. Генерация особей. Сгенерировать $N_A = \sum_{id=1}^{OP_A} N_A^{id}$ точек $P = \left\{ x^i \right\}_{i=1}^{N_A}$, равномерно распределенных на множестве D.

Шаг 3. Группировка особей в популяции.

3.1. Положить $id = 1$.

3.2. Случайным образом выбрать из множества P N_A^{id} точек $x^{j_1}, x^{j_2}, \ldots, x^{j_{N_A^{id}}}$ и положить $\tilde{P} = \left\{ x^{j_i} \right\}_{i=1}^{N_A^{id}}$.

3.3. Положить $Pop^{id} = \tilde{P}$.

3.4. Положить $P = P \setminus \tilde{P}$, $id = id + 1$. Если $id > OP_A$, то перейти к шагу 4. В противном случае – к шагу 3.2.

Шаг 4. Добавить в множество $Aver_{total}$ величину $\frac{1}{\sum_{id=1}^{OP_A} N_A^{id}} \cdot \sum_{id=1}^{OP_A} \sum_{i=1}^{N_A^{id}} f\left(x^{id,i} \right)$, а в каждое множество $Aver_{gr}^{id}$ величину $\frac{1}{N_A^{id}} \cdot \sum_{i=1}^{N_A^{id}} f\left(x^{id,i} \right)$.

Шаг 5. Задать номер текущей итерации $iter = 1$, перейти к шагу 6.

Шаг 6. Модуль сдвига.

6.1. Положить $id = 1$.

6.2. Положить $Pop_{prev}^{id} = \left\{ x_{prev}^{id,i} \right\}_{i=1}^{N_A^{id}} = Pop^{id}$.

6.3. Выполнить N_{ITER}^{id} итераций алгоритма оптимизации, ассоциированного с популяцией под номером id.

6.4. Найти множество векторов сдвига для текущей популяции $\Delta_x^{id} = \left\{ \delta_x^{id,i} = x^{id,i} - x_{prev}^{id,i} \right\}_{i=1}^{N_A^{id}}$.

6.5. Найти множество приращений целевой функции $\Delta_f^{id} = \left\{ \delta_f^{id,i} = f\left(x^{id,i} \right) - f\left(x_{prev}^{id,i} \right) \right\}_{i=1}^{N_A^{id}}$ для текущей популяции.

6.6. Положить $id = id + 1$. Если $id > OP_A$, то перейти к шагу 7. В противном случае – к шагу 6.2.

Шаг 7. Добавить в множество $Aver_{total}$ величину $\frac{1}{\sum_{id=1}^{OP_A} N_A^{id}} \cdot \sum_{id=1}^{OP_A} \sum_{i=1}^{N_A^{id}} f\left(x^{id,i} \right)$, а в каждое множество $Aver_{gr}^{id}$ величину $\frac{1}{N_A^{id}} \cdot \sum_{i=1}^{N_A^{id}} f\left(x^{id,i} \right)$.

Шаг 8. Модуль обучения.

8.1. Положить $id = 1$.

8.2. Положить $i = 1$.

8.3. Сгенерировать случайную величину $p \sim R(0,1)$, где $R(0,1)$ – равномерное распределение на интервале $(0,1)$. Если $p < pr_T$, то перейти к шагу 8.4. В противном случае – к шагу 8.7.

8.4. Найти номер особи, ближайшей к текущей: $j^* = \underset{j\in\{1,2,\ldots,N_A^{id}\}\setminus i}{\operatorname{Arg\,min}} d_E(x^{id,i}, x^{id,j})$, где $d_E\left(x^i, x^j\right) = \sqrt{\sum_{k=1}^{n}\left(x_k^i - x_k^j\right)^2}$.

8.5. Найти положение обученной особи по формуле $\tilde{x} = x^{id,j^*} + \xi_T \cdot \left(x^{id,i} - x^{id,j^*}\right)$.

8.6. Если $f(\tilde{x}) < f\left(x^{id,j^*}\right)$, то заменить x^{id,j^*} на $\tilde{x}$.

8.7. Положить $i = i+1$. Если $i > N_A^{id}$, то перейти к шагу 8.8. В противном случае – к шагу 8.3.

8.8. Положить $id = id+1$. Если $id > OP_A$, то перейти к шагу 9. В противном случае – к шагу 8.2.

Шаг 9. Модуль имитации.

9.1. Положить $id = 1$.

9.2. Найти сумму всех приращений целевой функции $\delta_{total,f} = \sum_{i=1}^{N_A^{id}} \min\left\{0, \delta_f^{id,i}\right\}$, где $\delta_f^{id,i} = f\left(x^{id,i}\right) - f\left(x_{prev}^{id,i}\right)$. Если $\delta_{total,f} = 0$, то положить вероятности имитации i-го вектора сдвига $pr_i = \frac{1}{N_A^{id}}$ и перейти к шагу 9.4. В противном случае перейти к шагу 9.3.

9.3. Найти вероятности имитации i-го вектора сдвига по формуле $pr_i = \frac{1}{\delta_{total,f}} \cdot \min\left\{0, \delta_f^{id,i}\right\}$.

9.4. Положить $i = 1$.

9.5. Сгенерировать случайную величину $p \sim R(0,1)$. Если $p < pr_I$, то перейти к шагу 9.6. В противном случае – к шагу 9.11.

9.6. Сгенерировать случайную величину $p \sim R(0,1)$. Положить $j = 1$.

9.7. Если $p < pr_j$, то перейти к шагу 9.9. В противном случае – к шагу 9.8.

9.8. Положить $p = p - pr_j$, $j = j+1$ и перейти к шагу 9.7.

9.9. Найти положение особи после имитации по формуле $\tilde{x} = x^{id,i} + \xi_I \cdot \delta_x^{id,j}$, где $\delta_x^{id,i} = x^{id,i} - x_{prev}^{id,i}$.

9.10. Если $f(\tilde{x}) < f\left(x^{id,i}\right)$, то заменить $x^{id,i}$ на $\tilde{x}$.

9.11. Положить $i = i+1$. Если $i > N_A^{id}$, то перейти к шагу 9.12. В противном случае – к шагу 9.5.

9.12. Положить $id = id+1$. Если $id > OP_A$, то перейти к шагу 10. В противном случае – к шагу 9.4.

Шаг 10. Модуль локальной настройки.

10.1. Если $iter - \left[iter / freq_L\right] = 0$ (где $[\cdot]$ – целая часть от деления), то перейти к шагу 10.2. В противном случае – к шагу 11.

10.2. Задать множества $Good = \varnothing$ и $Bad = \varnothing$ – «хороших» и «плохих» групп соответственно. Положить $id = 1$.

10.3. Определить величину $r=\dfrac{Aver_{gr}^{id,iter-1}-Aver_{gr}^{id,iter}}{\left|Aver_{gr}^{id,iter-1}\right|}$, где $|\cdot|$ – количество элементов множества. Если $r<b_L$, то добавить id в множество Bad, в противном случае – в множество $Good$.

10.4. Положить $id=id+1$. Если $id>OP_A$, то перейти к шагу 10.5. В противном случае – к шагу 10.3.

10.5. Если $|Good|=OP_A$, то перейти к шагу 11. В противном случае – к шагу 10.6.

10.6. Если $|Bad|=OP_A$, то выбрать два случайных номера id_c^1 и id_c^2 из множества Bad и перейти к шагу 10.8. В противном случае – к шагу 10.7.

10.7. Выбрать случайный номер id_c^1 из множества $Good$ и id_c^2 из множества Bad.

10.8. Найти элементы $x_g=\operatorname{Arg}\min\limits_{x\in Pop^{id_c^1}} f(x)$ и $x_b=\operatorname{Arg}\max\limits_{x\in Pop^{id_c^2}} f(x)$ и поменять их местами.

Шаг 11. Модуль глобальной настройки.

11.1. Если $iter-\left[iter/freq_G\right]=0$, то перейти к шагу 11.2. В противном случае – к шагу 12.

11.2. Определить величину $r=\dfrac{Aver_{total}^{iter-1}-Aver_{total}^{iter}}{\left|Aver_{total}^{iter-1}\right|}$. Если $r<b_G$, то перейти к шагу 11.3. В противном случае – к шагу 12.

11.3. Положить $P=Pop^1\cup\ldots\cup Pop^{OP_A}$, $id=1$.

11.4. Случайным образом выбрать из множества P N_A^{id} точек и положить $\tilde{P}=\left\{x^{j_i}\right\}_{i=1}^{N_A^{id}}$.

11.5. Положить $Pop^{id}=\tilde{P}$.

11.6. Положить $P=P\setminus\tilde{P}$, $id=id+1$. Если $id>OP_A$, то перейти к шагу 12. В противном случае – к шагу 11.4.

Шаг 12. Положить $iter=iter+1$. Если $iter>ITER_{\max}$, то перейти к шагу 13, в противном случае – к шагу 6.

Шаг 13. Выбор решения. В качестве решения взять $x^*=\operatorname{Arg}\min\limits_{x\in Pop^1\cup\ldots Pop^{OP_A}} f(x)$.

На основе предложенного алгоритма разработано программное обеспечение для поиска глобального минимума функций. Среда разработки – Microsoft Visual Studio, язык программирования – C#.

4. РЕКОМЕНДАЦИИ ПО ВЫБОРУ ПАРАМЕТРОВ МЕТОДА

Максимальное количество итераций $ITER_{\max}$ определяет, как долго будет продолжаться поиск новых решений. Рекомендуемое значение параметра: $ITER_{\max}=100\div 200$.

Число используемых алгоритмов оптимизации OP_A соответствует количеству популяций особей. Рекомендуемое значение данного параметра: $OP_A=3\div 5$.

Значения параметров вероятности обучения и имитации pr_T и pr_I, близкие к 1, значительно увеличивают расход вычислительных ресурсов и время работы алгоритма. Рекомендуемые значения параметров: $pr_T=0,5-0,7$, $pr_I=0,4-0,6$.

Чем меньше значения параметров ξ_T и ξ_I, тем тщательнее исследуется множество допустимых решений. Рекомендуемые значения параметров: $\xi_T=0,4-0,6$, $\xi_I=0,5-0,7$.

Чем меньше частота локальной настройки $freq_L$, тем чаще происходит обмен особями между разными популяциями. В свою очередь, чем меньше значение параметра частоты глобальной настройки $freq_G$, тем чаще может происходить формирование новых популяций. Рекомендуемые значения параметров: $freq_L = 10 \div 15$, $freq_L = 20 \div 25$.

Чем меньше значение параметра b_L, тем больше популяций помещаются в множество $Good$. Порог глобальной настройки b_G определяет частоту обновления популяций. Рекомендуемое значение данного параметра: $b_L = 0,3 - 0,5$, $b_G = 0,4 - 0,5$.

5. ТЕСТОВЫЕ ПРИМЕРЫ. АНАЛИЗ ЭФФЕКТИВНОСТИ МЕТОДА

Во всех примерах, приведенных ниже, проводились серии из 100 решений одной и той же задачи с одними и теми же значениями параметров. Для полученной выборки $\{f^1, f^2, ..., f^{100}\}$ вычислялись выборочное среднее $\overline{f} = \frac{1}{100}\sum_{i=1}^{100} f^i$ и среднеквадратическое отклонение $\overline{\sigma}_f = \sqrt{\overline{S}_{100}}$, $\overline{S}_{100} = \frac{1}{99}\sum_{i=1}^{100}(f^i - \overline{f})^2$.

Пример 1. Функция Растригина: $f(x,y) = 20 + x^2 + y^2 - 10(\cos 2\pi x + \cos 2\pi y)$, область поиска $D = [-100;100] \times [-100;100]$. Точное решение: $x^* = y^* = 0$, $f(x^*, y^*) = 0$.

Результаты решения примера отражены в табл. 1.

Таблица 1

Влияние параметров меметического алгоритма. Функция Растригина

Параметры алгоритмов		$\overline{f}$	$f_{наим}$	$\overline{\sigma}$
Memetic	$ITER_{\max} = 50$; $OP_A = 3$; $pr_T = 0,6$; $\xi_T = 0,5$; $pr_I = 0,6$; $\xi_I = 0,5$; $freq_L = 10$; $b_L = 0,5$; $freq_G = 20$; $b_G = 0,5$	0,00013	0	0,00153
HS	$N_{iter} = 20$; $HMS = 100$; $HMCR = 0,8$; $PAR = 0,2$; $bw = (50;50)$			
DE	$M = 20$; $NP = 20$; $F = 0,8$; $CR = 0,9$			
FPA	$N_{iter} = 20$; $n = 25$; $p = 0,8$; $\gamma = 0,1$; $\lambda = 1,5$			
Memetic	$ITER_{\max} = 100$; $OP_A = 3$; $pr_T = 0,6$; $\xi_T = 0,5$; $pr_I = 0,6$; $\xi_I = 0,5$; $freq_L = 10$; $b_L = 0,5$; $freq_G = 20$; $b_G = 0,5$	0	0	$3,24 \cdot 10^{-9}$
HS	$N_{iter} = 50$; $HMS = 100$; $HMCR = 0,8$; $PAR = 0,2$; $bw = (50;50)$			
DE	$M = 50$; $NP = 20$; $F = 0,8$; $CR = 0,9$			
FPA	$N_{iter} = 50$; $n = 25$; $p = 0,8$; $\gamma = 0,1$; $\lambda = 1,5$			
Memetic	$ITER_{\max} = 50$; $OP_A = 3$; $pr_T = 0,3$; $\xi_T = 0,3$; $pr_I = 0,3$; $\xi_I = 0,3$; $freq_L = 10$; $b_L = 0,5$; $freq_G = 20$; $b_G = 0,5$	0,00151	$2,36 \cdot 10^{-6}$	$1,17 \cdot 10^{-5}$
HS	$N_{iter} = 20$; $HMS = 100$; $HMCR = 0,8$; $PAR = 0,2$; $bw = (50;50)$			
DE	$M = 20$; $NP = 20$; $F = 0,8$; $CR = 0,9$			
FPA	$N_{iter} = 20$; $n = 25$; $p = 0,8$; $\gamma = 0,1$; $\lambda = 1,5$			

Пример 2. Функция Швефеля: $f(x,y) = -837{,}9658 - x \cdot \sin\sqrt{|x|} - y \cdot \sin\sqrt{|y|}$, область поиска $D = [-100;100] \times [-100;100]$. Точное решение: $x^* = y^* = 420{,}96$, $f(x^*, y^*) = -837{,}9658$. Результаты решения примера отражены в табл. 2.

Таблица 2

Влияние параметров меметического алгоритма. Функция Швефеля

Параметры алгоритмов		$\overline{f}$	$f_{наим}$	$\overline{\sigma}$
Memetic	$ITER_{\max} = 50$; $OP_A = 3$; $pr_T = 0{,}6$; $\xi_T = 0{,}5$; $pr_I = 0{,}6$; $\xi_I = 0{,}5$; $freq_L = 10$; $b_L = 0{,}5$; $freq_G = 20$; $b_G = 0{,}5$	–765,865	–829,432	56,124
HS	$N_{iter} = 20$; $HMS = 100$; $HMCR = 0{,}8$; $PAR = 0{,}2$; $bw = (50;50)$			
DE	$M = 20$; $NP = 20$; $F = 0{,}8$; $CR = 0{,}9$			
FPA	$N_{iter} = 20$; $n = 25$; $p = 0{,}8$; $\gamma = 0{,}1$; $\lambda = 1{,}5$			
Memetic	$ITER_{\max} = 100$; $OP_A = 3$; $pr_T = 0{,}6$; $\xi_T = 0{,}5$; $pr_I = 0{,}6$; $\xi_I = 0{,}5$; $freq_L = 10$; $b_L = 0{,}5$; $freq_G = 20$; $b_G = 0{,}5$	–833,194	–837,958	5,835
HS	$N_{iter} = 50$; $HMS = 100$; $HMCR = 0{,}8$; $PAR = 0{,}2$; $bw = (50;50)$			
DE	$M = 20$; $NP = 20$; $F = 0{,}8$; $CR = 0{,}9$			
FPA	$N_{iter} = 50$; $n = 50$; $p = 0{,}8$; $\gamma = 0{,}1$; $\lambda = 1{,}5$			
Memetic	$ITER_{\max} = 50$; $OP_A = 3$; $pr_T = 0{,}6$; $\xi_T = 0{,}5$; $pr_I = 0{,}6$; $\xi_I = 0{,}5$; $freq_L = 5$; $b_L = 0{,}3$; $freq_G = 15$; $b_G = 0{,}3$	–815,904	–837,594	27,393
HS	$N_{iter} = 20$; $HMS = 100$; $HMCR = 0{,}8$; $PAR = 0{,}2$; $bw = (50;50)$			
DE	$M = 20$; $NP = 20$; $F = 0{,}8$; $CR = 0{,}9$			
FPA	$N_{iter} = 20$; $n = 25$; $p = 0{,}8$; $\gamma = 0{,}1$; $\lambda = 1{,}5$			

Данные, приведенные в табл. 1 и 2, свидетельствуют об эффективной работе описанного алгоритма. Приемлемые результаты решения задач достигаются за небольшое количество итераций.

ЗАКЛЮЧЕНИЕ

В работе предложен модульный гибридный меметический алгоритм для решения задачи поиска условного глобального экстремума функций. Реализован комплекс программ, позволяющий применять разработанный алгоритм для различных функций, а также проводить анализ эффективности его работы. Эффективность метода продемонстрирована на нескольких модельных примерах, обладающих как простой, так и сложной структурой линий уровня.

СПИСОК ЛИТЕРАТУРЫ

1. Dawkins R. Universal Darwinism in D.S. Bendall (ed.), Evolution: From Molecules to Men. – Cambridge: Cambridge University Press. 1983. P. 403–425.

2. Moscato P. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms // Caltech Concurrent Computation Program (report 826). Pasadena, California, USA. 1989.

3. Storn R., Price K. Differential evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces // Journal of Global Optimization, Kluwer Academic Publishers. Vol. 11. 1997. P. 341–359.

4. Пантелеев А.В., Дмитраков И.Ф. Применение метода дифференциальной эволюции и его модификаций в задаче поиска оптимального управления дискретными детерминированными системами // Научный вестник МГТУ ГА. – 2011. – № 169 (7). – С. 5–12.

5. Yang X.-S. Flower Pollination Algorithm for Global Optimization // Unconventional Computation and Natural Computation. 2012. Lecture Notes in Computer Science. Vol. 7445. P. 240–249.

6. Chukiat Worasucheep. A Harmony Search with Adaptive Pitch Adjustment for Continuous Optimization // International Journal of Hybrid Information Technology. 2011. Vol. 4. No. 4. P. 13–24.

7. Geem Z.W., Kim J.H., Loganathan G.V. A New Heuristic Optimization Algorithm: Harmony Search // Simulations. 2001. Vol. 76. P. 60–68.

8. Пантелеев А.В., Метлицкая Д.В., Алешина Е.А. Методы глобальной оптимизации. Метаэвристические стратегии и алгоритмы. – М: Вузовская книга, 2013. – 244 с.

MODULAR HYBRID MEMETIC ALGORITHM FOR FINDING A CONDTIONAL GLOBAL EXTREMUM FOR FUNCTIONS OF SEVERAL VARIABLES

Panteleev A.V., Pismennaya V.A.

This paper presents a hybrid memetic algorithm for finding a conditional global extremum of functions. The algorithm combines such characteristics as modularity and adaptability which provides flexibility and controllability of the algorithm and reduces the influence of parameters. On the basis of the proposed algorithm the software complex is formed in the C# language. The method effectiveness is demonstrated on several well-known model examples of finding a conditional global extremum for functions of several variables.

Key words: memetic algorithm, conditional global extremum, meme, hybrid algorithm, modular algorithm, harmony search, differential evolution, flower pollination algorithm.

REFERENCES

1. Dawkins R. Universal Darwinism in D.S. Bendall (ed.), Evolution: From Molecules to Men. Cambridge: Cambridge University Press, 1983. Pp. 403–425.

2. Moscato P. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Caltech Concurrent Computation Program (report 826). Pasadena, California, USA. 1989.

3. Storn R., Price K. Differential evolution – a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization, Kluwer Academic Publishers. Vol. 11. 1997. Pp. 341–359.

4. Panteleev A.V., Dmitrakov I.F. Primenenie metoda differencial'noj ehvolyucii i ego modifikacij v zadache poiska optimal'nogo upravleniya diskretnymi determinirovannymi sistemami (Application of the differential evolution method and its modifications in the problem of finding optimal control of discrete deterministic systems). Scientific Bulletin of MGTU GA. 2011. No. 169 (7). Pp. 5–12. (in Russian).

5. Yang X.-S. Flower Pollination Algorithm for Global Optimization. Unconventional Computation and Natural Computation. 2012. Lecture Notes in Computer Science, Vol. 7445. Pp. 240–249.

6. Chukiat Worasucheep. A Harmony Search with Adaptive Pitch Adjustment for Continuous Optimization. International Journal of Hybrid Information Technology. 2011. Vol. 4. No. 4. Pp. 13–24.

7. Geem Z.W., Kim J.H., Loganathan G.V. A New Heuristic Optimization Algorithm: Harmony Search. Simulations. 2001. Vol. 76. Pp. 60–68.

8. **Panteleev A.V., Metlitskaya D.V., Aleshina E.A.** Metody global'noi optimizatsii. Metaevristicheskie strategii i algoritmy. (Global optimization methods, Metaheuristic strategies and algorithms), Moscow, Vuzovskaya kniga, 2013, 244 p. (in Russian).

СВЕДЕНИЯ ОБ АВТОРАХ

Пантелеев Андрей Владимирович, 1955 г.р., окончил МГТУ им. Н.Э. Баумана (1978). Доктор физико-математических наук, профессор, заведующий кафедрой математической кибернетики факультета «Прикладная математика и физика» Московского авиационного института (Национального исследовательского университета), автор более 150 научных публикаций и 34 книг. Области научных интересов – методы синтеза оптимальных нелинейных систем управления, методы оптимизации, электронный адрес: dep805@mai.ru.

Письменная Виктория Александровна, 1991 г.р., окончила МАИ (2014), аспирантка факультета «Прикладная математика и физика» Московского авиационного института. Области научных интересов – методы оптимизации, метаэвристические алгоритмы, электронный адрес: wildangel9@yandex.ru.