

УДК 004.023

ДИСКРЕТНЫЙ АЛГОРИТМ ОПТИМИЗАЦИИ НА ОСНОВЕ РАСПРЕДЕЛЕНИЯ ВЕРОЯТНОСТЕЙ С ТРАНСФОРМАЦИЕЙ ЦЕЛЕВЫХ ЗНАЧЕНИЙ

© 2024 г. К. С. Сарин^а, *

^а Томский государственный университет систем управления и радиоэлектроники
634510 Томск, ул. Ленина, д. 40, Россия

* E-mail: sarin.konstantin@mail.ru

Поступила в редакцию 18.03.2024 г.

После доработки 06.06.2024 г.

Принята к публикации 17.06.2024 г.

Оптимизационные задачи поиска в дискретном пространстве и, в частности, бинарном, где переменная может принимать только два значения, имеют большое прикладное значение. В статье предлагается новый популяционный алгоритм дискретной оптимизации, основанный на распределении вероятностей переменных. Распределения определяют вероятность выбора дискретных значений переменных при поиске и формируются с помощью трансформации целевых значений решений в их весовые коэффициенты. Работоспособность алгоритма оценивалась на унимодальных и мультимодальных тестовых функциях с бинарными переменными. Результаты эксперимента показали высокую эффективность предлагаемого алгоритма на оценках сходимости и стабильности.

Ключевые слова: методы оптимизации, дискретная оптимизация, бинарная оптимизация, метаэвристики, стохастические алгоритмы, распределение вероятностей

DOI: 10.31857/S0132347424060049, **EDN:** DYZOO

1. ВВЕДЕНИЕ

Решение задач оптимизации является необходимостью практически во всех сферах жизнедеятельности человека. Настоящая работа сосредоточена на комбинаторной оптимизации, которая занимается проблемами нахождения оптимума с дискретными значениями возможных решений. Одним из частных случаев этой проблемы является бинарная оптимизация, в которой элементы вектора решения могут принимать только два значения. Практическое применение таких задач весьма обширно. В области медицины бинарная оптимизация применялась для диагностики опухолей головного мозга [1], нахождения подмножеств согласованных признаков при прогнозировании эффективности реабилитации пациентов после коронавирусной инфекции [2], классификации сложных заболеваний [3], классификации аритмии по электрокардиограмме [4]. В экономической сфере для выбора издателей журналов при размещении рекламы [5], планировании рабочего процесса [6], планировании выпуска новой версии программного обеспечения [7], проектировании производственных ячеек [8]. В науке и технике бинарная оптимизация использовалась

для нахождения подмножества информативных признаков при построении прогностических систем [9–11], восстановлении нагрузки в первичных распределительных сетях [12], диагностики неисправности энергосистем [13], решении проблемы позиционирования антенны [14], проектировании сварных балок [15], разделении аппаратного и программного обеспечения во встроенных системах [16]. Так же в [17] отмечается, что помимо чисто комбинаторных задач, задачи с вещественными числами могут быть представлены в двоичном виде и решены в дискретном числовом пространстве.

Для решения задач бинарной оптимизации применяют два типа методов. Первый тип — это традиционные детерминированные методы оптимизации, а второй тип основан на стохастических, недетерминированных алгоритмах. Традиционными являются методы релаксации, Лагранжа, ветвей и границ, целочисленное программирование [18, 19]. Эти методы являются трудозатратными и предназначены для решения задач небольших размерностей, что на практике встречается очень редко. Кроме того, большинство традиционных методов требуют

аналитическое задание целевой функции, а также ее дифференцируемость и непрерывность. Задачи большой размерности с множеством локальных оптимумов значительно ухудшают поиск традиционных методов.

Недетерминированные методы, представляемые метаэвристическими алгоритмами [20–23], устраняют вышеперечисленные проблемы. В отличие от традиционных методов данные алгоритмы не подвергнуты “застреванию” в локальных оптимумах, в меньшей степени зависят от исходных отправных точек, не ограничены видом целевой функции и способны решать оптимизационную проблему “черного ящика” [24].

В [25] доказано, что не существует эвристического алгоритма, который мог бы работать достаточно эффективно для решения всех задач оптимизации. Разработанные в настоящее время алгоритмы дают удовлетворительные результаты при решении некоторых задач, но не всех. Поэтому в этой области ведутся активные исследования, в результате чего предлагаются новые эвристические алгоритмы.

Цель настоящей работы заключается в разработке эффективного алгоритма дискретной оптимизации, конкурирующего с популярными алгоритмами в бинарном пространстве поиска.

Основной научный вклад работы представлен следующими пунктами.

1. Разработан новый популяционный метаэвристический алгоритм оптимизации для поиска в дискретном пространстве. Алгоритм использует распределения вероятностей для выбора значений переменных. Распределения формируются с помощью трансформации целевых значений решений в весовые коэффициенты.

2. Эмпирически доказана эффективность предложенного алгоритма для поиска в бинарном пространстве с помощью критериев сходимости и стабильности. Статистический тест Уилкоксона показал значимое преимущество предлагаемого алгоритма по сравнению с генетическим алгоритмом и алгоритмом роящихся частиц для оптимизации унимодальных и мультимодальных тестовых функций.

Остальная часть статьи организована следующим образом. В п. 2 рассмотрены подходы и методы решения задач бинарной оптимизации с помощью метаэвристических алгоритмов; в п. 3 представлен новый алгоритм и детали его работы; в п. 4 описана экспериментальная часть исследования; в п. 5 обсуждены полученные результаты; в заключении сделаны выводы о проделанной работе.

2. БЛИЗКИЕ РАБОТЫ ПО ТЕМЕ ИССЛЕДОВАНИЯ

Наиболее популярные алгоритмы бинарной оптимизации относятся к алгоритмам роевого интеллекта. Подобно эволюционным они основаны на механизмах природы и представляют собой модель скоординированного поведения объектов, которые могут быть представителями флоры, фауны или физическими объектами. Эволюционные вычисления основаны на конкуренции и естественном отборе, тогда как роевой интеллект опирается главным образом на сотрудничество агентов [26].

Большинство алгоритмов роевого интеллекта разработаны для непрерывной оптимизации и для того чтобы осуществлять поиск в бинарном пространстве применяются механизмы адаптации, называемые бинаризацией [27]. Самым популярным методом бинаризации является использование трансформационных функций, которые переводят непрерывные значения элементов векторов решений в значения из диапазона $[0, 1]$. Затем применяется правило бинаризации, при котором решение преобразуется в бинарное значение из множества $\{0, 1\}$. С помощью функций трансформации были адаптированы алгоритмы роящихся частиц [17, 28], искусственных водорослей [29], шимпанзе [30], роя сальпов [31], стаи китов [32]. В [28] были исследованы различные варианты функций трансформации для алгоритма роящихся частиц. Лучшая сходимость была достигнута алгоритмом с V-образной функцией трансформации.

Метод бинаризации на основе модификации алгебраических операций преобразует вещественные операторы, используемых в формулах перемещения частиц, в их логические аналоги, что позволяет оперировать бинарными решениями. Например, вместо сложения используется операция дизъюнкции, а вместо умножения — конъюнкция. С помощью данного метода были адаптированы алгоритмы роящихся частиц [33], мозгового штурма [13], роста деревьев [34], летучих мышей [33], непрерывной муравьиной колонии [5], кукушкин поиск [35], черной дыры [36].

Квантовый метод бинаризации тоже преобразует операторы непрерывного алгоритма. В этом методе каждое допустимое решение имеет позицию и вектор квантования, который содержит вероятности принять значение 1 для соответствующего элемента решения. Вектор квантования обновляется с учетом положений глобальных и локальных лидеров. Используя данный метод, были адаптированы алгоритмы

роящихся частиц [37], искусственных водорослей [7], гравитационный поиск [38], муравьиной колонии [39].

Среди алгоритмов эволюционного интеллекта для бинарной оптимизации широко применялся генетический алгоритм [40–42]. Также были предложены его модификации, так, например, в работе [43] представлен гибрид на основе генетического алгоритма и алгоритма роящихся частиц. Сначала оба алгоритма находят решения независимо друг от друга, а затем результаты объединяются с помощью метода средневзвешенной комбинации. После этого применяется локальный поиск для нахождения окончательного решения.

Оценка эффективности алгоритмов в большинстве исследований проводилась при решении определенных прикладных задач. Для объективной оценки работы алгоритмов применяют тестовые функции, которые позволяют определить эффективность при нахождении оптимума различных целевых функций, например, унимодальных, мультимодальных, овражных, разрывных, выпуклых, вогнутых. При бинарной оптимизации применяют тестовые функции для поиска в непрерывном пространстве. Бинарное пространство поиска образуют путем дискретизации непрерывного и последующим бинарным кодированием дискретных значений [28, 44–46].

3. НОВЫЙ ДИСКРЕТНЫЙ АЛГОРИТМ ОПТИМИЗАЦИИ

В настоящей работе рассматривается проблема оптимизации, в которой минимизируется критерий эффективности. В данном разделе представлен оригинальный дискретный метаэвристический алгоритм оптимизации на основе распределения вероятностей с трансформацией целевых значений (Probability Distributions with Transformation of target values, PDT). Алгоритм является итерационным, где на каждой итерации формируется вероятностная модель. Модель определяет вероятность появления конкретного дискретного значения переменной. Вероятности формируются на основе частоты появления дискретного значения каждой переменной среди решений популяции, причем меньшее значение целевой функции должно увеличивать вклад решения в повышение вероятности. Для этого вводятся трансформационные функции, которые переводят значение целевой функции решения популяции в весовой коэффициент. На рис. 1 представлена блок-схема алгоритма.

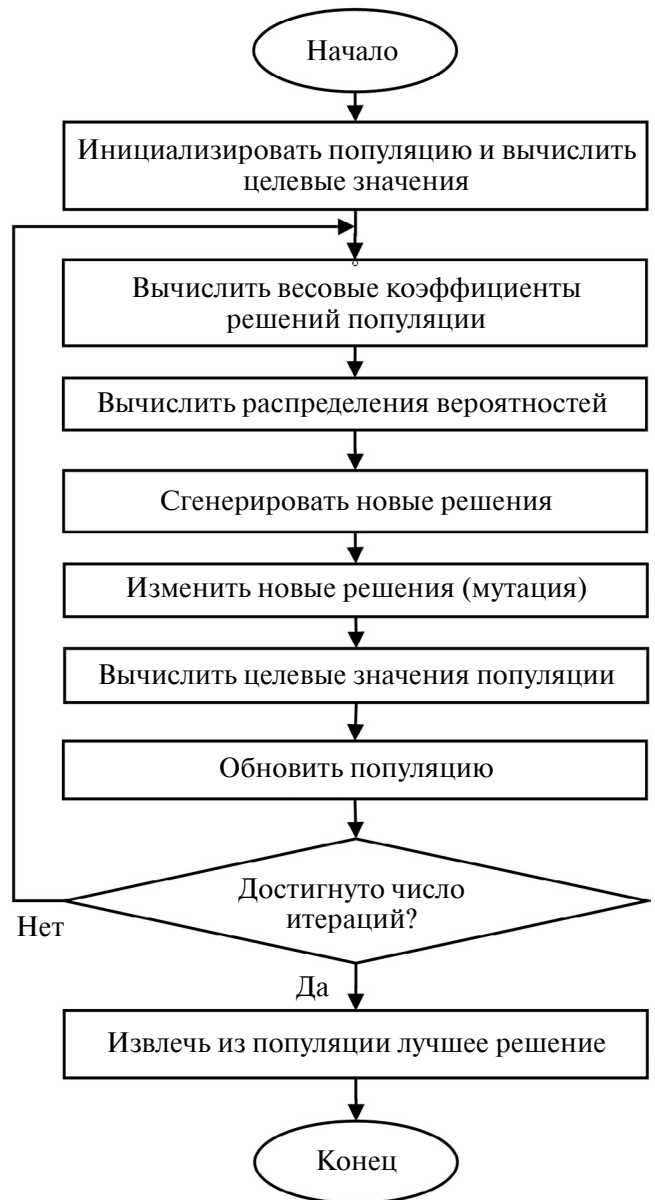


Рис. 1. Блок-схема алгоритма.

На этапе инициализации определяется начальная популяция решений Pop случайным или иным образом. Далее рассчитываются весовые коэффициенты решений популяции w . Весовой коэффициент принимает значение из диапазона $[0, 1]$, чем меньше целевое значение решения, тем больше значение w . Для того чтобы сформировать весовые коэффициенты из целевых значений предлагается использовать функции трансформации. В качестве таких функций, например, могут быть использованы следующие: T_L — линейная функция, T_S — сигмоида, T_Q — квадратичная функция, T_{Th} — гиперболический тангенс. Графики функций представлены на рис. 2. Область определения функций ограничена отрезком $[f_{\min}, f_{\max}]$, где f_{\min}

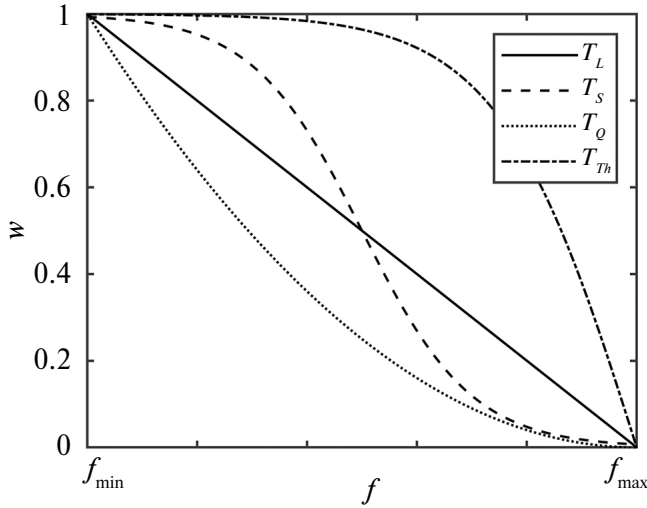


Рис. 2. Графики функций трансформаций для перевода целевых значений в веса решений.

и f_{\max} — минимальное и максимальное значение целевой функции в популяции соответственно, а f — текущее значение. Аналитические выражения функций трансформации показаны ниже:

$$T_L(f) = \frac{f_{\max} - f}{f_{\max} - f_{\min}},$$

$$T_S(f) = \frac{1}{1 + e^{(10(f - f_{\min}) / (f_{\max} - f_{\min}) - 5)}},$$

$$T_Q(f) = \left(\frac{f - f_{\min}}{f_{\max} - f_{\min}} \right)^2,$$

$$T_{Th}(f) = \left| \text{th} \left(4 \left(\frac{f - f_{\min}}{f_{\max} - f_{\min}} - 1 \right) \right) \right|.$$

После получения весовых коэффициентов решений рассчитываются распределения вероятностей. Каждое распределение состоит из вероятностей получить переменной определенное дискретное значение. Вероятности рассчитываются на основе значений переменных и весовых коэффициентов решений.

С помощью полученных распределений генерируется новая популяция и подвергается мутации, чтобы предотвратить преждевременную сходимость. Далее популяция обновляется лучшими решениями текущей и новой популяции. После этого снова рассчитываются весовые коэффициенты решений, и продолжается новый цикл. По завершении заданного количества итераций из популяции выбирается лучшее решение.

Ниже представлено пошаговое описание алгоритма.

Вход: Установить размер популяции N , число итераций $MaxIter$, вероятность мутации p_a и функцию трансформации T . Обозначим l_{jk} k -е дискретное значение j -й переменной.

Выход: R — найденное решение.

Начало

Шаг 1. Инициализация.

Случайным или иным образом сгенерировать популяцию решений $Pop = [Pop_1, Pop_2, \dots, Pop_N]$ и вычислить соответствующее целевое значение $f = [f_1, f_2, \dots, f_N]$.

Шаг 2. Инициализировать счетчик итераций $t = 1$. Начало итерационного процесса.

Шаг 3. Вычислить весовые коэффициенты решений популяции Pop с помощью функции трансформации T :

$$w_i = T(f_i),$$

где $i = 1, \dots, N$.

Шаг 4. Вычислить распределения вероятностей.

Для каждого дискретного значения k переменной j определить сумму весовых коэффициентов решений Pop , которые принимают данное дискретное значение k . Обозначим такую сумму S_{jk} , где $j = 1, \dots, n$, $k = 1, \dots, m$:

$$S_{jk} = \sum_{i=1}^N w_i \cdot \begin{cases} 1, & \text{если } Pop_{ij} = l_{jk} \\ 0, & \text{иначе} \end{cases}.$$

Вычислить эмпирическую вероятность появления k -го значения переменной j :

$$P_{jk} = \frac{S_{jk}}{\sum_{k=1}^m S_{jk}}.$$

Шаг 5. Сгенерировать новые решения.

Формируется популяция новых решений Pop^{new} на основе вероятностей P каждой переменной:

$$Pop_{ij}^{\text{new}} = l_{jk},$$

где k удовлетворяет условию $p_{k-1} < \text{rand}(0, 1) \leq p_k$,

$$p = \left[0, P_{j1}, P_{j1} + P_{j2}, \dots, \sum_{k=1}^m P_{jm} \right],$$

$j = 1, \dots, n$, $k = 1, \dots, m$.

Шаг 6. Изменить новые решения (Мутация).

Изменить значения элементов векторов решений Pop^{new} с вероятностью p_a . Новые значения выбираются случайным образом из области значений элемента вектора решений.

Шаг 7. Вычислить значение целевых функций f_i^{new} , где $i = 1, \dots, N$, для каждого решения Pop^{new} .

Шаг 8. Обновить популяцию Pop путем выбора N лучших решений из множества решений $Pop \cup Pop^{new}$. Обновить значения целевых функций f_i согласно решениям Pop .

Шаг 9. Проверка остановки алгоритма.

Если $t < MaxIter$, то $t = t + 1$ и перейти на **Шаг 3**, иначе перейти на **Шаг 10**.

Шаг 10. Извлечь в R лучшее решение из популяции Pop .

Конец

4. ЭКСПЕРИМЕНТ

В настоящем разделе представлены эксперименты с разработанным дискретным алгоритмом оптимизации на основе распределения вероятностей с трансформацией целевых значений. Алгоритм тестировался для бинарной проблемы оптимизации, т. е. когда переменные принимают только два значения. В экспериментальном исследовании использовались восемнадцать различных унимодальных и мультимодальных эталонных функций, широко применяемых для тестирования алгоритмов оптимизации [28, 44–46]. В табл. 1 представлены их характеристики, а на рис. 3 графики в двумерном пространстве поиска. Функции f_1 – f_{11} являются унимодальными, т. е. содержат только один глобальный оптимум. Функции f_{12} – f_{18} являются мультимодальными и содержат один глобальный и множество локальных оптимумов, число которых экспоненциально растет с увеличением размерности задачи. Эксперимент проводился согласно методике работы [44].

Реализация алгоритма осуществлялась на языке MATLAB в среде программирования MATLAB R2022b. Программа доступна по ссылке <https://cloud.tusur.ru/index.php/s/395znYyx87rRoDP>. Эксперимент проводился на персональном компьютере под управлением операционной системы Windows 10 с 8 Гб оперативной памяти и процессором Intel Core i7-12700.

4.1. Дискретизация непрерывных значений

Поскольку алгоритм является дискретным и оперирует в эксперименте бинарными векторами решений, элементы которых принимают значение 0 или 1, проводится кодировка вещественных значений бинарным вектором. Процедура перевода бинарного вектора решения в значения вещественных переменных представлена на рис. 4. Данная процедура выполняется всякий раз, когда алгоритму необходимо рассчитать значение целевой функции. Количество переменных в эксперименте 5, количество битов для кодиро-

вания значения каждой переменной – 15. Таким образом, величина бинарного вектора решений составляет $n = 5 \times 15 = 75$ элементов. Количество дискретных значений, которое может иметь каждая переменная, соответствует 2^{15} . Эти значения определяются с помощью равномерного квантования на диапазоне поиска переменной. Шаг дискретизации определяется следующим образом:

$$\Delta h = \frac{R_{\max} - R_{\min}}{2^{15} - 1},$$

где R_{\min} и R_{\max} – нижняя и верхняя граница диапазона значений переменной соответственно. Фактически, бинарное значение переменной – это бинарное представление порядкового номера дискретного значения на диапазоне $[R_{\min}, R_{\max}]$ с шагом дискретизации Δh .

Если переменная x кодируется бинарным вектором $[b_1, \dots, b_{15}]$, то вещественное значение этой переменной определяется следующим образом:

$$x = R_{\min} + \Delta h \sum_{i=1}^{15} b_i \cdot 2^{i-1}.$$

4.2. Критерии эффективности

Для оценки эффективности работы алгоритма применялись два критерия [47]. Первый оценивает сходимость алгоритма и определяется средним отклонением найденного целевого значения от фактического:

$$E = \frac{1}{n_{run}} \sum_{i=1}^{n_{run}} |f_i - f'|,$$

где n_{run} – количество запусков алгоритма; f_i – найденное алгоритмом значение целевой функции в i -м запуске; f' – фактическое значения оптимума целевой функции. Второй критерий оценивает стабильность работы недетерминированного алгоритма и определяется среднеквадратичным отклонением найденного оптимума целевой функции:

$$STD = \sqrt{\frac{1}{n_{run}} \sum_{i=1}^{n_{run}} (f_i - M)^2},$$

где M – среднее значение целевой функции;

$$M = \frac{\sum_{i=1}^{n_{run}} f_i}{n_{run}}.$$

Меньшее значение обоих критериев соответствует лучшему значению эффективности.

Кроме вышеприведенных критериев в работе представлены графики сходимости алгоритмов, позволяющие оценить скорость сходимости стохастических алгоритмов и показывающие

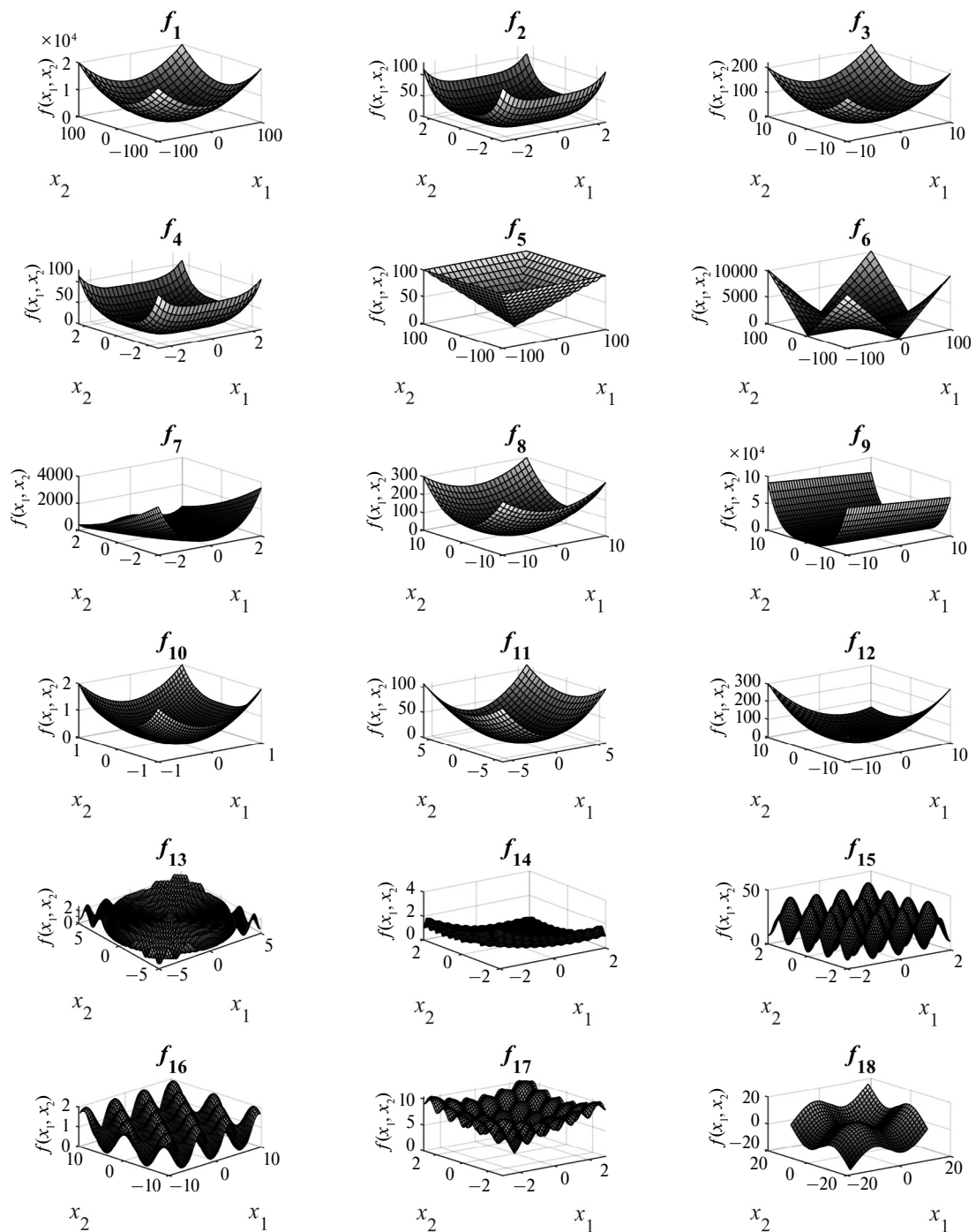


Рис. 3. Графики тестовых функций в двумерном пространстве поиска.

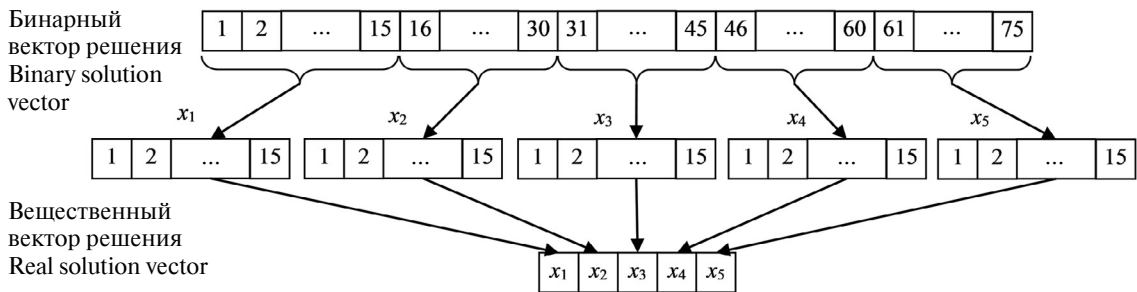


Рис. 4. Перевод бинарного вектора решения в непрерывный вектор для вычисления значения целевой функции.

Таблица 1. Тестовые функции эксперимента

№	Целевая функция	Диапазон поиска	Значение оптимума функции
1	$f_1(\mathbf{x}) = \sum_{i=1}^n x_i^2$	$[-100; 100]$	0
2	$f_2(\mathbf{x}) = \sum_{i=1}^n ix_i^4$	$[-2.56; 2.56]$	0
3	$f_3(\mathbf{x}) = \sum_{i=1}^n (x_i + 0.5)^2$	$[-10; 10]$	0
4	$f_4(\mathbf{x}) = \sum_{i=1}^n ix_i^4 + rand(0, 1)$	$[-2.56; 2.56]$	0
5	$f_5(\mathbf{x}) = \max_{i=1}^n (x_i)$	$[-100; 100]$	0
6	$f_6(\mathbf{x}) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-100; 100]$	0
7	$f_7(\mathbf{x}) = \sum_{i=1}^{n-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right)$	$[-2; 2]$	0
8	$f_8(\mathbf{x}) = \sum_{i=1}^n ix_i^2$	$[-10; 10]$	0
9	$f_9(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2$	$[-10; 10]$	0
10	$f_{10}(\mathbf{x}) = \sum_{i=1}^n x_i ^{i+1}$	$[-1; 1]$	0
11	$f_{11}(\mathbf{x}) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-6; 6]$	0
12	$f_{12}(\mathbf{x}) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$	$[-10; 10]$	-30 (при $n = 5$)
13	$f_{13}(\mathbf{x}) = 1 - \cos \left(2\pi \sqrt{\sum_{i=1}^n x_i^2} \right) + 0.1 \sqrt{\sum_{i=1}^n x_i^2}$	$[-25; 25]$	0
14	$f_{14}(\mathbf{x}) = 0.1 \left(\sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1})) + \dots \right.$ $\left. \dots (x_n - 1) (1 + \sin^2(2\pi x_n)) \right) + \sum_{i=1}^n u(x_i, 5, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-5; 5]$	0

Таблица 1. Окончание

№	Целевая функция	Диапазон поиска	Значение оптимума функции
15	$f_{15}(\mathbf{x}) = \sum_{i=1}^n (x_i - 10 \cos(2\pi x_i) + 10)$	$[-2; 2]$	0
16	$f_{16}(\mathbf{x}) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-10; 10]$	0
17	$f_{17}(\mathbf{x}) = -20e^{\left(-0,2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right)} - e^{\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right)} + 20 + e$	$[-3; 3]$	0
18	$f_{18}(\mathbf{x}) = \sum_{i=1}^n -x_i \sin\left(\sqrt{ x_i }\right)$	$[-15; 15]$	-50.0929 (при $n = 5$)

зависимость критерия E от итерации [28, 44, 45]. Значение E , приводимое на графиках, является средним значением по запускам алгоритма.

4.3. Выбор функции трансформации целевых значений

Для выбора функции трансформации целевых значений в весовые коэффициенты решений были использованы следующие функции: T_L – линейная функция, T_S – сигмоида, T_Q – квадратичная функция, T_h – гиперболический тангенс.

Алгоритм PDT с разными функциями трансформации был использован для поиска оптимума тестовых функций. Было осуществлено 30 запусков на каждой тестовой функции. Полученные значения критериев эффективности приведены в табл. 2.

Для улучшения оценки эффективности эволюционных алгоритмов в [47] отмечается, что необходимо проводить статистические тесты. Недостаточно сравнивать алгоритмы по значениям E и STD [48], необходимо провести статистический тест, чтобы доказать, что предлагаемый новый

Таблица 2. Оценка эффективности алгоритма PDT с различными функциями трансформации

f	T_L		T_S		T_Q		T_h	
	E	STD	E	STD	E	STD	E	STD
f_1	0.000047	0.000000	0.000049	0.000014	0.000047	0.000000	0.000059	0.000034
f_2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
f_3	0.402924	0.231313	0.454002	0.280296	0.478528	0.240684	0.498614	0.281871
f_4	0.005653	0.004306	0.006070	0.004650	0.005624	0.003659	0.005170	0.003590
f_5	0.041098	0.096504	0.030112	0.026176	0.042929	0.140588	0.037639	0.042328
f_6	0.015259	0.000000	0.015259	0.000000	0.015666	0.002229	0.015463	0.001114
f_7	2.629031	1.500038	2.511857	1.363161	3.214144	1.576402	3.036793	1.573228
f_8	0.000001	0.000000	0.000001	0.000000	0.000001	0.000000	0.000003	0.000003
f_9	0.885491	0.863594	0.850277	0.472212	0.803704	0.485206	0.835841	0.739384
f_{10}	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
f_{11}	0.000001	0.000000	0.000001	0.000000	0.000001	0.000000	0.000001	0.000002
f_{12}	0.239046	0.613565	0.870265	1.432694	0.885326	2.910895	0.362761	0.898002
f_{13}	0.119874	0.040684	0.149874	0.062972	0.163207	0.071839	0.123821	0.056141
f_{14}	0.057948	0.077372	0.058916	0.059387	0.053028	0.053763	0.033397	0.046236
f_{15}	0.331668	0.603435	0.199166	0.404835	0.398790	0.618058	0.099557	0.303747
f_{16}	0.036349	0.016301	0.028747	0.014743	0.029202	0.015521	0.034756	0.019396
f_{17}	0.000374	0.000041	0.000374	0.000041	0.000367	0.000000	0.000389	0.000069
f_{18}	0.407336	1.540216	0.405125	1.540966	0.610862	1.858916	0.000837	0.003885

алгоритм представляет собой значительное улучшение по сравнению с другими существующими методами.

Чтобы судить о том отличаются ли статистически значимо результаты работы алгоритма с различными функциями трансформации друг от друга, был проведен непараметрический статистический тест Фридмана на уровне значимости $\alpha = 0.05$. Значения асимптотической значимости p -value, которые меньше 0.05, можно рассматривать как убедительное свидетельство против нулевой гипотезы H_0 [47].

Тест Фридмана множественных сравнений не выявил отклонение гипотезы H_0 для обоих критериев. Гипотеза H_0 здесь утверждение об отсутствии значимых различий между вариантами алгоритма с различными функциями трансформации. Асимптотическая значимость для критерия E соответствует значению p -value = 0.757, а для критерия STD значению p -value = 0.590. Таким образом, выбор рассмотренных функций трансформации существенно не повлияет на эффективность работы алгоритма. В дальнейшем будет использоваться линейная функция.

4.4. Параметры эксперимента

Эффективность предлагаемого алгоритма PDT оценивалась в сравнении с такими популярными алгоритмами оптимизации как генетический алгоритм (GA) и бинарный алгоритм роящихся частиц (BPSO). Алгоритмы выполнялись в одинаковых условиях. Общие настройки имели следующие значения. Размер популяции — 30,

Таблица 3. Значения параметров алгоритмов

Алгоритм	Параметр	Значение
PDT	Функция трансформации T	Линейная T_L
	Вероятность мутации p_a	0.05
GA	Вид селекции	Рулеточная
	Вид скрещивания (вероятность)	Одноточечный (0.9)
	Вид мутации (вероятность)	Равномерный (0.005)
BPSO	Коэффициенты C_1, C_2	2, 2
	Вес инерции W	Линейно уменьшается с 0.9 до 0.4
	Максимальная скорость	6
	Функция трансформации	V-образная

количество итераций — 100, количество переменных — 5, число бит на одну переменную — 15, количество запусков алгоритма на каждую тестовую функцию — 30. Специфичные параметры алгоритмов GA и BPSO были установлены в значения, рекомендованные в [28, 45]. Значения специфичных параметров приведены в табл. 3.

4.5. Результаты эксперимента

В результате выполнения эксперимента были получены значения критериев эффективности каждого алгоритма. Данные значения приведены в табл. 4. Последняя строка таблицы содержит средние значения показателей. На рис. 5 показаны

Таблица 4. Оценки эффективности алгоритмов

f	GA		BPSO		PDT	
	E	STD	E	STD	E	STD
f_1	0.005349	0.028272	27.845964	37.522025	0.000047	0.000000
f_2	0.000000	0.000000	0.008293	0.018114	0.000000	0.000000
f_3	0.440548	0.254233	0.799580	0.513039	0.402924	0.231313
f_4	0.021143	0.046160	0.067826	0.054019	0.005653	0.004306
f_5	0.649841	1.354239	5.634938	3.111477	0.041098	0.096504
f_6	0.016683	0.004988	4.112874	3.333243	0.015259	0.000000
f_7	3.113280	2.491653	4.288055	2.188550	2.629031	1.500038
f_8	0.000009	0.000025	0.804053	0.875332	0.000001	0.000000
f_9	10.522179	24.073438	1.827357	1.320775	0.885491	0.863594
f_{10}	0.000000	0.000000	0.000166	0.000383	0.000000	0.000000
f_{11}	0.000004	0.000008	0.303480	0.574945	0.000001	0.000000
f_{12}	0.760543	1.068967	0.571626	0.780258	0.239046	0.613565
f_{13}	0.457784	0.259707	0.418610	0.159854	0.119874	0.040684
f_{14}	0.072918	0.062301	0.054331	0.053543	0.057948	0.077372

Таблица 4. Окончание

f	GA		$BPSO$		PDT	
	E	STD	E	STD	E	STD
f_{15}	1.227600	1.001766	1.408564	0.734589	0.331668	0.603435
f_{16}	0.042970	0.016959	0.061572	0.021475	0.036349	0.016301
f_{17}	0.000583	0.000525	0.357644	0.145604	0.000374	0.000041
f_{18}	4.935197	6.147513	3.989981	3.238601	0.407336	1.540216
	1.24	2.05	2.92	3.04	0.29	0.31

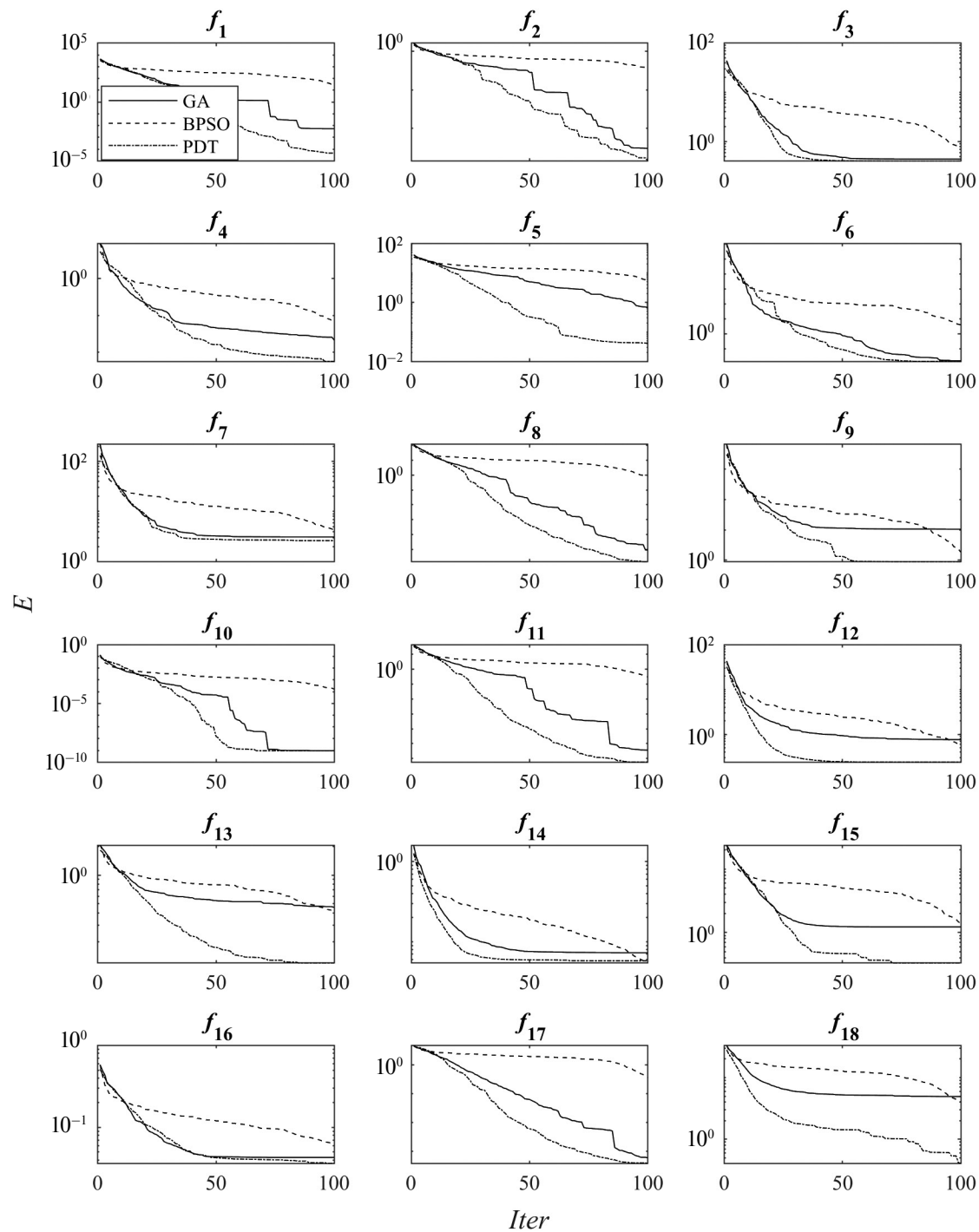


Рис. 5. Графики сходимости алгоритмов.

Таблица 5. Статистическое сравнение алгоритмов

	PDT-GA		PDT-BPSO	
	<i>E</i>	<i>STD</i>	<i>E</i>	<i>STD</i>
<i>P-value</i>	<0.01	<0.01	<0.01	<0.01
Гипотеза H_0	Отвергается	Отвергается	Отвергается	Отвергается

графики сходимости алгоритмов, позволяющие оценить скорость сходимости. Графики представлены в логарифмической шкале по оси критерия сходимости, что позволяет более четко отследить скорость сходимости алгоритмов на протяжении всей их работы.

5. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Чтобы определить, значительно ли отличаются результаты эффективности предлагаемого алгоритма от аналогов, воспользуемся парным статистическим тестом Уилкоксона [47]. Нулевая гипотеза H_0 теста утверждает отсутствие значимых различий в оценках эффективности сравниваемых алгоритмов. В табл. 5 представлены результаты сравнения. Асимптотическая значимость для критериев *E* и *STD* при сравнении с генетическим алгоритмом и алгоритмом роящихся частиц соответствует значению $p\text{-value} < 0.01$. Сумма отрицательных рангов теста превалирует над положительными. Это говорит о том, что значения критериев алгоритма PDT статистически значимо меньше алгоритмов GA и BPSO на уровне значимости $\alpha = 0.01$.

Анализ рис. 5 показывает, что на начальных итерациях скорость алгоритма роящихся частиц для тестов f_6, f_7, f_9, f_{15} и f_{16} оказывается выше остальных алгоритмов, но начиная, примерно, с пятнадцатой итерации она спадает. В целом же алгоритм на основе распределения вероятностей с трансформацией целевых значений опережает по скорости своих конкурентов.

6. ЗАКЛЮЧЕНИЕ

Предложенный дискретный алгоритм на основе распределения вероятностей с трансформацией целевых значений показал статистически значимое улучшение показателей сходимости и стабильности, таких как отклонение от оптимума и среднеквадратичное отклонение целевых значений. Сравнения проводились с генетическим алгоритмом и бинарным алгоритмом роящихся частиц. Для эксперимента использовались восемнадцать тестовых унимодальных и мультимодальных функций. В среднем отклонение от оптимума уменьшилось в 4.3 раза по сравнению

с генетическим алгоритмом и в 10.1 раза по сравнению с бинарным алгоритмом роящихся частиц. Полученные результаты говорят об эффективности предложенного алгоритма для оптимизации в бинарном пространстве.

7. БЛАГОДАРНОСТИ

Автор выражает глубокую признательность Илье Александровичу Ходашинскому, профессору Томского государственного университета систем управления и радиоэлектроники, за обсуждение настоящего исследования и ряда ценных советов и замечаний.

ИСТОЧНИК ФИНАНСИРОВАНИЯ

Исследование выполнено за счет гранта Российского научного фонда № 24-21-00168, <https://rscf.ru/project/24-21-00168/>.

СПИСОК ЛИТЕРАТУРЫ

1. Aly R.H.M., Rahouma K.H., Hamed H.F. Brain Tumors Diagnosis and Prediction Based on Applying the Learning Metaheuristic Optimization Techniques of Particle Swarm, Ant Colony and Bee Colony. *Procedia Computer Science*. 2019. V. 163. P. 165–179.
2. Ходашинский И.А., Смирнова И.Н., Бардамова М.Б., Сарин К.С., Светлаков М.О., Зайцев А.А., Тицкая Е.В., Тонкошкурова А.В., Антипова И.И., Ходашинская А.И., Зарипова Т.Н. Метод нахождения подмножеств согласованных признаков при прогнозировании эффективности реабилитации пациентов после перенесенной коронавирусной инфекции // *Сибирский журнал клинической и экспериментальной медицины*. Т. 38. № 4. С. 270–279.
3. Phogat M., Kumar D. Classification of complex diseases using an improved binary cuckoo search and conditional mutual information maximization. *Computacion y Sistemas*. 2020. V. 24. P. 1121–1129.
4. Houssein E.H., Ibrahim I.E., Neggaz N., Hassaballah M., Wazery Y.M. An efficient ECG arrhythmia classification method based on Manta ray foraging optimization. *Expert Systems with Applications*. 2021. V. 181. P. 115131.
5. Aytimur A., Babayigit B. Binary Artificial Bee Colony Algorithms for {0–1} Advertisement Problem. *Proceedings of the 2019 6th International Conference on*

- Electrical and Electronics Engineering (ICEEE), Istanbul, Turkey, 16–17 April 2019. P. 91–95.
6. *Mohammadzadeh A., Masdari M., Gharehchopogh F.S., Jafarian A.* Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing. *Evolutionary Intelligence*. 2021. V. 14. P. 1997–2025.
 7. *Pirozmand P., Ebrahimnejad A., Alrezaamiri H., Motameni H.* A novel approach for the next software release using a binary artificial algae algorithm. *Journal of Intelligent and Fuzzy Systems*. 2021. V. 40. P. 5027–5041.
 8. *Almonacid B., Aspee F., Soto, R., Crawford B., Lama J.* Solving the manufacturing cell design problem using the modified binary firefly algorithm and the egyptian vulture optimisation algorithm. *IET Software*. 2017. V. 11. P. 105–115.
 9. *Ходашинский И.А., Сарин К.С.* Отбор классифицирующих признаков с помощью популяционного случайного поиска с памятью. *Автоматика и телемеханика*. 2019. № 2. С. 161–172.
 10. *Ходашинский И.А., Сарин К.С.* Отбор классифицирующих признаков: сравнительный анализ бинарных метаэвристик и популяционного алгоритма с адаптивной памятью // *Программирование*. 2019. № 5. С. 3–9.
 11. *Sarin K., Hodashinsky I., Slezkin A.* Feature selection and identification of fuzzy classifiers based on the cuckoo search algorithm. *Communications in Computer and Information Science*. 2018. V. 934. P. 22–34.
 12. *El-Dakroury H.E.D.M., Gad A., Abdelaziz A.Y.* Load Restoration in Primary Distribution Networks Using the Binary Particle Swarm Optimization. *Proceedings of the 2019 IEEE Electrical Power and Energy Conference (EPEC), Ottawa, ON, Canada, 12–14 October 2016*. P. 1–6.
 13. *Xiong G., Shi D., Zhang J., Zhang Y.* A binary coded brain storm optimization for fault section diagnosis of power systems. *Electric Power Systems Research*. 2018. V. 163. P. 441–451.
 14. *Dahi Z.A.E.M., Mezioud C., Draa A.* A 0–1 bat algorithm for cellular network optimisation: A systematic study on mapping techniques. *International Journal of Reasoning-based Intelligent Systems*. 2017. V. 9. P. 22–42.
 15. *Hussien A.G., Hassanien A.E., Houssein E.H., Amin M., Azar A.T.* New binary whale optimization algorithm for discrete optimization problems. *Engineering Optimization*. 2020. V. 52. P. 945–959.
 16. *Mourad K., Boudour R.* A modified binary firefly algorithm to solve hardware/software partitioning problem. *Informatica*. 2021. V. 45. P. 1–12.
 17. *Kennedy J., Eberhart R.C.* A discrete binary version of the particle swarm algorithm. *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997*. 1997. V. 5. P. 4104–4108.
 18. *Serigne G., Philippe M.* A linearization framework for unconstrained quadratic (0–1) problems. *Discrete Applied Mathematics*. 2009. V. 157. P. 1255–1266.
 19. *Sherali H.D., Driscoll P.J.* Evolution, and state-of-the-art in integer programming. *Journal of Computational and Applied Mathematics*. 2000. V. 124. P. 319–340.
 20. *Ходашинский И.А.* Методы повышения эффективности роевых алгоритмов оптимизации // *Автоматика и телемеханика*. 2021. № 6. С. 3–45.
 21. *Gendreau M., Potvin J.-Yv. (Eds.)* Handbook of metaheuristics. Springer, 2019, 604 p.
 22. *Карпенко А.П.* Методы повышения эффективности метаэвристических алгоритмов глобальной оптимизации // *Математические методы в технике и технологиях – ММТТ*. 2017. Т. 1. С. 77–83.
 23. *Курейчик В.В., Родзин С.И.* Биоэвристики, инспирированные фауной (обзор) // *Информационные технологии*. 2023. Т. 29. № 11. С. 559–573.
 24. *Pardalos P.M., Rasskazova V., Vrahatis M.N. (Eds.)* Black box optimization, machine learning, and no-free lunch theorems. Springer, 2021. 388 p.
 25. *Wolpert D.H., Macready W.G.* No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*. 1997. V. 1. P. 67–82.
 26. *Becerra-Rozas M., Lemus-Romani J., Cisternas-Caneo F., Crawford B., Soto R., Astorga G., Castro C., Garcia J.* Continuous Metaheuristics for Binary Optimization Problems: An Updated Systematic Literature Review. *Mathematics*. 2022. V. 11. № 1. P. 129.
 27. *Бардамова М.Б., Буймов А.Г., Тарасенко В.Ф.* Способы адаптации алгоритма прыгающих лягушек к бинарному пространству поиска при решении задачи отбора признаков // *Доклады Томского государственного университета систем управления и радиоэлектроники*. 2020. Т. 23. № 4. С. 57–62.
 28. *Mirjalili S., Lewis A.* S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm and Evolutionary Computation*. 2013. V. 9. P. 1–14.
 29. *Turkoglu B., Uymaz S.A., Kaya E.* Binary Artificial Algae Algorithm for feature selection. *Applied Soft Computing*. 2022. V. 120. P. 108630.
 30. *Pashaei E., Pashaei E.* An efficient binary chimp optimization algorithm for feature selection in biomedical data. *Neural Computing and Applications*. 2022. V. 34. P. 6427–6451.
 31. *Jain S., Dharavath R.* Memetic salp swarm optimization algorithm based feature selection approach for crop disease detection. *Journal of Ambient Intelligence and Humanized Computing*. 2023. V. 14. — P. 1817–1835.
 32. *Mohd Yusof N., Muda A.K., Pratama S.F., Abraham A.* A novel nonlinear time-varying sigmoid transfer function in binary whale optimization algorithm for descriptors selection in drug classification. *Molecular Diversity*. 2023. V. 27. № 1. P. 71–80.

33. Merikhi B., Soleymani M. Automatic data clustering framework using nature-inspired binary optimization algorithms. *IEEE Access*. 2021. V. 9. P. 93703–93722.
34. Zhong C., Chen Y., Peng J. Feature selection based on a novel improved tree growth algorithm. *International Journal of Computational Intelligence Systems*. 2020. V. 13. P. 247–258.
35. Pandey A.C., Rajpoot D.S., Saraswat M. Feature selection method based on hybrid data transformation and binary binomial cuckoo search. *Journal of Ambient Intelligence and Humanized Computing*. 2020. V. 11. P. 719–738.
36. Yepes V., Marti J.V., Garcia J. Black hole algorithm for sustainable design of counterfort retaining walls. *Sustainability*. 2020. V. 12. P. 2767.
37. Lai X., Hao J.K., Fu Z.H., Yue D. Diversity-preserving quantum particle swarm optimization for the multidimensional knapsack problem. *Expert Systems with Applications*. 2020. V. 149. P. 113310.
38. Barani F., Mirhosseini M., Nezamabadi-Pour H. Application of binary quantum-inspired gravitational search algorithm in feature subset selection. *Applied Intelligence*. 2017. V. 47. P. 304–318.
39. Ross O.H.M. A review of quantum-inspired metaheuristics: Going from classical computers to real quantum computers. *IEEE Access*. 2019. V. 8. P. 814–838.
40. Shreem S.S., Turabieh H., Al Azwari S., Baothman F. Enhanced binary genetic algorithm as a feature selection to predict student performance. *Soft Computing*. 2022. V. 26. P. 1811–1823.
41. Nicolau M. Application of a simple binary genetic algorithm to a noiseless testbed benchmark. *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. 2009. P. 2473–2478.
42. Haupt R.L., Haupt S.E. *Practical genetic algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2004. 253 p.
43. Ghosh M., Guha R., Alam I., Lohariwal P., Jalan D., Sarkar R. Binary Genetic Swarm Optimization: A Combination of GA and PSO for Feature Selection. *Journal of Intelligent Systems*. 2019. V. 29. P. 1598–1610.
44. Bas E., Ulker E. A binary social spider algorithm for continuous optimization task. *Soft Computing*. 2020. V. 24. P. 12953–12979.
45. Mirjalili S., Mirjalili S.M., Yang X.-S. Binary bat algorithm. *Neural Computing and Applications*. 2014. V. 25. P. 663–681.
46. Pan J.-S., Hu P., Chu S.-C. Binary fish migration optimization for solving unit commitment. *Energy*. 2021. V. 226. P. 120329.
47. Derrac J., Garcia S., Molina D., Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*. 2011. V. 1. P. 3–18.
48. Garcia S., Molina D., Lozano M., Herrera F. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*. 2009. V. 15. № 6. P. 617–644.

DISCRETE OPTIMIZATION ALGORITHM BASED ON PROBABILITY DISTRIBUTION WITH TRANSFORMATION OF TARGET VALUES

© 2024 K. S. Sarin^a

^a Tomsk State University of Control Systems and Radioelectronics
Prospect Lenina 40, Tomsk, 634050 Russia

Optimization problems of searching in discrete space and, in particular, binary space, where a variable can take only two values, are of great practical importance. This paper proposes a new population discrete optimization algorithm based on probability distributions of variables. Distributions determine the probability of accepting one or another discrete value and are formed by transforming the target values of decisions into their weighting coefficients. The performance of the algorithm was assessed using unimodal and multimodal test functions with binary variables. The experimental results showed the high efficiency of the proposed algorithm in terms of convergence and stability estimates.

Keywords: optimization methods, discrete optimization, binary optimization, metaheuristics, stochastic algorithms, probability distribution