

СИБИРСКИЙ ЖУРНАЛ ИНДУСТРИАЛЬНОЙ МАТЕМАТИКИ

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Главный редактор	В. Л. Береснев
Зам. главного редактора	А. Л. Карчевский
Отв. секретарь	В. А. Дедок

ЧЛЕНЫ РЕДКОЛЛЕГИИ

Г. В. Алексеев	С. Б. Медведев
Б. Д. Аннин	Р. Г. Новиков
В. С. Белоносов	Д. Е. Пальчунов
В. Н. Белых	П. И. Плотников
Ю. С. Волков	В. Г. Романов
В. П. Ильин	Е. М. Рудой
С. И. Кабанихин	В. М. Садовский
А. Н. Карапетянц	Д. И. Свириденко
М. В. Клибанов	А. С. Терсенов
С. С. Кутателадзе	В. С. Тимофеев
В. А. Левин	В. В. Шайдуров
Н. И. Макаренко	

Журнал основан в 1998 году
Выходит 4 раза в год
Том 25, №4(92)
Октябрь - декабрь 2022 г.

СОДЕРЖАНИЕ

Аюпова Н. Б., Голубятников В. П., Минущкина Л. С. Об инвариантных поверхностях в фазовых портретах моделей кольцевых генных сетей	5
Боронина М. А., Куликов И. М., Черных И. Г., Винс Д. В. Использование комбинации схем Рунге и Рунге-Кутты для численного решения уравнений магнитной гидродинамики в задачах космической плазмы	14
Васильев В. И., Кардашевский А. М., Попов В. В. Итерационное решение ретроспективной обратной задачи теплопроводности с неоднородными граничными условиями Дирихле	27
Демиденко Г. В. Метод решения одной биологической задачи большой размерности	42
Дудко О. В., Лаптева А. А., Рагозина В. Е. Эволюция волновой картины кусочно-линейного одноосного растяжения и сжатия разномодульного упругого стержня	54
Ермишина В. Е. Гиперболическая модель сильнонелинейных волн в двухслойных течениях неоднородной жидкости	71
Мишин А. В. Учёт обобщённой производной и коллективного влияния фаз на процесс гомогенизации	86
Нещадим М. В. Уравнение Лиувилля и точно транзитивные представления алгебры $sl_2(\mathbb{R})$..	99
Остросаблин Н. И. Единственность решения граничных задач статических уравнений теории упругости с несимметричной матрицей модулей упругости	107
Паламарчук Е. С. Оптимальное управление в задаче долгосрочного трекинга экспоненциального процесса Орнштейна — Уленбека	116
Перцев Н. В., Бочаров Г. А., Логинов К. К. Численное моделирование динамики популяции Т-лимфоцитов в лимфоузле	136
Петраков И. Е. Контактная задача изгиба многослойной композитной пластины с учётом различных модулей упругости при растяжении и сжатии	153
• Селиверстов Е. Ю. Иерархический метод установки параметров параллельных популяционных метаэвристических алгоритмов оптимизации	164
Сказка В. В. Кососимметрические разностные аналоги четвёртого порядка аппроксимации первой производной	179
Скворцова М. А., Ыскак Т. Оценки решений дифференциальных уравнений с распределённым запаздыванием, описывающих конкуренцию нескольких видов микроорганизмов	193
Терсенов Ар. С. О существовании вязких решений анизотропных параболических уравнений с переменными показателями анизотропности	206
Чумаков Г. А., Чумакова Н. А. О локализации неустойчивого решения одной системы трёх нелинейных обыкновенных дифференциальных уравнений с малым параметром	221

НОВОСИБИРСК

ИЗДАТЕЛЬСТВО ИНСТИТУТА МАТЕМАТИКИ

УДК 519.6

ИЕРАРХИЧЕСКИЙ МЕТОД УСТАНОВКИ ПАРАМЕТРОВ ПАРАЛЛЕЛЬНЫХ ПОПУЛЯЦИОННЫХ МЕТАЭВРИСТИЧЕСКИХ АЛГОРИТМОВ ОПТИМИЗАЦИИ

© 2022 Е. Ю. Селиверстов

*Московский государственный технический университет им. Н. Э. Баумана,
ул. 2-я Бауманская, 5/1, г. Москва 105005, Россия*

E-mail: evgeny.seliverstov@omniverse.ru

Поступила в редакцию 21.03.2022 г.; после доработки 26.07.2022 г.;
принята к публикации 29.09.2022 г.

Основной сложностью в применении метаэвристических алгоритмов решения задачи глобальной параметрической оптимизации (М-алгоритмов) является наличие свободных параметров, существенно влияющих на показатели сходимости и эффективности оптимизации. Задачу отыскания оптимальных значений свободных параметров называют задачей установки, среди методов решения которой различают статические методы настройки, выполняемые до алгоритма оптимизации, и динамические методы адаптации параметров, выполняемые совместно с ним. Предлагается иерархический метод установки параметров для класса популяционных М-алгоритмов. Отличительной особенностью метода является использование иерархической модели алгоритма оптимизации, где на нижнем уровне иерархии представлен алгоритм из класса последовательных М-алгоритмов, а на верхнем — параллельный алгоритм с островной моделью параллелизма. Установка параметров производится иерархическим методом, сочетающим метод настройки параметров последовательного алгоритма оптимизации для нижнего уровня иерархии модели и метод адаптации параметров параллельного алгоритма для верхнего уровня. В методе применяется векторный критерий оптимальности задачи установки, включающий метрики скорости сходимости и достигаемых экстремумов алгоритма оптимизации. Рассматривается подход для оценки скорости сходимости многошагового метода оптимизации. Представлены результаты исследования эффективности предлагаемого метода на тестовых задачах оптимизации из пакета СЕС.

Ключевые слова: методы глобальной оптимизации, метаэвристические алгоритмы, установка параметров, адаптация параметров.

DOI: 10.33048/SIBJIM.2022.25.413

ВВЕДЕНИЕ

Алгоритмы решения задачи глобальной оптимизации разделяют на точные, гарантирующие нахождение оптимального решения с заданной точностью, и приближённые, среди которых выделяют активно развивающийся класс метаэвристических алгоритмов глобальной оптимизации (далее М-алгоритмов). М-алгоритмы представляют собой итерационные алгоритмы, основанные на сочетании и управлении подчинёнными эвристиками для направленного исследования пространства поиска. Под эвристиками понимают алгоритмы получения приближённого решения заданной задачи оптимизации за меньшее время, чем при решении точным алгоритмом. М-алгоритмы способны отыскать решение, близкое к оптимальному, без априорных знаний о свойствах целевой функции и пространства поиска и поэтому применимы для решения широкого класса задач оптимизации высокой размерности и вычислительной сложности.

Ограничим класс МА рассматриваемых М-алгоритмов итерационными популяционными алгоритмами (population-based metaheuristics), где каждый агент популяции представляет отдельное значение вектора варьируемых параметров задачи оптимизации, изменяемое в рамках итераций на основании характеристик как отдельного агента, так и всей популяции. В классе МА выделяют подклассы эволюционных алгоритмов (эволюционная стратегия, дифференциальная эволюция) [1], поведенческих алгоритмов (алгоритм роя частиц, муравьиной колонии) [2, 3] и ряд других [4].

Важным направлением развития М-алгоритмов является применение распараллеливания для сокращения времени решения задачи оптимизации. Распространён подход к построению параллельных М-алгоритмов на основе островной модели параллелизма [5–7].

М-алгоритмы имеют свободные параметры, влияющие на скорость сходимости и эффективность оптимизации. Они существенно различаются между отдельными алгоритмами [1]. Задача оптимизации стратегии М-алгоритма с целью поиска его оптимальной стратегии (значения вектора свободных параметров) называется задачей установки параметров (parameter setting) [8]. Методы установки параметров разделяют на методы настройки параметров (parameter tuning) [9, 10], использующие изменение параметров до выполнения алгоритма, и методы адаптации параметров (также называемые методами управления параметрами, parameter control) [11, 12], использующие изменение параметров во время выполнения алгоритма. При использовании методов адаптации к завершению процесса оптимизации получаем стратегию, оптимальную для данной целевой функции, тогда как для методов настройки параметров априорно заданные параметры в общем случае не являются оптимальными для данной функции.

В данной работе рассматривается метод установки параметров для параллельных М-алгоритмов, основанный на иерархической модели М-алгоритма из последовательного и параллельного уровней, что позволяет совмещать различные типы методов установки параметров — настройку параметров последовательного алгоритма и адаптацию параметров параллельного алгоритма.

1. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ АЛГОРИТМА

Определим базовую задачу оптимизации следующим образом:

$$\min_{X \in D_X} \Phi(X) = \Phi(X^*), \quad (1)$$

где X — вектор варьируемых параметров базовой задачи; $\Phi(X)$ — целевая функция задачи; X^* — оптимальное значение вектора X ; D_X — область допустимых значений вектора X .

В качестве математической модели $M(P)$ параллельного М-алгоритма A используем иерархическую модель, состоящую из двух уровней: верхний уровень — модель $M_p(P_p)$ параллельного алгоритма A , основанная на островной модели параллелизма; нижний уровень — модель $M_s(P_s)$ последовательного алгоритма $A_s \in \text{МА}$, формализующая логику потока управления последовательного М-алгоритма.

Модель $M(P)$ полагаем параметризованной вектором из непрерывных параметров $P = \{P_i \in \mathbb{R}\}$, который разделяем на вектор параметров P_s модели нижнего уровня (последовательного алгоритма) и вектор параметров P_p модели верхнего уровня (параллельного алгоритма). Назовём значение вектора $P = (P_s, P_p)$ стратегией алгоритма A . Назовём множество D_P областью допустимых значений вектора параметров P .

Островная модель параллелизма основана на разделении общей популяции на непересекающиеся острова (субпопуляции), для которых осуществляется независимая друг от друга оптимизация и периодическая межостровная коммуникация элементами популяции. Рассматриваем N_S островов, $I = \{I_i, i \in [0, N_S)\}$.

Итерации параллельного алгоритма с островной моделью параллелизма выполняются по следующей схеме.

1. Инициализируем номер итерации $t = 0$ и вектор варьируемых параметров $X_0 = X^0$.
2. Выполняем операцию инициализации острова $I_i = S_I(i)$.
3. Выполняем операцию оптимизации для каждого острова I_i на протяжении N_T итераций и получаем значение вектора варьируемых параметров $X^{*,i,t}$.
4. Выполняем операции коммуникации $C(i, j)$ между островами $I_i, I_j, i \neq j, i, j \in [0, N_S)$.
5. Если условие завершения алгоритма не выполнено, присваиваем $t = t + 1$ и переходим к шагу 3.

Итоговое оптимальное значение X^* вектора варьируемых параметров вычисляем как $X^* = \operatorname{argmin}_{X \in \{X^{*,i,t}, i \in [0, N_S)\}} \Phi(X)$.

Вид операций инициализации $S(i)$ и коммуникации $C(i, j)$ зависит от выбранного М-алгоритма, $A_s \in \text{МА}$. Параметры $\{N_S, N_T\} \subset P_p$ назовём стандартными параллельными параметрами, не зависящими от алгоритма A_s .

Отметим, что параметры P модели M влияют как на характеристики последовательного М-алгоритма оптимизации, так и на характеристики параллельного алгоритма (например, на число островов N_S островной модели параллелизма). Предметом изучения данной работы являются методы установки параметров P модели M , тогда как аспекты распараллеливания алгоритма и оценки параллельного ускорения не рассматриваются и являются предметом дальнейшего изучения. Графовое представление иерархической модели параллельного алгоритма для планирования выполнения на параллельной вычислительной системе рассмотрено в работе автора [13].

2. КРИТЕРИЙ ОПТИМАЛЬНОСТИ ЗАДАЧИ УСТАНОВКИ ПАРАМЕТРОВ

Введём векторный критерий оптимальности $\Psi(P, t)$, состоящий из компонентов:

$$\Psi(P, t) = (r_A(P, t), \Phi_A(P, t)), \quad (2)$$

где $r_A(P, t)$ — скорость сходимости параллельного алгоритма $A(P)$ на итерации t ; $\Phi_A(P, t)$ — достигнутое значение экстремума целевой функции М-алгоритма $A_s(P)$ на итерации t .

Вычисляем критерий $\Phi_A(P)$ на основании значений целевой функции $\Phi(X)$, полученных оптимизацией вектора X алгоритмом A_s на итерации t на острове i :

$$\Phi_A(P, t) = \min_{X \in \{X^{*,i,t}, i \in [0, N_S)\}} \Phi(X).$$

Обозначим как $\widehat{\Psi}^t = (\Psi_0, \Psi_1, \dots, \Psi_t)$ последовательность значений критериев оптимальности на шагах $[0, t]$. Введём глобальный номер итерации t^G , вычисляемый по номеру итерации t адаптивного алгоритма МО_Р и длительностям сезонов N_T при помощи функции

$$F_G(t) = \sum_{i=0}^t N_{T,i}. \quad (3)$$

Рассмотрим способ вычисления критерия скорости сходимости $r_A(P, t)$ на итерации t . На каждом шаге t алгоритма оптимизации $A(P)$ получаем значение достигнутого экстремума $\Phi_A(P, t)$ целевой функции М-алгоритма. Функция $\Phi(t)$ не является непрерывно дифференцируемой, поэтому применим аппроксимацию функции $\Phi(t)$ дважды дифференцируемым полиномом

$$f(x, a) = \sum_{j=1}^n a_j x^{j-1}. \quad (4)$$

Задаём вектор точек $x = (x_1 \dots x_i \dots x_m)$, где $x_i = F_G(t)(t + m - i)$, $i \in [1, m]$, и вектор $y = (f(x_1, a) \dots f(x_i, a) \dots f(x_m, a))$ значений функции Φ в этих точках, где $f(x_i, a) = \Phi_i$, $i \in [1, m]$. Запишем уравнения (4) для m точек в векторной форме $Qa = y$. Матрица Q размерностью $m \times n$ принимает вид

$$Q = \begin{bmatrix} 1 & x_1 & \dots & x_1^{n-1} \\ 1 & x_2 & \dots & x_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \dots & x_m^{n-1} \end{bmatrix}.$$

Решим задачу поиска вектора a , определяющего полином $f(x)$ в точках x, y , методом наименьших квадратов для линейной регрессии [14]. Минимизируем функцию невязок:

$$S(a) = \sum_{i=1}^m e_i^2(x, a),$$

где $e_i(x) = y_i - f(x_i, a)$ — невязка, $i \in [1, m]$. Тогда $S(a) = \sum_{i=1}^m (y_i - f(x_i, a))^2$.

Найдём частные производные функции $S(a)$ по a :

$$\frac{\partial S_i(x)}{\partial a_j} = \sum_{i=1}^m \left(-2(y_i - f(x_i, a)) \frac{df(x_i, a)}{da_j} \right).$$

Определим матрицу Якоби J из этих частных производных: $J_{ij} = x_i^{j-1}$. Видно, что для функции $f(x, a)$ матрица Q совпадает с матрицей J . Тогда

$$\frac{\partial S_i(x)}{\partial a_j} = \sum_{i=1}^m \left(-2(y_i - \sum_{k=1}^n (J_{ik} a_k)) J_{ij} \right).$$

Для нахождения оптимального значения a^* , доставляющего минимум функции S , потребуем $\frac{\partial S_i(x)}{\partial a_j} = 0$. Получаем уравнение $\sum_{i=1}^m y_i J_{ij} = \sum_{i=1}^m \sum_{k=1}^n J_{ik} J_{ij} a_k^*$ или в векторной форме

$$J^T y = J^T J a^*. \quad (5)$$

Для аппроксимации выбираем $m = 5$ точек и кубический полином (4) (значение $n = 4$). Для решения системы линейных уравнений (5) применяем численный метод на основе QR-разложения матрицы $J^T J$.

Определим значение второй производной в точке x_m : $\frac{\partial^2 f(x, a^*)}{\partial x^2}(x_m) = 6a_4^* x_m + 2a_3^*$.

Так как значения номеров шагов t могут быть достаточно большими, при аппроксимации применяем сдвиг всех значений вектора x на величину $-x_0$. Значение $\frac{\partial^2 f(x, a^*)}{\partial x^2}(x_m)$ при этом не изменяется.

Ограничиваем значение r_A диапазоном $[r^{\min}; r^{\max}]$, что позволяет задать чувствительность метода адаптации к изменению скорости сходимости, а также вычислять значение r_A при плохо обусловленной системе (5). В таком случае определяем скорость сходимости как соответствующую границу диапазона. Таким образом, скорость сходимости на итерации t определяем как

$$r_A(t) = \begin{cases} \max(\min(6a_4^* x_m + 2a_3^*, r^{\min}), r^{\max}), & \text{если } t \geq 5; \\ 1, & \text{иначе.} \end{cases}$$

Значение $r_A(t) < 1$ означает увеличение скорости сходимости на текущем шаге t алгоритма относительно предыдущего шага (ускорение оптимизации), а значение $r_A(t) > 1$ — уменьшение скорости сходимости (стагнацию оптимизации). На ранних итерациях $t < m$ скорость сходимости принимаем равной единице, так как для аппроксимации не накоплено достаточно данных.

3. МЕТОД УСТАНОВКИ ПАРАМЕТРОВ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА

Предлагаем иерархический метод установки параметров параллельного М-алгоритма оптимизации A (метод МО).

Оптимальной динамической стратегией алгоритма A называем последовательность стратегий (P_0, \dots, P_t) на шагах $[0, t]$. Результатами метода МО являются оптимальная динамическая стратегия, оптимальное значение X^* вектора варьируемых параметров и целевой функции $\Phi(X^*)$ базовой задачи (1).

Подчеркнём, что целью установки параметров является минимизация векторного критерия оптимальности Ψ в пространстве параметров стратегии P , а не в пространстве варьируемых параметров X алгоритма Φ . В результате выполнения одной итерации метода МО изменяются значения векторов параметров P, X . В методе варьируем только вектор параметров P , тогда как вектор X меняет значение в результате выполнения М-алгоритма, а значение критерия $\Psi(P, t)$ неявно зависит от значения вектора X .

Предлагаем иерархический метод МО как гибридный метод, состоящий из метода МО_S настройки параметров P_s и метода МО_P адаптации параметров P_p . Метод согласования МО проектируем как гибридный метод по схеме препроцессор/постпроцессор, где МО_S является препроцессором, а метод МО_P — постпроцессором.

Параметризованный алгоритм нижнего уровня $A_s(P_s)$ не зависит от параметров верхнего уровня P_p . В то же время изменение стратегии P_s приводит к изменению алгоритма $A_s(P_s)$. Таким образом, выполнение алгоритма A_s может осуществляться независимо от алгоритма оптимизации параметров верхнего уровня P_p . Поэтому в качестве метода установки параметров нижнего уровня выбираем метод настройки. Для установки параметров P_p используем значения критериев $r_A(P, t), \Phi_A(P, t)$ с каждого шага t алгоритма A , а следовательно, необходимо применение метода адаптации параметров.

Рассматриваем метод МО_S настройки вектора параметров P_s последовательного алгоритма $A_s \in \mathbb{M}\mathbb{A}$. Предлагаем метод МО_S решения этой задачи на основе однокритериального метода оптимизации роем частиц $\min_{P_s \in D_P} \Phi_S(P_s) = \Phi_S(P_s^*)$.

Вектором варьируемых параметров метода МО_S является P_s , а целевой функцией — функция $\Phi_S(P_s) = \sum_i k_{S,i} \Psi_i(P_s, 0), i \in [1, |\Psi|]$, полученная свёрткой векторного критерия оптимальности Ψ_S с априорно заданными коэффициентами $k_{S,i}$. Полученное оптимальное значение вектора параметров P_s^* считаем оптимальной стратегией для последовательного М-алгоритма A_s .

Предлагаем метод МО_P адаптации вектора параметров P_p параллельного алгоритма A . В основу метода МО_P положим процесс адаптации параметров P_p во время выполнения самого алгоритма $A(P)$. Шаг метода МО_P совпадает с итерацией параллельного алгоритма A . На каждом шаге t метода МО рассматриваем фиксированный алгоритм $A(P)$, определённый набором параметров P . Одна итерация метода МО_P полностью включает выполнение алгоритма A_s , а окончание работы метода МО_P совпадает с окончанием работы алгоритма A .

Разобьём итерацию метода МО_P на два шага: выполнение оптимизации алгоритмом A_s на всех островах и решение задачи адаптации. Так как метод МО_P является методом адаптации параметров, необходимо задать вектор-функцию F_P перехода к следующему значению вектора параметров P_p . Эту функцию определим исходя из номера шага t метода МО_P и значения

векторного критерия оптимальности $\Psi(P, t)$ адаптируемого алгоритма A . Получаем формулировку задачи адаптации:

$$P^{t+1} = F_P(P^t, t, \hat{\Psi}^t). \quad (6)$$

Поскольку функции адаптации для различных параметров в составе вектора P_p различны, представим вектор-функцию F_P как совокупность функций $F_P = \{F_{P,(i)}\}, i \in [1, |P_p|]$.

Предлагаем функции адаптации $F_{P,(i)}(t) = B(t) + \sum_{k=0}^{N_k-1} R(k, t)$ параметра $p = P_i$ на шаге t , отличающиеся построением из двух видов компонентов:

1) монотонный компонент $B(t)$, определённый диапазоном допустимых значений параметра p и шагом t ;

2) реактивные компоненты $R(k, t)$, обеспечивающие локальную реакцию на изменение $\Psi(P)$ на шаге τ_k .

В зависимости от характера параметра p выделяем два вида монотонных компонентов: убывающие B_D и возрастающие B_I .

Убывающий компонент B_D запишем в виде

$$B_D(t) = p^{\min} + \frac{U_0 e^{-C_2(F_G(t)-U_1)}}{1 + e^{-C_2(F_G(t)-U_1)}}. \quad (7)$$

Задаём граничные условия на значение компонента B_D и его первую производную $\frac{dB_D}{dt}$:

$$B_D(0) = p^{\max}, \quad \frac{dB_D(t)}{dt}(0) = -C_3.$$

Исходя из этих условий определяем значения промежуточных констант:

$$U_0 = (p^{\max} - p^{\min}) \left(1 + \frac{C_3}{C_2(p^{\max} - p^{\min}) - C_3} \right),$$

$$U_1 = \frac{1}{C_2} \left(\ln(C_2(p^{\max} - p^{\min}) - C_3) - \ln(C_3) \right).$$

Возрастающий компонент B_I запишем в виде

$$B_I(t) = \frac{p^{\max}}{1 + e^{-C_2(F_G(t)-U_2)}}, \quad (8)$$

где значения промежуточной константы U_2 определяем исходя из граничного условия $B_I(0) = p^{\min}$. Получаем $U_2 = \frac{1}{C_2} \ln \left(\frac{p^{\max}}{p^{\min}} - 1 \right)$.

Таким образом, монотонные компоненты B_D, B_I полностью заданы допустимым диапазоном значений параметра $p \in [p^{\min}, p^{\max}]$, начальной скоростью C_3 изменения параметра и скоростью C_2 изменения параметра в зависимости от t .

Вид и количество реактивных компонентов определяем на основании значений векторного критерия оптимальности $\Psi(P, t)$. Назовём *узлами реакции* множество из номеров шагов $R_\tau = \{t \in [0, t^{\max}] \mid W_R(t, \hat{\Psi}^t)\}$, для которых выполняется булева функция реакции W_R над последовательностью $\hat{\Psi}^t$ значений векторного критерия оптимальности. При помощи функции реакции определяем замедление или ускорение процесса оптимизации, что влечёт изменение адаптируемого параметра p функцией адаптации F_P . В исследовании используем одношаговую функцию реакции $\bar{W}_R: \bar{W}_R(t, \hat{\Psi}^t) = W_{R,(1)}(t, \hat{\Psi}^t) \vee W_{R,(2)}(t, \hat{\Psi}^t)$. Здесь функция $W_{R,(1)}$ обеспечивает определение достижения максимума r^{\max} :

$$W_{R,(1)}(t, \hat{\Psi}^t) = (|r_A(P_{p,t}, t) - r_A(P_{p,t-1}, t-1)| > \varepsilon_r) \wedge ((r^{\max} - r_A(P_{p,t}, t)) \leq \varepsilon_r).$$

Функция $W_{R,(2)}(t, \hat{\Psi}^t)$ обеспечивает периодическую реакцию только на номер шага t , что позволяет избежать стагнации алгоритма при выполнении условия $r_A(P, t) \leq r^{\max}$ на протяжении большого числа итераций. Сформулируем функцию на основе пилообразного сигнала с периодом T_W :

$$W_{R,(2)}(t, \hat{\Psi}^t) = \left(\left(\frac{t}{T_W} - \left\lfloor \frac{t}{T_W} \right\rfloor \right) = 0 \right) \wedge (t > T_W).$$

Зададим число реактивных компонентов равным числу узловых точек $N_k = |R_\tau|$. Реактивный компонент $R(k, t)$ представим в виде монотонно убывающей неотрицательной функции: $R(k, t) = L_k H(t - \tau_k) e^{-C_4(F_G(t) - \tau_k)}$, где L_k — весовой коэффициент компонента k , C_4 — априорно задаваемый коэффициент затухания реактивного компонента, $H(x)$ — функция Хевисайда со значением $H(0) = 1$.

Потребуем, чтобы значение функции F не превышало в узловых точках определённого порога $C_1 p^{\max}$ при любом числе реактивных компонентов. Также ограничим F значением самого монотонного компонента $B(t)$ при $C_1 B(t) \geq p^{\max}$. Запишем эти требования в виде неравенств

$$B(\tau_i) + \sum_{k=0}^{N_k-1} R(k, \tau_i) \leq \max(C_1 p^{\max}, B(\tau_i)), \quad i \in [0, N_k).$$

Пусть в каждой узловой точке сумма значений реактивных компонентов будет максимальной. Тогда получаем систему из N_k линейных уравнений в узловых точках τ_i относительно весовых коэффициентов L_k :

$$\sum_{k=0}^{N_k-1} L_k H(\tau_i - \tau_k) e^{-C_4(F_G(\tau_i) - F_G(\tau_k))} = (C_1 p^{\max} - B(\tau_i)) H(C_1 p^{\max} - B(\tau_i)), \quad i \in [0, N_k). \quad (9)$$

Запишем систему в матричной форме: $A_R L = b_R$, где

$$L = \begin{bmatrix} L_0 \\ L_1 \\ \vdots \\ L_{N_k-1} \end{bmatrix}, \quad b_R = \begin{bmatrix} (C_1 p^{\max} - B(\tau_0)) H(C_1 p^{\max} - B(\tau_0)) \\ (C_1 p^{\max} - B(\tau_1)) H(C_1 p^{\max} - B(\tau_1)) \\ \vdots \\ (C_1 p^{\max} - B(\tau_{N_k-1})) H(C_1 p^{\max} - B(\tau_{N_k-1})) \end{bmatrix}.$$

В силу $H(\tau_i - \tau_k) = 0$ при $i < k$ матрица A_R является нижнетреугольной и принимает вид

$$A_R = \begin{bmatrix} 1 & 0 & \dots & 0 \\ H(\tau_1 - \tau_0) e^{-C_4(F_G(\tau_1) - F_G(\tau_0))} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H(\tau_{N_k-1} - \tau_0) e^{-C_4(F_G(\tau_{N_k-1}) - F_G(\tau_0))} & H(\tau_{N_k-1} - \tau_1) e^{-C_4(F_G(\tau_{N_k-1}) - F_G(\tau_1))} & \dots & 1 \end{bmatrix}.$$

Заметим, что $A_{R,ii} = 1$ и $A_{R,ij} \in [0; 1)$, $i \neq j$. Следовательно, матрица A_R является невырожденной и обладает свойством диагонального преобладания. Поэтому для системы (9) можно применить прямой метод решения, например метод Гаусса, являющийся вычислительно устойчивым для данной системы [14].

Решая систему (9), получаем значения весовых коэффициентов L_k . Тогда реактивные компоненты $R(k, t)$ полностью заданы узловыми точками τ_k , $k \in [0, N_k)$, допустимым диапазоном значений параметра $p \in [p^{\min}, p^{\max}]$, коэффициентом C_1 допустимого превышения максимального значения параметра p и коэффициентом C_4 затухания.

Вид функций $F_{P,i}$ в общем случае зависит от состава вектора параметров P_p , включающего как стандартные параллельные параметры N_S, N_T модели M_p , так и параметры, специфичные для различных алгоритмов A . В рамках исследования для стандартных параллельных

параметров $P_p = \{N_S, N_T\}$ предложим функции адаптации \tilde{F}_{N_S} и \tilde{F}_{N_T} . При синтезе функций адаптации необходимо учитывать баланс между увеличением значений этих параметров, ведущих к улучшению достигаемого экстремума, и уменьшением числа оценок целевой функции Φ . При этом число оценок не имеет прямого влияния на время выполнения алгоритма оптимизации при использовании параллельной вычислительной системы. Поэтому диапазон значений N_S , N_T и коэффициенты задаём априорно исходя из требований к параллельному алгоритму и вычислительной системе.

Функцию адаптации \tilde{F}_{N_S} параметра N_S построим на основе монотонного убывающего компонента B_D (7), обеспечивающего снижение числа островов N_S по мере увеличения номера шага t , поскольку большее число островов улучшает первоначальное исследование пространства D_X параметров [15]. Для функции \tilde{F}_{N_S} задаём диапазон значений $p^{\max} = N_S^0$, $p^{\min} = N_S^{\min}$.

Функция адаптации \tilde{F}_{N_T} , построенная на основе монотонного возрастающего компонента B_I (8), обеспечивает нелинейное увеличение продолжительности сезона N_T по мере увеличения номера шага t , что позволяет на ранних шагах метода МОР производить быстрое исследование пространства D_X каждым островом за меньшее число шагов N_T , а на более поздних шагах производить локализацию экстремума за большее число шагов N_T алгоритма A . Продолжительность сезона N_T ограничена пороговым значением N_T^{\max} . Тогда задаём диапазон значений $p^{\max} = N_T^{\max}$, $p^{\min} = N_T^0$.

В качестве условия завершения алгоритма АОР предлагаем два условия.

1. Условие достижения стагнации [16] на протяжении t_ε итераций стагнации:

$$|\Phi_A(P^{t+1}, T^{t+1}) - \Phi_A(P^t, T^t)| < \varepsilon^{\max}, \quad (10)$$

где ε^{\max} — точность стагнации. Обозначим номер последней итерации t как t^{\max} .

2. Условие достижения максимального номера итерации t^{\max} :

$$t < t^{\max}. \quad (11)$$

Свойства сходимости параллельного алгоритма глобальной оптимизации A напрямую зависят от свойств сходимости последовательного алгоритма оптимизации A_s . Можно показать, что достаточным условием глобальной сходимости A является глобальная сходимость алгоритма A_s . При выполнении алгоритма A сходимость и оценку скорости сходимости определяем по полученной к шагу t монотонной невозрастающей последовательности $\hat{\Psi}^t$.

Промежуточные коэффициенты C_1, C_2, C_3, C_4 выбираем априорно исходя из требований к алгоритму установки МОР. Применительно к функциям $\tilde{F}_{N_T}, \tilde{F}_{N_S}$ коэффициент C_4 определяет чувствительность функций адаптации к изменению критерия скорости сходимости r_A (при больших значениях $C_4 > 0$ функции адаптации слабо зависят от r_A), коэффициент C_1 — относительное максимальное значение реактивных компонентов ($C_1 \in [0, 1]$). Коэффициенты C_2 и C_3 задают скорость изменения параметров N_T, N_S в зависимости от номера шага.

На рис. 1 приведён график некоторой функции $\tilde{F}_{N_S}(N_S, t)$ адаптации параметра N_S и её монотонных и реактивных компонентов от глобального номера итерации t , а на рис. 2 — некоторой функции $\tilde{F}_{N_T}(N_T, t)$ адаптации параметра N_T . Узлами реакции выбрано множество $R_\tau = \{40, 50\}$. В вычислительном эксперименте и на рис. 1, 2 используем функцию адаптации $\tilde{F}_{N_S}(N_S, t)$ параметра N_S со значениями констант $C_1 = 0.8$, $C_2 = 0.05$, $C_3 = 0.01$, $C_4 = 0.01$, диапазоном значений параметра $N_S \in [4, 16]$ и функцию адаптации $\tilde{F}_{N_T}(N_T, t)$ для параметра N_T со значениями констант $C_1 = 0.4$, $C_2 = 0.005$, $C_3 = 0.01$, $C_4 = 0.01$, диапазоном значений $N_T \in [1, 10]$.

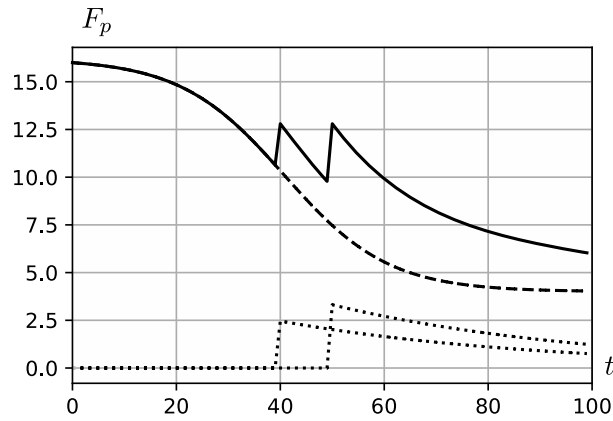


Рис. 1. Пример функции $\tilde{F}_{N_S}(N_S, t)$:
сплошная линия — $\tilde{F}_{N_S}(N_S, t)$; штриховая линия — $B_D(t)$; пунктирная линия — $R(k, t)$

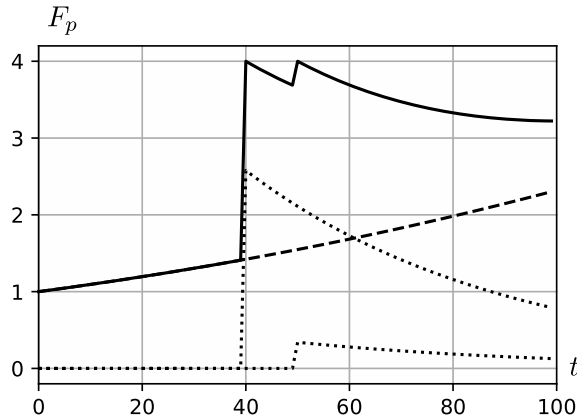


Рис. 2. Пример функции $\tilde{F}_{N_T}(N_T, t)$:
сплошная линия — $\tilde{F}_{N_T}(N_T, t)$; штриховая линия — $B_I(t)$; пунктирная линия — $R(k, t)$

4. ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

Проведём вычислительные эксперименты для сравнения адаптивного алгоритма оптимизации PAGOS, использующего предлагаемый в работе иерархический метод МО установки параметров, и классического алгоритма оптимизации CLASSIC, в котором не применяется управление параметрами, а используются их априорно заданные значения.

В качестве базовой задачи оптимизации применяем тестовые унимодальные и мультимодальные целевые функции из пакета CEC (Congress of Evolutionary Computing) [17]. В качестве последовательного М-алгоритма A_s для алгоритмов PAGOS и CLASSIC выбираем метаэвристический алгоритм PSO [2] (алгоритм оптимизации роем частиц), определённый параметрами $\chi = 0.7298$, $k = 1$, $\phi_1 = \phi_2 = 2.05$, $\gamma = 0.5$, размером роя 20 и глобальной топологией соседства [3]. Варианты алгоритма PSO [3, 18] обладают свойством локальной сходимости, а вариант алгоритма GCP SO [19] обладает также свойством глобальной сходимости.

В алгоритме PAGOS производим установку стратегии P при выполнении алгоритма оптимизации A . В алгоритме CLASSIC используем фиксированную стратегию P_p ($N_S = 16$, $N_T = 1$) и стратегию P_s нижнего уровня (параметр размера роя $P_{s,1} = 20$).

Для последовательности достигаемых экстремумов $\hat{\Psi}^{t^{\max}}$ и порогового экстремума Φ_{\min}^* произведём оценку скорости сходимости алгоритма. Последовательность $\{x_k\}$ сходится к значению x^* с линейной скоростью сходимости, если $\mu \in (0, 1)$, или с сублинейной скоростью

сходимости, если $\mu = 1$ [14], где $\lim_{t \rightarrow \infty} \frac{x_{t+1} - x^*}{x_t - x^*} = \mu$.

Последовательность $\hat{\Psi}^{t^{\max}}$ имеет сублинейную скорость сходимости к значению Φ_{\min}^* , так как для глобального метода оптимизации A допустима стагнация на некотором шаге, обусловленная как мультимодальностью задачи оптимизации, так и особенностями М-алгоритма A_s . Введём последовательность $a = \{a_i\}$ шагов алгоритма A , для которых получена стагнация $\Phi_{a_i} - \Phi_{a_i-1} < \varepsilon$ с некоторой погрешностью ε . Введём последовательность $\hat{a} = \{\hat{a}_i\}$ компенсированных шагов стагнации, для которых $(\Phi_{a_i} - \Phi_{a_i-1} < \varepsilon) \wedge (\Phi_{a_i+1} - \Phi_{a_i} > \varepsilon)$. Таким образом, на шаге $\hat{a}_i + 1$ происходит компенсация стагнации, которую испытывает алгоритм A на предыдущем шаге \hat{a}_i . Введём коэффициент компенсации стагнации $k_A = |\hat{a}|/|a|$.

Сформируем последовательность $\hat{\Psi}^{t^{\max}} = (\Psi_i, i \notin \hat{a})$ путём исключения скомпенсированных шагов стагнации. Последовательность $\hat{\Psi}$ обладает линейной скоростью сходимости с близким к единице значением константы скорости μ . При увеличении значения коэффициента компенсации стагнации k_A значение константы μ уменьшается. В рамках эксперимента константу скорости сходимости оцениваем выражением $\mu = \sup \left\{ \frac{\hat{\Psi}^{t+1} - \Phi_{\min}^*}{\hat{\Psi}^t - \Phi_{\min}^*} \right\}, t \in [0, t^{\max} - 1)$.

В вычислительных экспериментах для оценки эффективности оптимизации введём следующие коэффициенты.

Назовём *коэффициентом улучшения* отношение $\tilde{k}_\Phi = \frac{\tilde{\Phi}^{\text{CLASSIC}}(t^{\max})}{\tilde{\Phi}^{\text{PAGOS}}(t^{\max})}$, где $\tilde{\Phi}^{\text{CLASSIC}}$ — среднее значение экстремума, достигаемого классическим алгоритмом на глобальном номере итерации $F_G(t^{\max})$ (3), а $\tilde{\Phi}^{\text{PAGOS}}$ — среднее значение экстремума, достигаемого адаптивным алгоритмом PAGOS.

Назовём *коэффициентом ускорения* отношение $\tilde{k}_C = \frac{\tilde{C}^{\text{CLASSIC}}}{\tilde{C}^{\text{PAGOS}}}$, где $\tilde{C}^{\text{CLASSIC}}$ — среднее число испытаний целевой функции, выполненное классическим алгоритмом при достижении порогового значения экстремума Φ_{\min}^* , выбираемого для заданной тестовой целевой функции, а \tilde{C}^{PAGOS} — адаптивным алгоритмом PAGOS.

В табл. 1 приведены результаты эксперимента по изучению скорости сходимости. Эксперимент иллюстрирует, как быстро снижаются значения достигнутого экстремума по мере итераций алгоритма оптимизации. Приведём значения математического ожидания коэффициента улучшения \tilde{k}_Φ для различных тестовых целевых функций и различных глобальных номеров итераций $t^G = F_G(t)$. Проведём 30 независимых запусков. Критерием остановки алгоритма считаем условие (11) достижения значения $t^{\max} = 300$.

Таблица 1

Коэффициент улучшения \tilde{k}_Φ для различных тестовых функций

Задача	Итерация					
	30	50	75	100	150	200
Ackley	1.0	1.0	1.1	1.2	2.7	3.6
Rosenbrock	1.0	1.4	1.4	1.6	2.1	1.3
Sphere	0.9	1.1	1.5	2.2	5.9	9.9
Rastrigin	1.0	1.1	1.2	1.3	1.8	2.9
Schwefel	1.0	1.1	1.2	1.3	2.5	4.2
Griewank	0.9	1.1	1.2	1.1	1.3	2.7

Результаты этого эксперимента показывают, что адаптивный алгоритм получает лучшее значение экстремума $\tilde{\Phi}^*$, чем классический алгоритм при том же числе итераций. Значение ко-

эффициента улучшения $\tilde{k}_\Phi(t)$ растёт с номером итерации t . На ранних итерациях ($F_G(t) \leq 30$) адаптивный алгоритм может показывать худшую скорость сходимости.

Приведём в табл. 2, 3 результаты эксперимента по сравнению эффективности оптимизации для различных размерностей базовой задачи оптимизации ($d = 20, 50$).

Таблица 2

Сравнение индикаторов эффективности при размерности задачи $d = 20$

Задача	Алгоритм	$\tilde{\Phi}^*$	$c_{v,\Phi}$	Φ_{\min}^*	\hat{p}	\tilde{k}_C	\tilde{k}_Φ	\tilde{k}_A
Ackley	CLASSIC	$6,31 \cdot 10^{-6}$	0,29	–	100%	–	–	55%
	PAGOS	$3,39 \cdot 10^{-6}$	0,24	0,001	100%	2,3	1,9	57%
Rosenbrock	CLASSIC	11,4	0,06	–	100%	–	–	18%
	PAGOS	0,3	3,74	90	100%	1,3	43,1	77%
Sphere	CLASSIC	$1,76 \cdot 10^{-7}$	0,45	–	100%	–	–	57%
	PAGOS	0,0	1,69	0,100	100%	1,6	$5,93 \cdot 10^{13}$	92%
Rastrigin	CLASSIC	97,9	0,10	–	100%	–	–	15%
	PAGOS	15,9	0,41	120	100%	1,6	6,2	33%
Schwefel	CLASSIC	54,5	3,54	–	73%	–	–	11%
	PAGOS	$7,84 \cdot 10^{-11}$	1,09	0,100	100%	4,1	$6,96 \cdot 10^{11}$	75%
Griewank	CLASSIC	0,1	1,06	–	53%	–	–	31%
	PAGOS	0,0	1,06	0,100	100%	3,0	9,3	66%

Таблица 3

Сравнение индикаторов эффективности при размерности задачи $d = 50$

Задача	Алгоритм	$\tilde{\Phi}^*$	$c_{v,\Phi}$	Φ_{\min}^*	\hat{p}	\tilde{k}_C	\tilde{k}_Φ	\tilde{k}_A
Ackley	CLASSIC	$4,82 \cdot 10^{-5}$	0,30	–	100%	–	–	38%
	PAGOS	$7,90 \cdot 10^{-6}$	0,18	0,001	100%	3,9	6,1	59%
Rosenbrock	CLASSIC	40,3	0,26	–	97%	–	–	4%
	PAGOS	0,9	1,81	90	100%	1,1	43,4	36%
Sphere	CLASSIC	$1,09 \cdot 10^{-5}$	0,34	–	100%	–	–	40%
	PAGOS	$5,75 \cdot 10^{-17}$	0,86	0,100	100%	2,9	$1,90 \cdot 10^{11}$	98%
Rastrigin	CLASSIC	396,4	0,07	–	100%	–	–	12%
	PAGOS	85,8	0,24	500	100%	1,1	4,6	28%
Schwefel	CLASSIC	40254,4	0,16	–	0%	–	–	13%
	PAGOS	$3,80 \cdot 10^{-5}$	0,40	0,100	100%	–	$1,06 \cdot 10^9$	67%
Griewank	CLASSIC	0,0	5,38	–	97%	–	–	38%
	PAGOS	0,0	1,47	0,100	100%	3,2	7,5	81%

Для каждой тестовой функции сравним значения следующих индикаторов эффективности: $\tilde{\Phi}^*$ и $c_{v,\Phi}$ — средний достигаемый экстремум и его коэффициент вариации; Φ_{\min}^* — пороговое значение экстремума тестовой функции; \tilde{k}_Φ — коэффициент среднего улучшения; \tilde{k}_C — коэффициент среднего ускорения; \hat{p} — оценка вероятности успешной локализации экстремума Φ^* с критерием успешности $\Phi^* < \Phi_{\min}^*$; \tilde{k}_A — средний коэффициент компенсации стагнации.

Проведём 30 независимых запусков. Критерием останова алгоритма считаем условие стагнации оптимизации (10) со значением невязки $\varepsilon^{\max} = 10^{-4}$ и числом итераций стагнации $t_\varepsilon = 50$. При расчёте средних значений коэффициентов учитываем запуски алгоритма с успешной локализацией экстремума. Прочерки в таблице для значений коэффициента \tilde{k}_C означают, что классический алгоритм CLASSIC не смог достичь порогового значения экстремума ни при одном запуске.

По результатам сравнения индикаторов эффективности оптимизации видно, что адаптивный алгоритм достигает лучшего значения экстремума при достижении стагнации, чем классический алгоритм оптимизации при сравнимых числах итераций и испытаний целевой функции. Значение коэффициента среднего улучшения \tilde{k}_F составляет не менее четырёх для сложных мультимодальных тестовых функций (Rastrigin, Rosenbrock, Schwefel). Для унимодальных функций адаптивный алгоритм показывает существенно большие значения \tilde{k}_F , чем для мультимодальных.

Отметим, что коэффициент среднего улучшения \tilde{k}_F несущественно изменяется при увеличении d , что демонстрирует эффективность предлагаемого метода при различных размерностях задачи. Полученные значения коэффициента среднего ускорения \tilde{k}_C демонстрируют, что адаптивный алгоритм производит меньшее число оценок целевой функции при достижении порогового значения Φ_{\min}^* . Классический алгоритм часто неспособен достичь экстремума, особенно для задач высокой размерности.

Результаты экспериментов свидетельствуют, что применение иерархического метода установки параметров улучшает эффективность оптимизации как по значению достигаемого экстремума, так и по скорости сходимости и числу оценок целевой функции. Предлагаемый метод позволяет получить решение, близкое к оптимальному, с большей вероятностью, чем последовательный М-алгоритм, особенно для сложных мультимодальных функций с высокой размерностью пространства поиска.

ЗАКЛЮЧЕНИЕ

В работе предложен иерархический метод установки параметров параллельного популяционного метаэвристического алгоритма оптимизации. Рассмотрена двухуровневая модель параллельного алгоритма, состоящая из М-алгоритма на нижнем уровне и параллельного алгоритма оптимизации с островной моделью параллелизма на верхнем, причём на обоих уровнях выделены векторы параметров моделей алгоритма. Подход иерархического метода состоит в совместном решении задачи настройки параметров модели нижнего уровня и задачи адаптации параметров верхнего уровня. Преимуществом предложенного метода перед известными методами установки параметров является оптимизация не только варьируемых параметров базового алгоритма оптимизации, но и параметров параллельного алгоритма, влияющих как на показатели сходимости, так и на параллельную эффективность и время выполнения алгоритма. Предложен подход к синтезу функций адаптации параметров на основании критерия оптимальности с предыдущих шагов оптимизации, включающего достигнутые экстремумы и значения скорости сходимости алгоритма оптимизации.

На тестовых задачах оптимизации продемонстрировано преимущество применения адаптивных М-алгоритмов, использующих предложенный метод, по сравнению с неадаптивным последовательным М-алгоритмом как по величинам достигаемых экстремумов (до 43 раз), так и по скорости сходимости, совершая в 2.3 раз меньше испытаний целевой функции. Целесообразно применение метода для класса поведенческих алгоритмов оптимизации, поскольку это позволяет эффективно управлять размером популяции без значительного увеличения числа оценок целевой функции и изменения структуры конкретного М-алгоритма.

Дальнейшим направлением исследований представляется использование предложенного метода для установки параметров параллельных М-алгоритмов на вычислительных системах на основе графических процессоров.

ЛИТЕРАТУРА

1. *De Jong K.A.* Evolutionary Computation: A Unified Approach. Cambridge: Springer-Verl., 2006.
2. *Kennedy J., Eberhart R.* Swarm Intelligence. San Francisco: Morgan Kaufmann, 2001.
3. *Clerc M.* Particle Swarm Optimization. London: Wiley-ISTE, 2006.
4. *Dokeroglu T., Sevinc E., Kucukyilmaz T., Cosar E.* A survey on new generation metaheuristic algorithms // Comput. Indust. Engrg. 2019. V. 137; <https://doi.org/10.1016/j.cie.2019.106040>
5. *Bäck T., Fogel D., Michalewicz Z.* Evolutionary Computation 2: Advanced Algorithms and Operations. Philadelphia: Institute of Physics Publ., 2000.
6. *Lorion Y., Bogon T., Timm I.J., Drobnik O.* An agent based parallel particle swarm optimization — APPSO // IEEE Swarm Intelligence Symp. 2009. P. 52–59; <https://doi.org/10.1109/SIS.2009.4937844>
7. *Niu B., Zhu Y., He X., Wu H.* MCPSO: A multi-swarm cooperative particle swarm optimizer // Appl. Math. Comput. Special Iss. Intelligent Comput. Theory and Methodology. 2008. V. 185, N 2. P. 1050–1062; <https://doi.org/10.1016/j.amc.2006.07.026>
8. *De Jong K.* Parameter setting in EAs: a 30 year perspective // Parameter Setting in Evolutionary Algorithms. Studies Comput. Intelligence. 2007. V. 54. P. 1–18; https://doi.org/10.1007/978-3-540-69432-8_1
9. *Smit S., Eiben A.* Comparing parameter tuning methods for evolutionary algorithms // Proc. 2009 IEEE Congress on Evolutionary Comput. 2009. P. 399–406; <https://doi.org/10.1109/CEC.2009.4982974>
10. *Eiben A.E., Smit S.K.* Parameter tuning for configuring and analyzing evolutionary algorithms // Swarm and Evolutionary Comput. 2011. V. 1. P. 19–31; <https://doi.org/10.1016/j.swevo.2011.02.001>
11. *Eiben A., Michalewicz Z., Schoenauer M., Smith J.* Parameter control in evolutionary algorithms // Parameter Setting in Evolutionary Algorithms. Studies in Computational Intelligence. 2007. V. 54. P. 19–46; <https://doi.org/10.1109/4235.771166>
12. *Zhan, Z., Zhang, J., Li, Y., Chung, H.* Adaptive particle swarm optimization // IEEE Trans. Systems, Man, and Cybernetics. Part B: Cybernetics. 2009. V. 39, N 6. P. 1362–1381; <https://doi.org/10.1109/TSMCB.2009.2015956>
13. *Seliverstov E., Karpenko A.* Hierarchical model of parallel metaheuristic optimization algorithms // Procedia Computer Science. Proc 13 Internat. Symp. «Intelligent Systems 2018» (INTELS'18). 2019. V. 150. P. 441–449; <https://doi.org/10.1016/j.procs.2019.02.075>
14. *Nocedal J., Wright S.* Numerical Optimization. N. Y.: Springer Sci. & Business Media, 2006.
15. *Mendes R., Kennedy J., Neves J.* The fully informed particle swarm: Simpler, maybe better // IEEE Trans. Evolutionary Comput. 2004. V. 8. P. 204–210; <https://doi.org/10.1109/TEVC.2004.826074>
16. *Griva I., Nash S., Sofer A.* Linear and Nonlinear Optimization,. Philadelphia: Soc. Indust. Appl. Math., 2008.
17. *Li X., Tang K., Omidvar M. N., Yang Z., Qin K., China H.* Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization // Technical Report, Evolutionary Computation and Machine Learning Group, RMIT, 2013.
18. *Bonyadi M.R., Michalewicz Z.* Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm // IEEE Trans. Evolutionary Comput. 2016. V. 20, N 3. P. 370–385.
19. *Bergh F.V., Engelbrecht A.P.* A convergence proof for the particle swarm optimiser // Fund. Informat. 2010. V. 105, N 4. P. 341–374.

UDC 519.6

HIERARCHICAL METHOD OF PARAMETER SETTING FOR POPULATION-BASED METAHEURISTIC OPTIMIZATION ALGORITHMS

© 2022 E. U. Seliverstov^a

*Bauman Moscow State Technical University,
ul. 2-ya Baumanskaya 5/1, Moscow 105005, Russia*

E-mail: evgeny.seliverstov@omniverse.ru

Received 21.03.2022, revised 26.07.2022, accepted 29.09.2022

Abstract. Metaheuristic algorithms for a global optimization problem have unbound strategy parameters that affect solution accuracy and algorithm efficiency. The task of determining optimal values of unbound parameters is called a parameter setting problem that can be solved by static parameter setting methods (performed before the algorithm run) and dynamic parameter control methods (during run). The paper introduces a novel hierarchical parameter setting method for the class of population-based metaheuristic optimization algorithms. The distinctive feature of this method is usage of the hierarchical algorithm model. The lower level represents a sequential algorithm from this class, and the upper level represents an algorithm with the island parallel model. Parameter setting is performed by the hierarchical method, which composes parameter tuning for the sequential algorithm and adaptive parameter control for the parallel algorithm. Parameter control is based on vector fitness criteria which consist of a convergence rate and a solution value. An approach for estimating the convergence rate for a multistep optimization method is proposed. Experimental results for CEC benchmark problems are presented and discussed.

Keywords: global optimization, metaheuristic algorithms, parameter setting, parameter control.

DOI: 10.33048/SIBJIM.2022.25.413

REFERENCES

1. De Jong K.A. *Evolutionary Computation: A Unified Approach*. Cambridge: Springer-Verl., 2006.
2. Kennedy J., Eberhart R. *Swarm Intelligence*. San Francisco: Morgan Kaufmann, 2001.
3. Clerc M. *Particle Swarm Optimization*. London: Wiley-ISTE, 2006.
4. Dokeroglu T., Sevinc E., Kucukyilmaz T., Cosar E. A survey on new generation metaheuristic algorithms. *Comput. Indust. Engrg.*, 2019, Vol. 137; <https://doi.org/10.1016/j.cie.2019.106040>
5. Bäck T., Fogel D., Michalewicz Z. *Evolutionary Computation 2: Advanced Algorithms and Operations*. Philadelphia: Inst. Physics Publ., 2000.
6. Lorion Y., Bogon T., Timm I.J., Drobniak O. An agent based parallel particle swarm optimization—APPSO. *IEEE Swarm Intelligence Symp.*, 2009, pp. 52–59; <https://doi.org/10.1109/SIS.2009.4937844>
7. Niu B., Zhu Y., He X., Wu H. MCPSO: A multi-swarm cooperative particle swarm optimizer. *Appl. Math. Comput. Special Iss. Intelligent Comput. Theory and Methodology*, 2008, Vol. 185, No. 2, pp. 1050–1062; <https://doi.org/10.1016/j.amc.2006.07.026>
8. De Jong K. Parameter setting in EAs: a 30 year perspective. *Parameter Setting in Evolutionary Algorithms. Studies Comput. Intelligence*, 2007. V. 54. P. 1–18; https://doi.org/10.1007/978-3-540-69432-8_1

9. Smit S., Eiben A. Comparing parameter tuning methods for evolutionary algorithms. *Proc. 2009 IEEE Congress on Evolutionary Comput.*, 2009, pp. 399–406; <https://doi.org/10.1109/CEC.2009.4982974>
10. Eiben A.E., Smit S.K. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Comput.*, 2011, Vol. 1, pp. 19–31; <https://doi.org/10.1016/j.swevo.2011.02.001>
11. Eiben A., Michalewicz Z., Schoenauer M., Smith J. Parameter control in evolutionary algorithms. *Parameter Setting in Evolutionary Algorithms. Studies Comput. Intelligence*, 2007, Vol. 54, pp. 19–46; <https://doi.org/10.1109/4235.771166>
12. Zhan, Z., Zhang, J., Li, Y., Chung, H. Adaptive particle swarm optimization. *IEEE Trans. Systems, Man, and Cybernetics. Part B: Cybernetics*, 2009, Vol. 39, No. 6, pp. 1362–1381; <https://doi.org/10.1109/TSMCB.2009.2015956>
13. Seliverstov E., Karpenko A. Hierarchical model of parallel metaheuristic optimization algorithms. *Proc. Comput. Sci. Proc. 13 Internat. Symp. «Intelligent Systems 2018» (INTELS'18)*, 2019, Vol. 150, pp. 441–449; <https://doi.org/10.1016/j.procs.2019.02.075>
14. Nocedal J., Wright S. Numerical Optimization. N. Y.: Springer Sci. & Business Media, 2006.
15. Mendes R., Kennedy J., Neves J. The fully informed particle swarm: Simpler, maybe better. *IEEE Trans. Evolutionary Comput.*, 2004, Vol. 8, pp. 204–210; <https://doi.org/10.1109/TEVC.2004.826074>
16. Griva I., Nash S., Sofer A. Linear and Nonlinear Optimization. Philadelphia: Soc. Indust. Appl. Math., 2008.
17. Li X., Tang K., Omidvar M. N., Yang Z., Qin K., China H. Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization. Technical Report, Evolutionary Computation and Machine Learning Group, RMIT, 2013.
18. Bonyadi M.R., Michalewicz Z. Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm. *IEEE Trans. Evolutionary Comput.*, 2016, Vol. 20, No. 3, pp. 370–385.
19. Bergh F.V., Engelbrecht A.P. A convergence proof for the particle swarm optimiser. *Fund. Informat.*, 2010, Vol. 105, No. 4, pp. 341–374.