УДК 004.023

О.П. Тимофеева, С.А. Неимущев, Л.И. Неимущева

ИССЛЕДОВАНИЕ ПОПУЛЯЦИОННЫХ АЛГОРИТМОВ В РЕШЕНИИ ЗАДАЧ НЕПРЕРЫВНОЙ ОПТИМИЗАЦИИ

Нижегородский государственный технический университет им. Р.Е. Алексеева

Задачи поиска оптимальных решений часто возникают как в науке, так и в прикладных областях человеческой деятельности. Планирование событий, управление экономическими системами, проектирование производственных систем направлено на поиск наилучшего варианта для нахождения поставленной цели. Задачи могут иметь множество особенностей, что объясняет отсутствие универсального алгоритма для их решения. Особенно сложны в решении задачи со сложным ландшафтом поверхности поиска, связанные с такими понятиями, как многоэкстремальность, овражность, многокритериальность. Для эффективного решения сложных оптимизационных задач актуально использовать класс алгоритмов, называемый популяционными алгоритмами. В статье рассматриваются принципы работы обезьяньего алгоритма, поиска косяком рыб и электромагнитного алгоритма. Приведен набор задач для тестирования алгоритмов и сравнительный анализ результатов тестирования.

Ключевые слова: непрерывная оптимизация, популяционные алгоритмы, обезьяний алгоритм, поиск косяком рыб, электромагнитный алгоритм.

Введение

В последние годы глобальная оптимизация стала быстро развивающейся областью. Вопросы поиска оптимальных решений возникают во многих сферах, как в области научных исследований, так и в прикладных областях. Многие задачи оптимизации, например, в физике, химии; а также проблемы построения методов, например, в классе машинного обучения, включают нелинейные функции многих переменных с различными дополнительными атрибутами (мультимодальность, разрыв и т.д.), которые трудно оптимизировать с помощью обычных математических инструментов, таких как градиентные методы, относящихся к группе аналитических методов. Появление метаэвристических алгоритмов стало ответом на спрос на надежные, быстрые и гибкие методы оптимизации.

Для общего обозначения членов популяции используется термин «агент». В каждом виде популяционного алгоритма присутствует свой вид агента: частицы, муравьи, пчелы, обезьяны, а в более общем случае индивидуумы или особи. Популяционные алгоритмы предполагают наличие одного и более агентов. На каждом шаге решения агенты перемещаются в соответствии с конкретным алгоритмом. В зависимости от конкретного популяционного алгоритма у агента есть стратегия поведения. Это формирует сложное поведение популяции и называется роевым интеллектом.

На первых шагах решения задачи агенты занимаются исследованием поисковой области для нахождения вариантов решения, а на дальнейших этапах занимаются уточнением полученных результатов. Поскольку принцип работы популяционных алгоритмов является стохастическим, то на эффективность их работы сильно влияет результат начального приближения, полученного на стадии инициализации популяции. По этой причине для корректной оценки эффективности этих алгоритмов алгоритм прогоняется несколько раз, исходя из различных начальных приближений, полученных после инициализации. Таким образом, для оценки алгоритма используются такие критерии, как вероятность нахождения глобального экстремума, скорость сходимости и определение среднего необходимого числа прогонов (испытаний) [1].

Одной из особенностей популяционных алгоритмов является наличие большого числа свободных параметров, от которых напрямую зависит эффективность. Рекомендации по подбору свободных параметров отсутствуют, поэтому они подбираются уникально для каждой

задачи. Для исследования эффективности популяционных алгоритмов были выбраны три алгоритма, идеи которых инспирированы живой и неживой природой: обезьяний алгоритм, поиск косяком рыб и электромагнитный алгоритм.

Обезьяний алгоритм

Обезьяний алгоритм моделирует поведение обезьян в процессе их лазания по горам с целью поиска пищи. Полагают, что обезьяны исходят из того, что чем выше гора, тем больше пищи на ее вершине [2].

Метод основан на моделировании поведения обезьян, которые исследуют гористую местность с целью поиска пищи. Цель обезьян – найти самую высокую вершину горы. Эта вершина является максимальным значением целевой функции. На вершину обезьяна будет взбираться с ее начальной позиции. Этот шаг называется набором высоты или процессом движения вверх (climb process). Для каждой обезьяны, когда она добирается до вершины, естественно осмотреться и узнать, есть ли вокруг горы выше. Если есть, то обезьяна прыгнет на эту гору, то есть совершит локальный прыжок (watch-jump process), а затем повторит процесс набора высоты. После повторений процесса набора высоты и процесса локального прыжка обезьяна найдет локально максимальную вершину вокруг своей начальной позиции. Чтобы найти гораздо более высокую вершину горы, каждой обезьяне необходимо совершить глобальный прыжок (somersault process). Оптимальное решение будет получено после окончания всех повторных итераций набора высоты, локального и глобального прыжков.

Инициализация популяции. Для каждой обезьяны в популяции необходимо инициализировать позицию. Для этого в алгоритме полагается, что пространство, содержащее потенциальные оптимальные решения, определено заранее. Обычно эта область имеет правильную форму, например, n-мерный гиперкуб, поскольку компьютеру удобно отбирать случайные точки из такой области. Такая точка будет получена в том случае, если удовлетворяет всем ограничениям.

Процесс движения вверх. Процесс подъема – это пошаговая процедура для изменения позиций обезьян от исходных до новых, которые могут улучшить целевую функцию. На каждой итерации в процессе подъема случайным образом генерируется вектор шага перемещения, значения которого могут равновероятно принимать значения свободного параметра a. Если значение целевой функции в новой позиции является более оптимальным, то обезьяна перемещается вдоль этого вектора, в противном случае она перемещается в противоположную этому вектору сторону. Количество итераций определяется свободным параметром алгоритма.

Локальный прыжок. После процесса подъема каждая обезьяна прибывает на свою собственную вершину. После этого она осматривается и определяет, существуют ли другие точки вокруг неё выше текущей. Данный процесс выполняется для каждого агента популяции и заключается в поиске более оптимальной позиции по сравнению с текущей в окрестности этого агента, размер которой определяется свободным параметром алгоритма b.

Этот параметр можно характеризовать как зрение обезьян. Он является тем расстоянием, которое обезьяна может просматривать с вершины горы и на которое она, соответственно, может с неё прыгнуть. Обычно, чем больше область допустимых значений функции, тем больше задается значение данного параметра.

Глобальный прыжок. Основная цель глобального прыжка – дать возможность обезьянам изучить новые поисковые области. В качестве основы выбирается центр масс позиций всех обезьян. Еще одним свободным параметром является интервал глобального прыжка $[c, d]$. Из этого интервала случайно генерируется число α. Если $\alpha \geq 0$, то обезьяна будет совершать глобальный прыжок по направлению центра тяжести, в противном случае будет двигаться от него.

Поиск косяком рыб

Поиск косяком рыб или поиск агрегацией рыб – это оптимизационный алгоритм, основанный на поведении стаи рыб. Многие виды рыб проявляют так называемое стадное поведение, направленное в основном на повышение их выживаемости. С одной стороны, группировка рыб в стае нужна для защиты от преследования хищниками, а, с другой стороны – как средство достижения коллективной цели, то есть поиска пищи [3]. Областью допустимых решений является аквариум, при этом положение рыбы в нем отражает текущий вектор решений. Этапами алгоритма здесь являются операторы, применяемые ко всей популяции. Основные операторы были взяты из поведения агрегации рыб в живой природе и представляют собой кормление и плавание. Такая характеристика как кормление вдохновлена естественным инстинктом рыб к поиску пищи. От этого процесса зависит такой параметр алгоритма, как вес рыбы. Вес рыбы рассчитывается как разница между значением целевой функции на текущем и предыдущем шаге. Процесс кормления в алгоритме является аналогичным процессу оценки решений в оптимизации целевой функции. Этап плавания направлен на подражание коллективному движению, производимому каждой рыбой в агрегации. Процесс плавания управляется кормлением, то есть зависит от веса рыб, и в конечном счете будет направлять процесс поиска с целью получения оптимальных позиций рыб.

Оператор кормления. Под областью поиска оптимума будем понимать аквариум, в котором находятся рыбы. Рыбы привлекаются пищей, которая разбросана по аквариуму. Чтобы найти больше пищи, рыбы могут выполнять независимые передвижения отдельно от стаи. В результате этого каждая рыба может увеличиваться или уменьшаться в весе в зависимости от ее успеха или неудачи в поиске пищи. Предполагается, что изменение веса рыбы пропорционально нормализованной разнице между значением целевой функции на текущем и предыдущем шаге относительно концентрации пищи в этих позициях.

Изменение веса рыбы оценивается один раз в каждом цикле алгоритма. Для того, чтобы ограничить вес рыбы, в алгоритме существует свободный параметр, называемый максимальным весом агента популяции и обозначаемый W_{scale}. Вес рыбы может изменяться от единицы до величины этого параметра. В начале работы алгоритма вес рыб равен значению $\frac{W_{scale}}{2}$.

Оператор плавания. В алгоритме поиска косяком рыб плавание агрегации рыб является результатом нескольких явлений, таких так индивидуальное и коллективное плавание, кормление, размножение, побег от хищников, а также перемещение в более пригодные для жизни районы аквариума. Именно эти процессы разбивают этот оператор на группы: индивидуальное плавание, коллективно-инстинктивное плавание и коллективно-волевое плавание.

Индивидуальные перемещения происходят у каждой рыбы в аквариуме в каждом цикле алгоритма. Направление плавания выбирается случайным образом. Если точка перемещения агента находится в границах аквариума, то рыба оценивает плотность пищи в этой точке. С точки зрения задачи оптимизации на этом этапе проверяется, находится ли точка в области допустимых значений, а также проверяется, лучше ли значение целевой функции в этой точке. Если какое-то из этих условий не выполняется, то индивидуальное движение рыбы не происходит. Вскоре после каждого отдельного движения происходит кормление. Для индивидуального плавания определяется свободный параметр алгоритма, называемый максимальным шагом индивидуального перемещения. Индивидуальное плавание может включать в себя как одну, так и несколько итераций, поэтому этот процесс также можно назвать локальным поиском решения.

Коллективно-инстинктивное плавание. После индивидуального плавания рыб вычисляется среднее взвешенное количество отдельных движений, основанных на мгновенном успехе всех рыб в популяции. Это означает, что рыбы, которые имели успешные индивидуальные перемещения, больше других влияют на результирующее направление.

После индивидуального и коллективно-инстинктивного плавания нужна еще одна корректировка позиций рыб, а именно *коллективно-волевое плавание*. Это движение сформировано как общая оценка успеха или неудачи популяции, основанная на постепенном изменении веса агрегации рыб в целом. То есть этот этап основан на общей производительности рыб. Если стая рыб набирает вес (что означает успех поиска), то все агенты популяции смещаются в направлении текущего центра тяжести всей агрегации, а если теряет вес, то от него. Считается, что этот этап помогает в возможности разведки области допустимых решений.

Коллективно-волевое движение будет иметь разное направление в зависимости от того, увеличился или уменьшился ранее зарегистрированный вес агрегации в зависимости от нового общего веса, наблюдаемого в конце текущего цикла. Для этого определяется свободный параметр алгоритма vol, называемый волевым шагом. Этот параметр умножается на случайное число, порожденное равномерным распределением в интервале $[0, 1]$. С каждой итерацией волевой шаг алгоритма линейно уменьшается с коэффициентом, называемым коэффициентом уменьшения шага.

Электромагнитный алгоритм

Электромагнитный алгоритм имеет в своей основе идею механизма притяжения-отталкивания теории электромагнетизма. Каждый элемент популяции – это заряженная частица, которая высвобождается в пространство. Пространством является область допустимых решений, а позиция частицы в этом пространстве – потенциальным вектором решения задачи [4].

Основной характеристикой частицы является заряд частицы, зависящий от значения целевой функции в текущей позиции. Он также определяет величину притяжения или отталкивания точки по отношению к популяции – чем выше значение целевой функции, тем выше величина притяжения. Алгоритм состоит из четырех фаз: фаза инициализации популяции, вычисление суммарной силы, действующей на каждую из частиц, движение частиц вдоль направления этой силы и применение локального поиска для продвижения к локальному минимуму.

Инициализация. Процедура инициализации используется для распределения m точек равномерно по области допустимых значений, которая, в свою очередь, является -мерным гиперкубом. После того, как точка была распределена в пространстве, необходимо вычислить значение целевой функции с набором параметров, соответствующих положению этой точки. Конец процедуры наступает тогда, когда все m точек были распределены.

Локальный поиск. Процедура локального поиска используется для получения информации об окрестности каждой точки. Свободные параметры $LSITER$ и δ представляют собой количество итераций в локальном поиске и множитель для поиска в окрестности соответственно. Процедура повторяется следующим образом: во-первых, вычисляется максимально допустимая длина шага в соответствии со свободным параметром δ. Во-вторых, для i-ой частицы, улучшение в ее положении x^i происходит итеративно по каждой координате. Для каждой координаты, положение x^i назначается временной точке y для хранения исходной информации. Затем в качестве длины шага выбирается случайное число, и точка y изменяет текущую координату на выбранную длину. Если в новой точке за количество итераций $LSITER$ достигается лучшее значение целевой функции, то положение x^i заменяется положением точки y и локальный поиск для частицы i заканчивается. В конце процедуры обновляется текущая лучшая точка.

Вычисление вектора суммарной силы. Если частица s_i находится в электростатическом поле с множеством других частиц s_j, то справедлив принцип суперпозиции, в соответствии с которым результирующий вектор $\overrightarrow{F_i}$ сил, действующих на s_i, представляет собой сумму векторов $\overrightarrow{F_{ij}}$ (рис. 1).

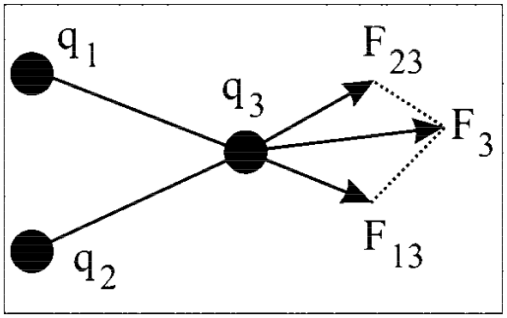

Рис. 1. Демонстрация принципа суперпозиции

На этапе вычисления вектора суммарной силы в первую очередь происходит вычисление зарядов точек согласно значениям их целевой функции. Однако в данной эвристике заряд каждой точки не является постоянным и меняется от итерации к итерации. Заряд каждой точки i, q^i, определяет силу притяжения или отталкивания точки i. Точки, которые имеют лучшие значения целевой функции, имеют более высокие заряды. Необходимо обратить внимание, что в отличие от электрических зарядов никакие знаки не привязаны к заряду отдельной точки в уравнении. Вместо этого направление определенной силы между двумя точками определяется после сравнения значений их целевой функции.

При рассмотрении двух точек точка, имеющая лучшее значение целевой функции, привлекает другую. Напротив, точка с худшим значением целевой функции отталкивает другую. Поскольку в популяции всегда существует точка, которая имеет минимальное значение целевой функции, то она действует как абсолютная точка притяжения, к которой привлекаются все остальные точки в популяции.

Движение вдоль вектора суммарной силы. После вычисления суммарного вектора силы F^i, i-ая частица перемещается в направлении этой силы на случайную длину шага. Здесь предполагается, что случайная длина шага λ равномерно распределена между 0 и 1. Очевидно, что существует множество других распределений, которые могут быть использованы при вычислении этой длины шага. Но для удобства вычислений применяется равномерное распределение. Длина шага выбирается случайным образом, чтобы гарантировать, что точки могут переместиться в еще не посещённые ранее области вдоль этого направления и вероятность такого перемещения не равна нулю.

Исследование результатов работы алгоритмов

Исследование популяционных алгоритмов проводилось на наборе тестовых математических функций без привязки к конкретной прикладной области.

Для каждого из алгоритмов свободные параметры и начальное приближение подбираются случайно, поэтому чтобы оценить эффективность работы конкретного алгоритма, необходимо использовать многократный прогон этого алгоритма. Для тестирования алгоритмов было реализовано программное обеспечение, позволяющее запускать каждый из алгоритмов на наборе тестовых функций как условной, так и безусловной оптимизации, модифицировать свободные параметры алгоритма, при этом имеется возможность запускать серию прогонов алгоритма и получить статистику по лучшему, худшему и среднему результату в серии. На рис. 2 представлен пример работы программы для обезьяньего алгоритма.

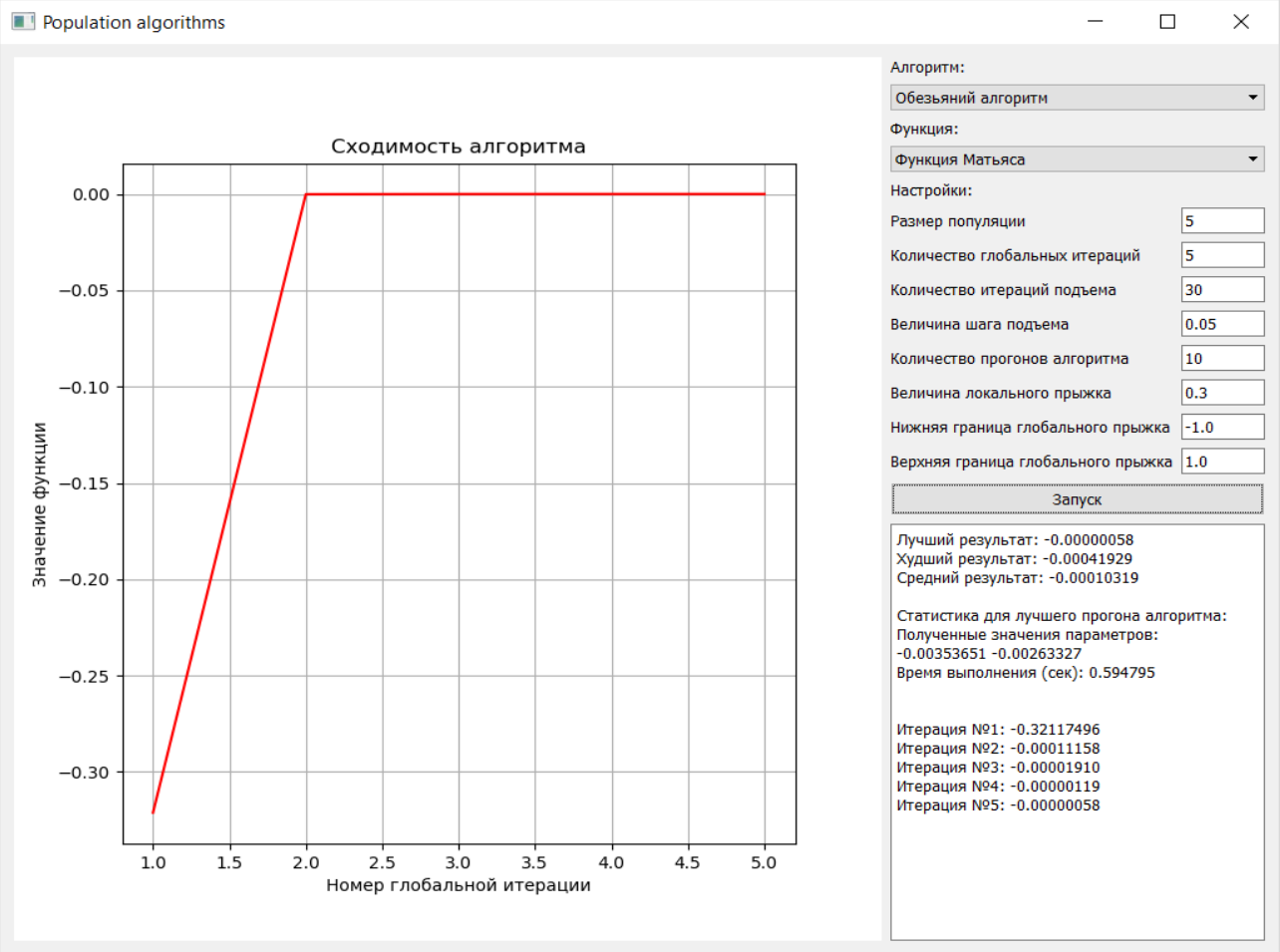

Рис. 2. Пример работы программы

Работоспособность оптимизационных алгоритмов проверялась на наборе тестовых функций, обладающих специфической топографией [5]. Тестовые функции представлены в табл. 1.

Таблица 1

Функции для тестирования оптимизационных алгоритмов

Название задачи	Постановка задачи
Функция Розенброка	$f_1 = (1 - x_1)^2 + 100(x_2 - x_1^2)^2 \to min$
Функция Растригина	$f_2 = 20 + x_1^2 + x_2^2 - 10(cos2\pi x_1 + cos2\pi x_2) \to min$
Функция Экли	$f_3 = 20 + e - 20e^{-\frac{1}{5}\sqrt{\frac{1}{2}(x_1^2+x_2^2)}} - e^{\frac{1}{2}(\cos(2\pi x_1)+\cos(2\pi x_2))} \to min$
Функция Химмельбау	$f_4 = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \to min$
Функция Матьяса	$f_5 = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2 \to min$

Окончание табл. 1.

Функция Била	$f_6 = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2 \rightarrow min$
Задача линейного программирования с пятью параметрами	$f_7 = \begin{cases} 70x_1 + 50x_2 + 4x_3 + 6x_4 + 8x_5 \rightarrow max \\ x_1 + x_2 + x_3 + x_4 + x_5 \leq 37 \\ 6x_1 + 10x_2 + 2x_3 + 2x_4 + 6x_5 \leq 168 \\ 5x_1 + 5x_2 + x_3 + 2x_4 + x_5 \leq 100 \\ 2x_1 + 3x_2 + 3x_5 \leq 30 \\ x_{1,2,3,4,5} \geq 0 \end{cases}$
Функция Розенброка с ограничениями в виде кубической и прямой	$f_8 = \begin{cases} (1 - x_1)^2 + 100(x_2 - x_1^2)^2 \rightarrow min \\ (x_1 - 1)^3 - x_2 + 1 < 0 \\ x_1 + x_2 - 2 < 0 \end{cases}$

В табл. 2 приведены результаты работы алгоритмов для тестовых функций. Для каждого алгоритма выполнено 10 прогонов и приведены три результата, где «Л» – лучший результат, «Х» – худший результат, «С» – средний результат.

Таблица 2

Результаты работы алгоритмов на тестовых функциях

f	Обезьяний алгоритм			Поиск косяком рыб			Электромагнитный алгоритм			Точный результат $f(x_1, x_2)$
	Л	Х	С	Л	Х	С	Л	Х	С	
f_1	0.0000	0.0000	0.0000	0.0129	0.2920	0.0745	0.0000	0.0030	0.0007	0
f_2	0.0000	4.9753	2.7860	0.0001	4.9837	1.3948	0.0000	0.0000	0.0000	0
f_3	0.0001	3.5745	0.3582	0.0001	0.0010	0.0007	0.0000	0.0001	0.0000	0
f_4	0.0000	0.0000	0.0000	0.0000	3.6189	0.5463	0.0000	0.0000	0.0000	0
f_5	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000	0
f_6	0.0001	0.0166	0.0048	0.0000	0.0029	0.0008	0.0000	0.0017	0.0004	0
f_7	1139.11	1115.92	1132.61	1132.37	1015.66	1084.29	1137.85	1077.75	1109.76	1144
f_8	0.0113	0.9995	0.9002	0.0000	4.9801	1.7377	0.0000	1.0397	0.7203	0

Заключение

В результате тестирования набора функций была подтверждена гибкость и универсальность популяционных алгоритмов при решении комплексных задач оптимизации. Выявлена хорошая сходимость алгоритма. Необходимо отметить, что реализация электромагнитного алгоритма дает наиболее точные результаты при решении задач с большим набором переменных. Все изученные алгоритмы решают поставленные задачи за достаточно короткое время, являются производительными, поэтому их можно применять и для решения практических задач в различных прикладных областях исследования.

Библиографический список

1. **Матренин, П.В.** Методы стохастической оптимизации: учебное пособие / П.В. Матренин, М.Г. Гриф, В.Г. Сакаев. – Новосибирск: Изд-во НГТУ, 2016. – 67 с.
2. **Zhao, R.** Monkey Algorithm for Global Numerical Optimization / R. Zhao, W. Tang // Journal of Uncertain Systems. – 2008. – № 3. – С. 165-176.
3. **Карпенко, А.П.** Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой: Учебное пособие / А.П. Карпенко. – М.: МГТУ им. Н.Э. Баумана, 2014. – 446 с.
4. **Ali M.M., Golalikhani M.** An electromagnetism-like method for nonlinearly constrained global optimization / M.M. Ali, M. Golalikhani // Computers & Mathematics with Applications. – 2010. – № 60. – С. 2279-2285.
5. Тестовые функции для оптимизации // Википедия. [2018—2018]. – URL: http://ru.wikipedia.org/wiki/Тестовые_функции_для_оптимизации (дата обращения: 01.04.2018).

Дата поступления в редакцию: 16.10.2018

O.P. Timofeeva, S.A. Neimushchev, L.I. Neimushcheva

INVESTIGATION OF POPULATION-BASED ALGORITHMS IN THE SOLUTION OF CONTINUOUS OPTIMIZATION PROBLEMS

Nizhny Novgorod state technical university n. a. R.E. Alekseev

Purpose: This article is devoted to research the effectiveness of population-based algorithms for solving various types of continuous optimization problems, selecting parameters for which the result of the algorithms will be most accurate, and also comparing the efficiency of these algorithms.
Design/methodology/approach: The methodology consists in testing three population-based algorithms: monkey algorithm, fish school search and electromagnetism-like algorithm on a set of test problems and make a comparative analysis of the accuracy of the results in comparison with the analytical solution, as well as analyzing the rate of convergence.
Findings: When using the same parameter values, the algorithms showed good convergence results on a set of tasks that differ from one another in the presence of certain attributes, as well as with the search area. In problems with a large number of parameters, the realization of the electromagnetic algorithm provided the most accurate results.
Research limitations/implications: This research opens further prospects for both studying new modifications of this family of algorithms and for applying the current implementation of the algorithms to existing practical optimization problems.
Originality/value: Implementations of algorithms created during the research have shown good results in solving test problems and can be used in applied problems or for comparative analysis with new modifications of algorithms.

Key words: continuous optimization, population-based algorithms, monkey algorithm, fish school search, electromagnetism-like algorithm.