

УДК 573.874

ОБОБЩЕННЫЙ ИНВЕРСНЫЙ ИНТЕРВАЛЬНЫЙ МЕТОД ГЛОБАЛЬНОЙ УСЛОВНОЙ ОПТИМИЗАЦИИ¹

А.В. ПАНТЕЛЕЕВ, В.Н. ПАНОВСКИЙ

Разработано алгоритмическое и программное обеспечение инверсного интервального метода глобальной условной оптимизации, приведены решения модельных примеров, доказательства теорем о сходимости методов. Разработана обобщенная модульная схема, имеющая два заменяемых модуля проверки и сжатия. Предложенный модульный подход позволяет пользователям реализовывать собственные варианты модулей алгоритма без потери сходимости метода в зависимости от особенностей решаемой задачи.

Ключевые слова: интервальный анализ, глобальный условный экстремум, инверсный интервальный метод, сходимость, программное обеспечение.

Введение

В современной математике достаточно большое внимание уделяется решению задач глобальной оптимизации [1-3]. Эти задачи возникают в ходе проектирования конструкций самолетов, вертолетов, космических аппаратов, когда возникает необходимость оптимизации характерных параметров (вес, дальность полета, аэродинамические характеристики). Существующие численные методы используют разнообразные подходы, но их использование связано с большими вычислительными нагрузками, излишними требованиями к постановке задачи, трудностями в достижении сходимости метода.

В качестве базового аппарата, составляющего основу работы алгоритмов, используется интервальный анализ [4-6]. Существующие интервальные методы оптимизации можно разделить на методы безусловной оптимизации (алгоритмы Мура-Скелбоу, Ичиды-Фуджи, Дюсселя, интервальный алгоритм «имитации отжига», метод случайного интервального дробления и др. [7-9]) и условной оптимизации (методы Хансена, Мура и др. [10; 11]).

Предлагаемый интервальный метод глобальной условной оптимизации в процессе поиска использует операцию инвертор [4] и имеет модульную структуру. В частных случаях задания модулей он совпадает с методом дихотомии прямого образа, методом отсечки и стохастической отсечки виртуальных значений [12; 13], которые относятся к группе методов дробления графика [9].

1. Используемые понятия и обозначения

Основной идеей интервального анализа является окружение вещественных чисел интервалами, а вещественных векторов – интервальными векторами или параллелотопами [4]. Для обозначения интервала используются строчные латинские буквы, заключенные в квадратные скобки $([a], [b], [c], \dots)$, или привычное представление $([a_l, a_u], [b_l, b_u], [c_l, c_u], \dots)$, где указываются нижняя и верхняя границы. Для задания параллелотопа применяется то же обозначение с полужирным начертанием букв $([\mathbf{a}], [\mathbf{b}], [\mathbf{c}], \dots)$ или как прямое произведение интервалов $([1; 3] \times [2; 3] \times \dots \times [10; 11])$.

Для произвольного интервала $[x]$ определены: нижняя граница (обозначается как $lb([x])$ или \underline{x}) - $\underline{x} = lb([x]) = \sup \{ \xi \in \mathfrak{R} \cup \{-\infty, \infty\} \mid \forall \zeta \in [x], \xi \leq \zeta \}$; верхняя граница (обозначается как $ub([x])$ или \overline{x}) - $\overline{x} = ub([x]) = \inf \{ \xi \in \mathfrak{R} \cup \{-\infty, \infty\} \mid \forall \zeta \in [x], \zeta \leq \xi \}$; ширина (обозначается как

¹ Работа выполнена при финансовой поддержке РФФИ (проект № 12-08-00892-а).

$\omega([x])$, определена только для непустого интервала - $\omega([x]) = \overline{[x]} - \underline{[x]}$; средняя точка (обозначается как $mid([x])$, определена только для ограниченного и непустого интервала) -

$$mid([x]) = \frac{\underline{[x]} + \overline{[x]}}{2}.$$

Те же параметры определены и для параллелотопов. Нижняя и верхняя границы и средняя точка становятся векторами, ширина же рассчитывается как максимум из ширин всех образующих параллелотоп компонент.

Интервальной оболочкой множества $X \subset \mathfrak{R}^n$ называется параллелотоп с наименьшей шириной, который содержит X . Если множество берется в квадратные скобки, это значит, что рассматривается интервальная оболочка этого множества.

Пусть \circ - некоторая бинарная операция, тогда $[x] \circ [y] = [\{\xi_1 \circ \xi_2 \mid \xi_1 \in [x], \xi_2 \in [y]\}]$; пусть f - некоторый унарный оператор, тогда $f([x]) = [\{f(\xi) \mid \xi \in [x]\}]$. Описанная бинарная операция позволяет определить арифметические операции над интервалами [4].

Множество интервалов обозначается как $\mathfrak{I}\mathfrak{R}$, интервальных векторов - как $\mathfrak{I}\mathfrak{R}^n$. Пусть имеется некоторая функция f , действующая из \mathfrak{R}^n в \mathfrak{R} . Функция $[f](\cdot)$ называется интервальной функцией включения для f , действующей из $\mathfrak{I}\mathfrak{R}^n$ в $\mathfrak{I}\mathfrak{R}$, если $f([\mathbf{x}]) = \{f(\xi) \mid \xi \in [\mathbf{x}]\} \subset [f]([\mathbf{x}]), \forall [\mathbf{x}] \in \mathfrak{I}\mathfrak{R}^n$. Функция включения позволяет получить априорную оценку множества значений функции, даже если оно не является выпуклым или связным (если вместо переменных используются интервалы и соответствующие арифметические операции, то полученная оценка называется оценкой прямого образа функции).

Инвертор $INV(f, [s], [l], \varepsilon)$ - функция, которая по заданному интервалу $[l]$, функции f и параметру точности ε находит в области поиска $[s]$ множество параллелотопов $P \in \mathfrak{P}$ такое, что $\forall [\mathbf{x}] \in P$ справедливо: $[\mathbf{x}] \subseteq [s], \omega([\mathbf{x}]) \geq \varepsilon, [f]([\mathbf{x}]) \subseteq [l]$ или $[\mathbf{x}] \subseteq [s], \omega([\mathbf{x}]) < \varepsilon, [f]([\mathbf{x}]) \cap [l] \neq \emptyset$.

2. Алгоритм SIVIA

В качестве реализации инвертора можно использовать алгоритм SIVIA [4].

Шаг 1. Задать функцию f , область поиска $[s]$, интервал $[l]$ и параметр точности ε . Создать пустое множество параллелотопов P и множество $X = \{[s]\}$.

Шаг 2. Для каждого параллелотопа $[\mathbf{x}] \in X$ найти $[f]([\mathbf{x}])$. Проверить выполнение следующих условий (сам параллелотоп $[\mathbf{x}]$ после проверки условий удаляется из X):

- если $[f]([\mathbf{x}]) \cap [l] = \emptyset$, то параллелотоп $[\mathbf{x}]$ не добавляется в множество P ;
- если $[f]([\mathbf{x}]) \subseteq [l]$ или $[f]([\mathbf{x}]) \cap [l] \neq \emptyset, \omega([\mathbf{x}]) < \varepsilon$, добавить параллелотоп $[\mathbf{x}]$ в множество P ;
- в противном случае задать параллелотопы

$$[\mathbf{x}]' = [x_1]_i \times \dots \times [mid([x_k]_i); \overline{[x_k]_i}] \times \dots \times [x_n]_i \quad \text{и} \quad [\mathbf{x}]'' = [x_1]_i \times \dots \times [\underline{[x_k]_i}; mid([x_k]_i)] \times \dots \times [x_n]_i,$$

где k - номер компоненты параллелотопа $[\mathbf{x}]_i$ с наибольшей шириной, и добавить их в множество X .

Шаг 3. Если $X \neq \emptyset$, то перейти к шагу 2, в противном случае – к шагу 4.

Шаг 4. Вернуть P .

3. Постановка задачи

Пусть имеются некоторый параллелотоп $[s]$ и заданная функция $f: \mathcal{R}^n \rightarrow \mathcal{R}$. Задача интервальной ε – оптимизации формулируется следующим образом: найти параллелотоп $[p]^*$, такой что

$$[p]^* \subseteq [s], \omega([p]^*) \leq \varepsilon, \exists [x] \subseteq [s], \omega([x]) \geq \varepsilon: \overline{[f]}([x]) \subseteq \underline{[f]}([p]^*). \quad (1)$$

4. Стратегия поиска решения

Предлагаемый подход называется обобщенным инверсным интервальным методом, поскольку содержит взаимозаменяемые модули и в процессе поиска использует инвертор (рис. 1).

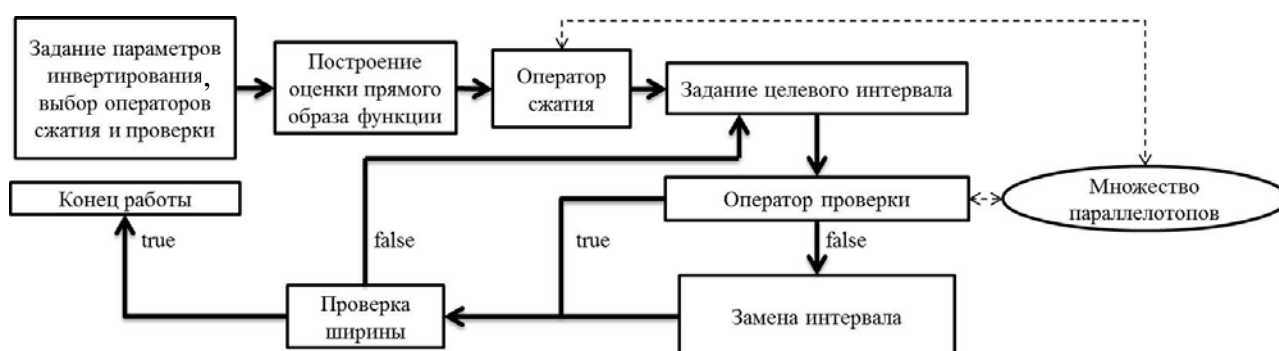


Рис. 1. Схема обобщенного детерминированного интервального метода

В качестве заменяемых модулей выступают операторы проверки и операторы сжатия:

- операторы проверки (результатом работы является статус проверки: удачный свидетельствует о том, что в множестве параллелотопов X есть параллелотоп $[p]$, для которого выполнено $\omega([p]) \geq \varepsilon, [f]([p]) \subseteq [l]$ или $\omega([p]) < \varepsilon, [f]([p]) \cap [l] \neq \emptyset$):

- Original Invertor operator (OI – operator) – базовая проверка, используемая в методе дихотомии прямого образа;

- Original Invertor with Renewal operator (OIR – operator) – в дополнение к базовой проверке, после каждой успешной операции, множество параллелотопов заменяется на множество, выработанное инвертором;

- First True with Renewal operator (FTR – operator) – проверка до первого подходящего элемента, множество заменяется на множество, выработанное инвертором;

- операторы сжатия (результатом работы является сжатие оценки прямого образа функции на области поиска):

- Search Area Split operator (SAS – operator) – сжатие на основе разбиения области поиска;

- Random Point Sample operator (RPS – operator) – сжатие на основе значений функций в случайно сгенерированных интервальных векторах нулевой ширины.

5. Алгоритм

Шаг 1. Задать $[s]$ – область поиска; ε – параметр точности, отвечающий за размер параллелотопа во множестве P , которое будет вырабатываться инвертором; ζ – параметр точности

для останковки алгоритма. Выбрать оператор сжатия с параметром w или A и оператор проверки с параметром w . Задать множество $X = \{[s]\}$.

Шаг 2. С помощью функции включения $[f]$ найти оценку прямого образа $[Y]$ функции f на области поиска $[s]$.

Шаг 3. Заменить полученную оценку $[Y]$ интервалом $[\tilde{Y}]$, полученным с помощью выбранного оператора сжатия с параметром w или A .

Шаг 4. Дихотомически разделить интервал $[Y] = [y_l; y_u]$ на два интервала $[Y_l] = \left[y_l; \frac{y_l + y_u}{2} \right]$ и $[Y_u] = \left[\frac{y_l + y_u}{2}; y_u \right]$.

Шаг 5. Проверить интервал $[Y_l]$ выбранным оператором проверки.

Шаг 6. Если проверка завершилась удачно, перейти к шагу 7, если нет – к шагу 8.

Шаг 7. Если $\omega([Y_l]) < \zeta$, тогда найти $P = INV(f, [s], [Y_l], \varepsilon)$ и перейти к шагу 9. В противном случае положить интервал $[Y]$, равным $[Y_l]$, и перейти к шагу 4.

Шаг 8. Положить интервал $[Y]$, равным $[Y_u]$. Если $\omega([Y]) < \zeta$, то найти $P = INV(f, [s], [Y], \varepsilon)$ и перейти к шагу 9. В противном случае перейти к шагу 4.

Шаг 9. Выбор параллелопада.

Шаг 9.1. Положить $i = 1$, множество $P_\varepsilon = P$.

Шаг 9.2. Если $\omega([p]_i) \leq \varepsilon$, где $[p]_i \in P_\varepsilon$, то перейти к шагу 9.4. Иначе перейти к шагу 9.3.

Шаг 9.3. Добавить в множество P_ε параллелопады $[p_1]_i \times \dots \times [mid([p_k]_i); \overline{p_k}]_i \times \dots \times [p_n]_i$ и $[p_1]_i \times \dots \times [p_k]_i \times [mid([p_k]_i); \overline{p_k}]_i \times \dots \times [p_n]_i$, где k – номер компоненты параллелопада $[p]_i$ с наибольшей шириной, удалить $[p]_i$ из множества P_ε .

Шаг 9.4. Увеличить i на единицу. Если $i \leq |P_\varepsilon|$, где $|\cdot|$ – мощность множества, перейти к шагу 9.2. Иначе перейти к шагу 9.5.

Шаг 9.5. Для каждого $[p]_i \in P_\varepsilon$ найти $[f]([p]_i)$ и выбрать $[p]^* = \arg \min_{[p]_i \in P_\varepsilon} [f]([p]_i)$.

6. Алгоритмы операторов проверки

На вход всех операторов проверки подается множество параллелопадов X , проверяемый интервал $[l]$ и параметр ширины w .

Алгоритм OI - оператора:

Шаг 1. Для каждого параллелопада из множества X применить инвертор для интервала $[l]$ с параметром $\varepsilon = w$, т.е. будет выработано множество множеств параллелопадов $\{P_i\}_{i=1}^{|X|}$.

Шаг 2. Если $\exists P_i \neq \emptyset$, то проверка выполнена удачно. В противном случае – неудачно.

Алгоритм OIR - оператора:

Шаг 1. Для каждого параллелопада $[x]_i$ из множества X применить инвертор для интервала $[l]$ с параметром $\varepsilon = w$. Таким образом, будет выработано множество множеств параллелопадов $\{P_i\}_{i=1}^{|X|}$.

Шаг 2. Задать множество $\tilde{X} = \bigcup_{i=1}^{|\mathbf{X}|} P_i$.

Шаг 3. Если $\tilde{X} = \emptyset$, то проверка выполнена неудачно. В противном случае проверка завершена удачно, заменить X на \tilde{X} .

Алгоритм FTR - оператора:

Шаг 1. Положить $i = 1$. Задать множество $\tilde{X} = X$.

Шаг 2. Проверить выполнение следующих условий (сам параллелотоп $[\mathbf{x}]_i$ после проверки удаляется из множества X):

- если $[f]([\mathbf{x}]_i) \cap [l] = \emptyset$, то продолжить работу;

- если $[f]([\mathbf{x}]_i) \subseteq [l]$ или $[f]([\mathbf{x}]_i) \cap [l] \neq \emptyset$, $\omega([\mathbf{x}]_i) < \varepsilon$, то проверка завершена удачно;

- в случае невыполнения условий 1 и 2 задать параллелотопы $[\mathbf{x}]' = [x_1]_i \times \dots \times [mid([x_k]_i); \overline{[x_k]_i}] \times \dots \times [x_n]_i$ и $[\mathbf{x}]'' = [x_1]_i \times \dots \times [\underline{[x_k]_i}; mid([x_k]_i)] \times \dots \times [x_n]_i$, где k – номер компоненты параллелотопа $[\mathbf{x}]_i$ с наибольшей шириной и добавить их в множество X .

Шаг 3. Если $i < |\mathbf{X}|$, то увеличить i на единицу и перейти к шагу 2, если $i = |\mathbf{X}|$ – перейти к шагу 4.

Шаг 4. Заменить X на \tilde{X} , генерировать сообщение о том, что проверка завершена неудачно.

7. Алгоритмы операторов сжатия

На вход всех операторов сжатия подается параллелотоп $[\mathbf{s}]$.

Алгоритм SAS - оператора (для работы необходим параметр ширины w):

Шаг 1. Задать $[\tilde{Y}] = \emptyset$.

Шаг 2. Для каждой из компонент вектора $[s_i]$ найти наименьшее целое число n_i , такое что $\frac{\omega([s_i])}{n_i} \leq w$.

Шаг 3. Каждый из интервалов $[s_i]$ представить в виде объединения непересекающихся интервалов $\bigcup_{j=1}^{n_i} [s_i^j]$. Таким образом, создается разбиение изначального параллелотопа $[\mathbf{s}] = \bigcup_{k=1}^N [\mathbf{s}]_k$,

где $N = \prod_{i=1}^n n_i$.

Шаг 4. Найти $[\tilde{Y}] = \bigcup_{k=1}^N [f]([\mathbf{s}]_k)$ – результирующий сжатый интервал.

Алгоритм RPS - оператора (для работы необходимо число точек A):

Шаг 1. Создать A случайных точек $\xi_i \in [s_i], i = 1, \dots, A$.

Шаг 2. Найти $[\tilde{Y}] = \left[[f]([\mathbf{s}]), \min_i f(\xi_i) \right]$ – результирующий сжатый интервал.

8. Оценка качества решения

Сформированные теоремы и доказательства основываются на конфигурации обобщенного инверсного метода с оператором проверки ОI при отсутствии оператора сжатия. При соответствующей замене модулей в алгоритме теоремы и доказательства аналогичны.

Теорема 1. Обобщенный инверсный метод позволяет найти решение задачи интервальной ε -минимизации.

Доказательство. Вследствие того, что на шаге выбора (шаг 9) параллелотоп выбирается из множества P_ε , его ширина никогда не превысит выбранного значения ε (из-за способа построения множества P_ε). На каждом цикле алгоритм изначально с помощью инвертора вырабатывает множество параллелотопов P , в котором для любого параллелотопа $[p] \in P$ выполнено $[p] \subseteq [s]$, $\omega([p]) \geq \varepsilon$ и $[f]([p]) \subseteq [\gamma_l]$ или $[p] \subseteq [s]$, $\omega([p]) < \varepsilon$ и $[f]([p]) \cap [\gamma_l] \neq \emptyset$, поэтому переход к анализу $[\gamma_u]$ свидетельствует о том, что множество P пусто, т.е. $\forall [x] \subseteq [s]$, $\omega([x]) < \varepsilon$ выполнено $[f]([x]) \subset [\gamma_u]$. Таким образом, в ходе каждого цикла алгоритма однозначно определяется интервал $[y]$, такой что $\nexists [x] \subseteq [s]$, для которого выполнено $\overline{[f]([x])} < [y]$. Это исключает возможность существования такого параллелотопа $[x] \subseteq [s]$, $\omega([x]) \geq \varepsilon$, для которого $\overline{[f]([x])} < [f]([p]^*)$.

Теорема 2. Пусть $x^* = \arg \min_{x \in [s]} f(x)$, $[p]^*$ решение задачи интервальной ε -минимизации обобщенным инверсным методом, тогда $f(x^*) \in [f]([p]^*)$.

Доказательство. Необходимо показать, что для параллелотопа $[p]^*$ одновременно выполняются два неравенства: $f(x^*) \geq [f]([p]^*)$ и $f(x^*) \leq \overline{[f]([p]^*)}$. Докажем выполнение первого неравенства. Пусть $f(x^*) < [f]([p]^*)$, тогда $\exists [\tilde{p}]$ решение интервальной задачи интервальной ε -минимизации, такое что $f(x^*) \in [f]([\tilde{p}])$. Тогда получаем: $[f]([\tilde{p}]) < [f]([p]^*)$, что невозможно, так как $[p]^*$ также является решением интервальной задачи интервальной ε -минимизации и при этом $[p]^* = \arg \min_{[p] \in P_\varepsilon} [f]([p])$. Таким образом, первое неравенство выполнено.

Докажем выполнение второго неравенства. Пусть $f(x^*) > \overline{[f]([p]^*)}$, тогда $[f]([p]^*) = \emptyset$, что невозможно, так как $[p]^* \neq \emptyset$. Таким образом, второе неравенство выполнено. Так как оба неравенства выполнены, теорема доказана.

Следствие из теоремы 2. В параллелотопе $[p]^*$ есть точка \tilde{x} , такая что $f(\tilde{x}) - f(x^*) \leq \omega([f]([p]^*))$.

9. Программное обеспечение. Решение тестовых примеров

На основе изложенного алгоритма написана программа поиска условного глобального минимума обобщенным инверсным методом. Среда разработки – Microsoft Visual Studio, язык программирования – С#.

Алгоритм протестирован при минимизации следующих функций:

1. First function of De Jong: $f(x, y) = x^2 + y^2$, область поиска $[-500; 500] \times [-500; 500]$.
2. Rastrigin's function: $f(x, y) = 20 + (x^2 - 10 \cdot \cos(2 \cdot \pi \cdot x)) + (y^2 - 10 \cdot \cos(2 \cdot \pi \cdot y))$, область поиска $[-5.12; 5.12] \times [-5.12; 5.12]$.
3. Schwefel's function: $f(x, y) = -x \cdot \sin(\sqrt{|x|}) - y \cdot \sin(\sqrt{|y|})$, область поиска $[-500; 500] \times [-500; 500]$.
4. Easom's function: $f(x, y) = -\cos(x) \cdot \cos(y) \cdot \exp(-(x - \pi)^2 - (y - \pi)^2)$, область поиска $[-100; 100] \times [-100; 100]$.
5. Ackley's function: $f(x, y) = -20 \cdot \exp\left(-0.2 \cdot \sqrt{\frac{x^2 + y^2}{2}}\right) - \exp\left(\frac{\cos(2 \cdot \pi \cdot x) + \cos(2 \cdot \pi \cdot y)}{2}\right) + 20 + e$, область поиска $[-32.768; 32.768] \times [-32.768; 32.768]$.
6. Beal's function: $f(x, y) = (1.5 - x \cdot (1 - y))^2 + (2.25 - x \cdot (1 - y^2))^2 + (2.625 - x \cdot (1 - y^3))^2$, область поиска $[-4.5; 4.5] \times [-4.5; 4.5]$.
7. Rosenbrock's function: $f(x, y) = (1 - x)^2 + 100 \cdot (y - x^2)^2$, область поиска $[-5; 10] \times [-5; 10]$.

Таблица 1

Результаты работы

Функция/время работы	Найденный параллелотоп	Значение функции на параллелотопе	Точное решение	Минимальное значение
1/0,121 с	$[0; 0.01] \times [0; 0.01]$	$[0; 0.0002]$	$(0, 0)^T$	0
2/0,203 с	$[0; 0.01] \times [0; 0.01]$	$[0; 0.03967]$	$(0, 0)^T$	0
3/27,702 с	$[420.967; 420.974] \times [420.967; 420.974]$	$[-837.9797; -837.9518]$	$(420.9687, 420.9687)^T$	-837.9658
4/0,022 с	$[3.137; 3.143] \times [3.137; 3.143]$	$[-1; -0.9999]$	$(\pi, \pi)^T$	-1
5/0,112 с	$[-0.008; 0] \times [-0.008; 0]$	$[0; 0.0354]$	$(0, 0)^T$	0
6/0,991 с	$[2.953; 2.961] \times [0.483; 0.492]$	$[0; 0.0019]$	$(3, 0.5)^T$	0
7/0,814 с	$[0.998; 1.005] \times [0.991; 0.998]$	$[0; 0.0362]$	$(1, 1)^T$	0

Параметры алгоритма: $\varepsilon = \zeta = 0.01$, оператор проверки – ОI с параметром $w = 0.01$, оператор сжатия – SAS с параметром $w = 50$.

Заключение

Полученные результаты (табл. 1) свидетельствуют об эффективности разработанного алгоритма и иллюстрируют приведенные теоремы. Модульная структура обобщенного инверсного алгоритма позволяет проводить его тонкую и гибкую настройку в соответствии с имеющимися вычислительными ресурсами и особенностями задачи.

ЛИТЕРАТУРА

1. Пантелеев А.В., Летова Т.А. Методы оптимизации: практический курс. - М.: Логос, 2011.
2. Пантелеев А.В., Метлицкая Д.В., Алешина Е.А. Методы глобальной оптимизации // Метаэвристические стратегии и алгоритмы. - М.: Вузовская книга, 2013.
3. Пантелеев А.В. Применение эволюционных методов глобальной оптимизации в задачах оптимального управления детерминированными системами. - М.: МАИ, 2013.
4. Jaulin L., Kieffer M., Didrit O., Walter E. Applied interval analysis. - London: Springer-Verlag, 2001.
5. Moore R.E. Interval analysis. - Englewood Cliffs, Prentice Hall, 1966.
6. Шарый С.П. Конечномерный интервальный анализ. - Новосибирск: XYZ, 2010.
7. Ratschek H., Rokne J. New computer methods for global optimization. - Chichester: Horwood, 1988.
8. Shary S.P. Randomized algorithms in interval global optimization // Numerical Analysis and Applications, V. 1, 2008. - P. 376-389.
9. Shary S.P. A surprising approach in interval global optimization // Reliable Computing, 2001. - P. 497-505.
10. Hansen E. Global optimization using interval analysis. - New York: Marcel Dekker, 2004.
11. Moore R.E. Methods and applications of interval analysis. - Philadelphia: SIAM, 1979.
12. Пановский В.Н. Применение аппарата интервального анализа для поиска глобального экстремума функций // Труды МАИ. - 2012. - № 51. [Электронный ресурс]. URL: <http://www.mai.ru/science/trudy/published.php?ID=28948>.
13. Пановский В.Н. Интервальные алгоритмы нахождения оптимального программного управления детерминированными системами // Молодежь и будущее авиации и космонавтики: конкурс научно-технических работ и проектов (Москва, 30 октября 2012 г.). - М., 2012. - С. 155.

GENERALIZED INVERSE INTERVAL METHOD OF GLOBAL CONSTRAINED OPTIMIZATION

Panteleyev A.V., Panovskiy V.N.

The algorithmic and program software of inverse interval method for global constrained optimization are considered. The solution of model examples and the proof of the theorems of the algorithm's convergence are presented. The generalized scheme of developed algorithms has been created. This scheme has two replaceable modules of compression and check. This module approach allows the users to implement their own versions of the algorithm without loss of the method convergence. This will help to tune the method according to the characteristics of the current problem.

Key words: interval analysis, global constrained optimization, convergence, inverse interval method, software.

Сведения об авторах

Пантелеев Андрей Владимирович, 1955 г.р., окончил МГТУ им. Н.Э. Баумана (1978), профессор, доктор физико-математических наук, заведующий кафедрой математической кибернетики МАИ (Национального исследовательского университета), автор более 170 научных работ, область научных интересов – методы синтеза оптимальных нелинейных систем управления, методы оптимизации.

Пановский Валентин Николаевич, 1992 г.р., окончил МАИ (2013), аспирант МАИ, автор 9 научных работ, область научных интересов – методы оптимизации, численные методы, эвристические алгоритмы.