

УДК 519.7

Обзорная статья


Краткий обзор методов оптимизации на основе роя частиц

Е. М. Казакова

Институт прикладной математики и автоматизации КВНЦ РАН
360000, КБР, г. Нальчик, ул. Шоратнова 89 А
E-mail: shogenovae@inbox.ru


Оптимизация роем частиц или particle swarm optimization (PSO) - это мета-эвристический метод глобальной оптимизации, первоначально предложенный Кеннеди и Эберхартом в 1995 году. В настоящее время это один из наиболее часто используемых методов оптимизации. В этом обзоре представлен краткий обзор исследований в области PSO за последние годы - методы инициализации роя и скорости в PSO, модификации, топологии соседства, гибридизации и обзор различных приложений PSO.

Ключевые слова: оптимизация, метод роя частиц, particle swarm optimization.

 DOI: 10.26117/2079-6641-2022-39-2-156-180

Поступила в редакцию: 07.07.2022

В окончательном варианте: 25.08.2022

Для цитирования. Казакова Е. М. Краткий обзор методов оптимизации на основе роя частиц // *Вестник КРАУНЦ. Физ.-мат. науки.* 2022. Т. 39. № 2. С. 156-180.  DOI: 10.26117/2079-6641-2022-39-2-156-180

Контент публикуется на условиях лицензии *Creative Commons Attribution 4.0 International* (<https://creativecommons.org/licenses/by/4.0/deed.ru>)

© Казакова Е. М., 2022

Введение

Алгоритмы на основе роя частиц все чаще используются для решения оптимизационных задач в различных областях. Часто при простой реализации эти алгоритмы способны эффективно решать сложные задачи оптимизации. Например, не обязательно иметь дифференцируемую и гладкую целевую функцию. Благодаря своей гибкости эти алгоритмы обладают потенциалом для решения различных реальных задач оптимизации во многих дисциплинах, особенно в инженерии и компьютерных науках.

Оптимизация методом роя частиц или particle swarm optimization (PSO), предложенная в [1], [2], представляет собой хорошо известный стохастический алгоритм на основе роя, инспирированный природой и первоначально разработанный

Финансирование. Работа выполнена без финансовой поддержки

в 1995 году Р. Эберхартом и Дж. Кеннеди, опираясь на упрощенную модель поведения птиц в стае и рыб в косяке. В PSO группа частиц (например, стая птиц) выполняет процесс поиска в пространстве решений. Каждая частица после смещения в пространстве решений делится своей текущей наилучшей позицией с одной или несколькими частицами роя. В итоге, частица определяет свой следующий шаг в пространстве решений, основываясь на пройденном пути каждой частицы. Как только все частицы обновят свои позиции на текущей итерации, начинается следующая итерация процесса поиска, осваивая области, близкие к оптимуму. В конце концов, весь рой, приближается к оптимуму целевой функции со скоростью сходимости, которая сильно зависит от выбранного варианта PSO и значений параметров [3].

Чтобы охватить научный вклад в отношении PSO, было выполнено несколько обзоров и метаанализов по PSO в различных областях применения. Бэнкс и др. [4] предложили в двух частях краткий обзор различных улучшений исходного метода PSO и его возможностей, а также проблем, связанных с алгоритмом PSO. Кроме того, в обзоре основное внимание уделялось исследованиям, касающимся адаптации для параллельной реализации, конфигурации алгоритма и динамических сред. В своем обзоре Карпенко А.П. [5] рассматривает различные модификации PSO со статической и динамической топологий соседства частиц, а также многороевый метод с миграцией частиц и параллельные методы роя частиц.

В данной статье проведен краткий обзор существующих исследований по этому направлению за последние годы. Также в обзоре представлены приложения PSO и его вариантов в различных областях, включая здравоохранение, промышленность и прочее.

1 Канонический алгоритм PSO

1.1 Описание алгоритма

Алгоритм PSO ищет оптимум, пересекая многомерное пространство решений роем частиц. Опыт каждой частицы и ее соседей способствует формированию поискового поведения частицы. Частицы координируют и делятся своими наилучшими позициями за пройденный путь с соседями (подмножеством роя или всем роем), исследуя пространство решения задачи. Таким образом, временное глобальное наилучшее решение может быть использовано для обновления возможного решения, полученного на каждой итерации. Однако это сильно зависит от топологии соседства. Например, если используется кольцевая топология (*lbest*), то информация о глобальном наилучшем положении не используется.

Как и в других метаэвристических алгоритмах [6], в PSO производится несколько основных этапов, включая инициализацию роя, вычисление фитнес-функции для каждой частицы, стратегию обновления и оценку приспособленности. Только последние два шага (обновление стратегии и оценки приспособленности) повторяются в последующих итерациях до тех пор, пока не будет удовлетворено условие завершения (например, достижение заданного числа итераций).

1.2 Процесс инициализации роя

Инициализация роя: частицы разбрасываются случайным образом на начальном этапе создания роя без определенного критерия. При последующих итерациях, обозначенных $t = 0, 1, \dots, T$, PSO начинается со случайно сгенерированного роя N , D -мерных частиц. Проще говоря с векторами положения с действительными значениями, представляющими начальные решения-кандидаты, путем инициализации положения каждой частицы x_i^0 (на итерации $t = 0$) в случайную позицию в пространстве поиска как

$$x_i^0 \sim U(x_{\min}, x_{\max})^D \quad (1)$$

где D обозначает размерность пространства поиска или размер решаемой задачи, а x_i частица имеет вектор начального положения x_i^0 со значениями, выбираемыми случайным образом из равномерно распределенного диапазона $U(x_{\min}, x_{\max})^D$, где x_{\min} и x_{\max} — границы пространства поиска. В общем случае для представления вектора положения i -ой частицы на каждой последующей итерации $t \geq 0$ можно использовать следующие обозначения:

$$x_i^t = (x_{i,1}^t, x_{i,2}^t, \dots, x_{i,D}^t). \quad (2)$$

Переменные решения (т. е. элементы вектора положения) задачи часто связаны с физическими мерами или компонентами, которые имеют естественные границы. Длина, вес и масса являются примерами таких границ, где значения переменных решения не должны быть отрицательными, поэтому следует определить диапазон $[x_{\min}, x_{\max}]$ значений, чтобы независимо ограничить значение каждой переменной решения в этом диапазоне.

Инициализация скорости: все частицы перемещаются в пространстве поиска со скоростью (размером шага), которая отражает эмпирические знания частиц, а также социальный обмен информацией о перспективных областях, посещенных в пространстве поиска, тем самым направляя процесс оптимизации в наиболее подходящие области. Аналогично инициализации положения, скорость v_i каждой частицы может быть инициализирована как

$$v_i^t = (v_{i,1}^t, v_{i,2}^t, \dots, v_{i,D}^t) \quad (3)$$

где v_i^t вектор скорости i -ой частицы на итерации t . $v_{i,1}^t, v_{i,2}^t, \dots, v_{i,D}^t$ являются параметрами скорости, каждый из которых инициализируется случайным образом (на итерации $t = 0$) в пределах установленного диапазона $[x_{\min}, x_{\max}]$ или $[-v_{\max}, v_{\max}]$, чтобы частицы не выходили из пространства поиска [11]. v_{\max} обозначается как максимально допустимый размер шага в любом измерении. v_{\max} необязателен, если правильно задать параметры управления, то он вообще не нужен.

Проверка пригодности: после инициализации роя и скорости, у каждой частицы в рое проверяется ее начальное индивидуальное лучшее положение $f(p_{\text{best}_i}^0)$, вместе с начальным глобальным (или окрестным) наилучшим положением $f(g_{\text{best}}^0)$, используя уравнения (5) и (6) соответственно. На этапе инициализации индивидуальному наилучшему решению $p_{\text{best}_i}^0$ i -ой частицы присваивается начальное положение x_i^0 .

1.3 Оценка пригодности

Частицы роя проверяются в конце каждой итерации на индивидуальное лучшее положение, а также глобальное лучшее положение. Предполагается, что каждая i -ая частица имеет уникальное значение пригодности $f(x_i^t)$ на каждой итерации t , которое вычисляется посредством оценки целевой функции. PSO запоминает индивидуальное лучшее решение (кандидат на лучшее глобальное решение), которое когда-либо встречалось каждой частице до текущей итерации t как

$$p_{best}^t = (p_{best_1}^t, p_{best_2}^t, \dots, p_{best_N}^t), \quad (4)$$

где p_{best}^t представляет индивидуальные лучшие позиции всех частиц в рое на итерации t . В зависимости от типа задачи оптимизации (минимизация или максимизация) индивидуальная лучшая позиция $p_{best_i}^t$, которую i -ая частица посетила до текущей итерации, скажем t , вычисляется как:

$$p_{best_i}^t = x_i^l \mid f(x_i^l) = \min_{k=0,1,\dots,t} \{f(x_i^k)\}, \quad (5)$$

где l — тот же индекс, что и k -ая итерация, в которой i -ая частица нашла наилучшую позицию до текущей итерации t . В конце каждой итерации все решения-кандидаты в p_{best}^t сортируются, и первое ранжированное решение выбирается как глобальное (или соседнее) наилучшее решение или позиция g_{best}^t , которую посетили i -ая частица-сосед:

$$g_{best}^t = p_{best_m}^t \mid f(p_{best_m}^t) = \min_{k=1,2,\dots,N} \{f(p_{best_i}^t)\}, \quad (6)$$

где m — тот же индекс, что и i -ая частица, занимающая в целом наилучшее положение среди частиц роя.

1.4 Обновление скорости

На каждой текущей итерации, скажем $t + 1$, скорость i -ой частицы v_i сначала регулируется путем направления ее параметров в сторону увеличения в положительном или отрицательном направлении в зависимости от сходимости текущей позиции, притягивая частицу к позициям в пространстве поиска, которые известны как хорошие из прошлого индивидуального опыта, а также из опыта других частиц в окрестности частицы. Исходный PSO [1], [2] был реализован для двух различных топологий соседства, глобального наилучшего (gbest) PSO и локального наилучшего (lbest) PSO.

Gbest PSO — это базовая версия алгоритма, представленного Кеннеди и Эберхартом [1], [2]. В поисках оптимального решения каждая частица движется: на каждой последующей итерации $t \geq 0$ — к своему более раннему индивидуальному положению p_{best}^t и глобальному лучшему положению g_{best}^t в рое путем добавления вектора скорости к положению частицы на предыдущей итерации

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1}, \quad (7)$$

где $v_{i,j}^0 = 0$, и $v_{i,j}^{t+1}$ вектор скорости i -ой частицы на итерации $t+1$ в j измерении

$$v_{i,j}^{t+1} = \omega v_{i,j}^t + c_1 r_{1,i,j}^{t+1} (p_{best,i,j}^t - x_{i,j}^t) + c_2 r_{2,i,j}^{t+1} (g_{best,j}^t - x_{i,j}^t). \quad (8)$$

Где:

инерциальный компонент $\omega v_{i,j}^t$, также называемый направлением полета, обеспечивает частице соответствующий импульс для перемещения по пространству поиска без резкого изменения направления на основе истории предыдущих направлений полета. Эта часть содержит параметр управления ω (inertia weight или IW);

когнитивный компонент $c_1 r_{1,i,j}^{t+1} (p_{best,i,j}^t - x_{i,j}^t)$ обозначает память о предыдущем индивидуальном лучшем положении, в котором получала свою лучшую результативность в прошлом;

социальный компонент $c_2 r_{2,i,j}^{t+1} (g_{best,j}^t - x_{i,j}^t)$ определяет текущую производительность, связанную с лучшими глобальными решениями, найденными до сих пор.

$j \in 1, 2, \dots, D$ обозначает размеры (или компоненты) i -ой частицы. Новые параметры, известные как коэффициенты познания и социального ускорения c_1 и c_2 , вместе с новым инерционным весом ω (inertia weight (IW)), которые были добавлены в инерционную версию PSO, представленную в [15], показали более высокую производительность, чем те, которые были с фиксированными параметрами. Параметры c_1 и c_2 , также называемые «коэффициентами ускорения», являются положительными константами, обычно используемыми для определения ускорения скорости познания i -ой частицы в направлении $p_{best,i}$ и g_{best} соответственно. ω был первоначально разработан для решения проблемы быстрого роста скорости в исходной версии PSO 1995 года [1], [2]. Кроме того, ω в сочетании с c_1 и c_2 можно использовать для баланса глобального исследования и локальной эксплуатации. $r_{1,i,j}^{t+1}$ и $r_{2,i,j}^{t+1}$ представляют собой два независимых случайных числа на текущей итерации $t+1$, которые равномерно распределены в диапазоне $[0, 1]$ и выбираются независимо для каждой i -ой частицы в каждом j измерении, чтобы поддерживать разнообразие роя. $p_{best,i,j}^t$ — наилучшее индивидуальное положение частицы или p_{best} , наилучшее положение, с которым сталкивается i -ая частица (на итерации t) в j измерении, $g_{best,j}^t$ наилучшее глобальное положение или g_{best} среди всех частиц в рое. В конце каждой итерации p_{best} и g_{best} оцениваются с помощью уравнений (5) и (6) соответственно, чтобы к концу алгоритма было получено глобальное наилучшее решение g_{best} .

lbest PSO похож на gbest PSO. Разница в том, что для lbest PSO социальная составляющая уравнения (8) определяется как

$$c_2 r_{2,i,j}^{t+1} (n_{best,j}^t - x_{i,j}^t), \quad (9)$$

где $n_{best,j}^t$ — наилучшее положение i -ой частицы в своей окрестности в j измерении, наиболее подходящее p_{best} не всего роя, как в gbest PSO, а в подмножестве роя, образующего окрестности i -ой частицы. Для lbest PSO окрестность каждой

i -ой частицы состоит из двух других частиц: $i-1$ и $i+1$. Топология соседства lbest PSO называется топологией кольца. Окрестность каждой частицы в gbest PSO — это просто весь рой; такая топология соседства называется полносвязной топологией. С момента появления в 1998 году параметр ω был включен в большинство недавно введенных вариантов PSO. По этой причине алгоритм с этим улучшением называется стандартным, каноническим или инерционным PSO (ω -PSO).

Канонический алгоритм PSO (ω -PSO):

создать и инициализировать N D -мерный рой

repeat

 for each_particle $i = 1, 2, \dots, N$ do

 if $f(x_i) < f(p_{best_i})$ then

$p_{best_i} = x_i$

 end

 if $f(p_{best_i}) < f(g_{best})$ then

$g_{best} = p_{best_i}$

 end

 end

 for each_particle $i = 1, 2, \dots, N$ do

 обновить скорость частицы используя (8);

 обновить позицию частицы используя (7);

 end

until достигнута максимальная итерация или выполнено условие остановки.

Методы инициализации роя

Многие исследователи PSO для случайной генерации положения частиц на этапе инициализации обычно предпочитают равномерное или гауссово распределение, что, не позволяет частицам полностью покрыть пространство поиска. Ряд исследователей пытались использовать другие методы инициализации роя для повышения разнообразия в PSO. В [7] влияние альтернативных функций инициализации на производительность оптимизации было изучено на основе логарифмического, нормального и логарифмически нормального распределений. Стало ясно, что различные стратегии инициализации могут в некоторых случаях значительно улучшить скорость сходимости при использовании эталонных функций. Фарук и др. [8] предложили другой метод, используя обобщенное обучение на основе оппозиций (generalized opposition-based learning - GOBL). Цель стратегии GOBL состоит в том, чтобы получить начальный рой с уже наиболее подходящими частицами, чтобы заложить прочную основу для выполнения остальной части алгоритма PSO. Внешние изменения повышают общую производительность PSO, особенно в задачах оптимизации с шумом.

Хаотический PSO (chaotic PSO - CPSO) исследовался в [9] с использованием нескольких хаотических карт. Были рассмотрены три варианта бинарного CPSO: CPSO1 с хаосом как в инициализации, так и в теле PSO, ChPSO2 с хаосом только

в частицах и ChPSO3 со случайной инициализацией частиц и в тело PSO вставлен хаос. Инициализация, выполненная на ChPSO1 и ChPSO2, подтвердила положительное влияние хаоса на PSO [9]. Недавно были применены роевые инициализации, основанные на 22 различных вероятностных распределениях, и систематически изучалось их влияние на производительность пяти различных алгоритмов [10]. Обнаружено, что предложенный механизм инициализации не оказывает существенного влияния на некоторые алгоритмы, такие как дифференциальная эволюция (differential evolution - DE), т.е. DE более устойчив, в то время как этот метод инициализации имеет большую чувствительность для других, таких как, например, PSO. Однако природа самой задачи оптимизации связана с процессом исследования такой чувствительности. Эмпирические результаты показали, что наиболее подходящими методами инициализации для ω -PSO были случайный выбор, бета-распределение и выборка латинского гиперкуба (latin hypercube sampling - LHS). Равномерная случайная инициализация, наиболее часто применяется потому, что хорошо работает во многих вариантах PSO.

Методы инициализации скорости

Было несколько мнений об инициализации скорости [12]. Установка $v_{i,j}^0 = 0$ для каждой i -ой частицы по всем измерениям является одним из методов, но этот метод ограничивает возможности исследования, т.к. существенно уменьшает начальный охват пространства поиска.

Другой метод состоит в том, чтобы установить $v_i^0 \sim U(x_{\min}, x_{\max})^D$, где более вероятно быстрое получение лучших решений, в силу того, что исследовательские способности роя значительно улучшаются за счет начальных случайных скоростей. Однако случайная инициализация увеличивает количество блуждающих частиц, что отрицательно сказывается на время сходимости и увеличивает вероятность выхода частиц за границы пространства поиска [12].

В ответ на эту проблему была предложена инициализация скорости частицы с небольшими случайными значениями для увеличения разнообразия роя. Было отмечено, что нулевая и малая случайная инициализация ведут себя одинаково. Эмпирически было показано, что фитнес глобального наилучшего решения улучшается гораздо медленнее при случайной инициализации скорости, чем при нулевых начальных скоростях или малых случайных начальных скоростях [12].

Гунасандари и др. [13] представили ограниченный по скорости булевский PSO (velocity-bounded boolean PSO - VbBoPSO), основанный на бинарном PSO (BPSO), в котором частицы инициализируются случайными бинарными позициями и бинарными скоростями, а длина скорости проверяется на диапазон значений, чтобы исследовать больше областей и для лучшей сходимости, вдохновленный [14].

BoPSO использует искусственную иммунную систему (AIS) [14], которая изменяет бинарные значения скорости с точки зрения максимально допустимой скорости v_{\max} . В VbBoPSO v_{\max} — это количество допустимых единиц в новом векторе скорости v_i^{t+1} длина вектора скорости уменьшается с использованием метода отрицательного отбора. Чтобы частицы не двигались быстрее, каждый новый вектор

скорости проверяется: превышает ли количество единиц v_{max} , затем случайным образом выбирается один бит со значением 1 в векторе скорости и устанавливается равным нулю, продолжая до тех пор, пока количество единиц не выходит за пределы v_{max} . Вариант VbBoPSO, улучшенного логического PSO с ограничением скорости (IVbBoPSO) для решения проблемы выбора признаков, был предложен для решения быстро возникающей стагнации в последующих итерациях. 28 эталонных функций были использованы для тестирования предложенных алгоритмов в сравнении с BPSO, BoPSO, BoPSO с мутацией скорости (BoPSO-vm) и BPSO на основе V-образной передаточной функции (VPSO).

2 Топология соседства

Топология соседства служит для повышения разнообразия роя при решении мультимодальных задач [16], механизм обмена информацией в рое определяет траектории поиска частиц. Наиболее распространенной структурой роя является панмиктическая, при которой выбор сопряженных частиц роя является случайным, и все частицы могут взаимодействовать друг с другом в процессе исследования. Следовательно, это способствует быстрому обмену информацией, что может привести к утрате роевого разнообразия и преждевременной сходимости. В структурированных роях обмен информацией зависит от приспособленности и топологических отношений, и свойственно как правило для близко расположенных частиц.

Подходы к структурированию роя можно разделить на две основные группы: клеточные/мелкозернистые подходы [17] и распределенные/крупнозернистые подходы [18]. Топологии соседства этих подходов изображены на рис.1 [19].

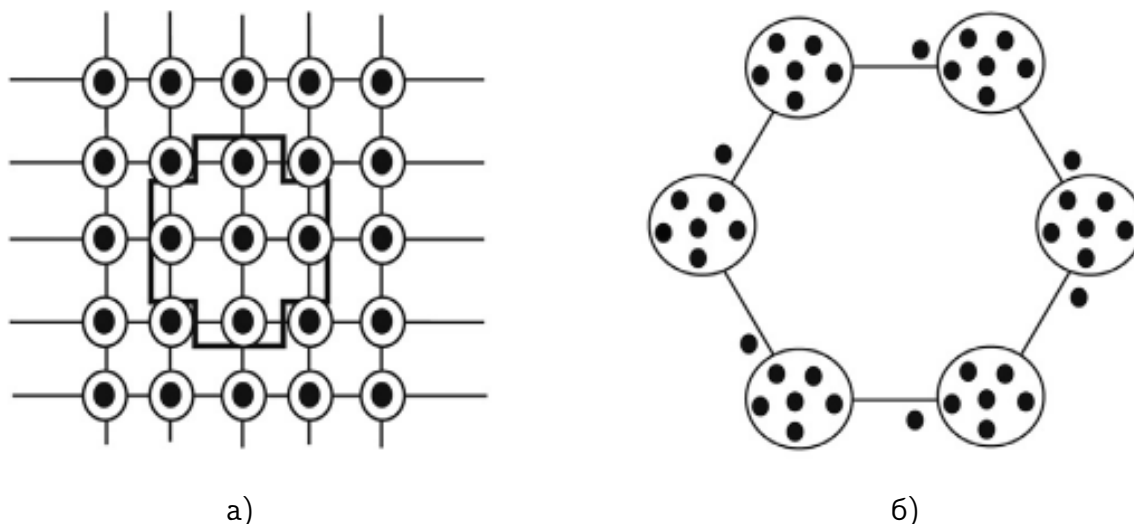


Рис. 1. Клеточный и распределенный подходы: а) клеточные/мелкозернистые, б) распределенные/крупнозернистые.

[Figure 1. Cellular and distributed approach: a) cellular/fine-grained, b) distributed/coarse-grained.]

Обычно в PSO рой инициализируется случайным образом, и каждая частица выполняет поиск, управляемый наиболее успешной частицей в рое. Топология соседства определяет широту влияния как на частиц, так и на производительность алгоритма. Для алгоритма PSO были предложены различные топологии соседства. Всесторонний обзор топологий соседства, предложенных для PSO представлен в работе [19]. Основные топологии соседства, предложенные для алгоритма PSO в этой работе, организованы на рис.2, также проиллюстрированы топологии соседства: полносвязного/gbest, кольца/lbest, колеса, случайного, фон Неймана и звезды на рис.3.



Рис. 2. Топологии соседства
[Figure 2. Neighborhood topologies]

3 Некоторые виды PSO

3.1 Адаптивный PSO

Адаптивный подход PSO был впервые предложен [20], в котором различные изменения в динамической системе автоматически отслеживаются путем введения повторной рандомизации для реагирования на динамические изменения. Позже в [21] был представлен адаптивный PSO на индивидуальном уровне с использованием критерия замещения, основанного на разнообразии между пригодностью текущей частицы и исторической лучшей приспособленностью, для адаптивного поддержания динамики роя путем удаления неактивных частиц.

Жан и др. [22] предложили один важный подход для улучшения сходимости и эффективности поиска PSO, а также для решения как унимодальных, так и мультимодальных функций. В соответствии с оценкой приспособленности каждой частицы и разнообразием (или распределением) роя в адаптивном PSO (Adaptive

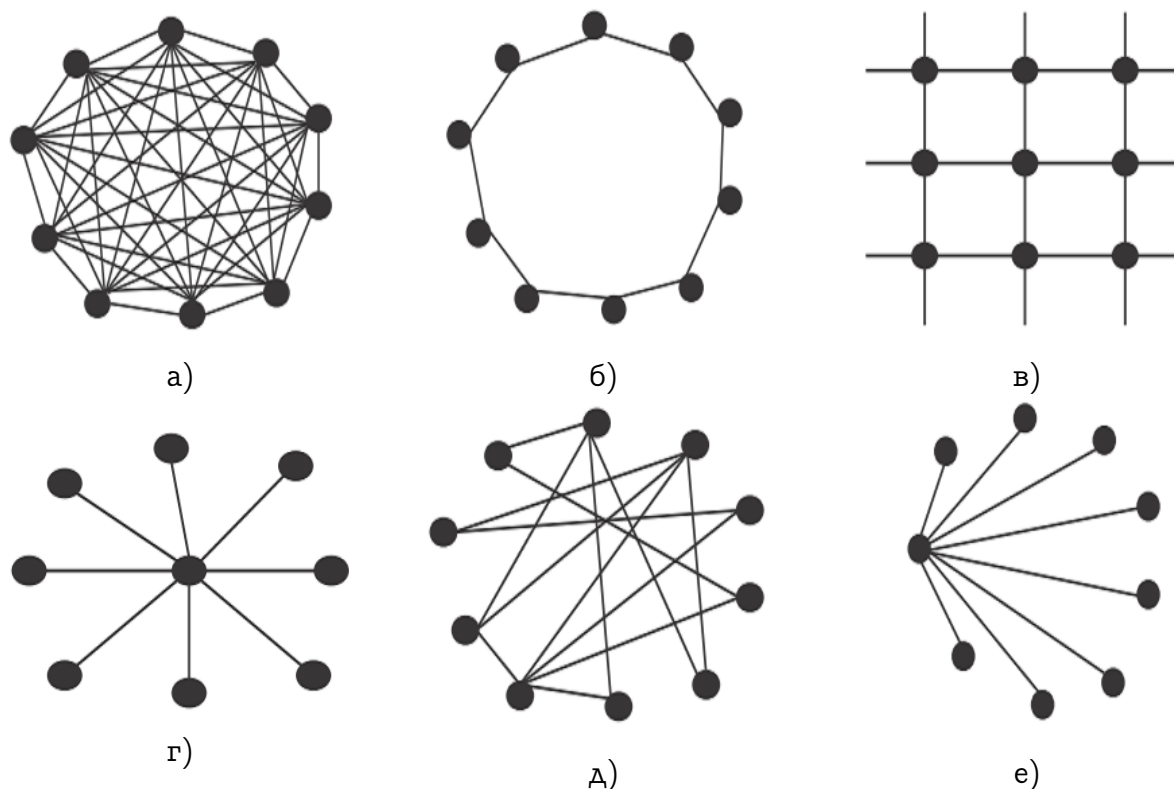


Рис. 3. Топологии соседства в рое частиц: а) полносвязная/gbest топология, б) топология кольцо/lbest, в) топология фон Неймана, г) топология звезда, д) случайная топология, е) топология колесо.

[Figure 3. Neighborhood topologies in particle swarms: a) all/gbest topology, b) ring/lbest topology, c) von Neumann topology, d) star topology, e) random topology, f) wheels topology.]

PSO - APSO), представленном этими авторами, определяются четыре эволюционных состояния PSO: исследование, эксплуатация, сходимость и выпрыгивание, для которых могут применяться различные стратегии, такие как адаптация параметров. Чтобы оценить разнообразие роя, среднее расстояние каждой частицы до всех других частиц рассчитывается как евклидова метрика

$$d_i^t = \frac{1}{N-1} \sum_{k=1, i \neq k}^N \sqrt{\left(\sum_{j=1}^D (x_{i,j}^t - x_{k,j}^t)^2 \right)},$$

Тогда e_f - эволюционный фактор, определяется выражением

$$e_f = \frac{d_g - d_{\min}}{d_{\max} - d_{\min}} \in [0, 1],$$

где d_g — значение d_i^t лучшей в глобальном масштабе частицы, а d_{\min} и d_{\max} — соответственно минимальное и максимальное расстояния между частицами. На основе этого фактора затем выбирается одно из эволюционных состояний для выражения текущего состояния алгоритма. Например, состояние разведки обозначается

значением e_f от среднего до существенного, а эксплуатация обозначается сжатым значением e_f . В свою очередь, при минимальном значении e_f наступает сходимость, а когда значение среднего расстояния лучшей частицы значительно превышает значение среднего расстояния других частиц, происходит «выпрыгивание» локальных оптимумов. На некоторых одномодальных и мультимодальных функциях APSO продемонстрировал свою способность улучшать поведение сходимости и, что наиболее важно, повышать точность алгоритма по сравнению с другими известными подходами.

Для решения проблемы подбора значений социальной и когнитивной компонент в статье [23] предлагается модификация канонического PSO, в соответствии с которой значения этих компонент не фиксированы для всего роя, а уникальны для каждой частицы и изменяются в ходе поиска, причем изменяются по принципам, основанным на эволюционных и генетических алгоритмах. В такой интерпретации каждая частица – агент, осуществляющий поиск, а значения социальной и когнитивной компонент этой частицы – генотип этого агента, определяющий каким именно образом, он реализует поиск.

3.2 PSO с квантовым поведением

В конце 1980-х годов было дано формальное определение квантового компьютера. В 2004 г. в статье [24] впервые было представлено PSO с квантовым поведением (Quantum-behaved PSO - QPSO) для улучшения скорости сходимости ω -PSO. В QPSO частицы стремятся в квантовом пространстве к глобальному оптимуму путем поиска во всем пространстве решений. В этой статье представлен алгоритм квантовой оптимизации роя частиц на основе дельта-потенциала. Для сравнения с классическим алгоритмом ω -PSO и для проверки производительности алгоритма QPSO было использовано семь эталонных функций. Результаты показали, что QPSO имеет большое преимущество перед классическим ω -PSO.

В [25] было предложено оптимальное планирование пути для свободно плавающей двухколесной маятниковой роботизированной системы на основе ее самобалансировки. Угловая траектория этого маятникового робота параметризуется с помощью QPSO. Рассматриваемая задача классифицируется как задача нелинейной оптимизации, решаемая с помощью алгоритма QPSO с целью оптимального планирования пути.

3.3 Параллельная реализация PSO

Помимо попадания в локальные оптимумы, еще одна проблема с PSO заключается в том, что с увеличением размерности решаемой задачи ухудшается производительность алгоритма [26]. Чтобы решить эту проблему, было предложено распределить обработку между несколькими процессорами компьютерной системы, создавая подгруппы, которые способствуют ускорению выполнения алгоритма. Учитывая независимость каждой подгруппы, парадигма параллельных вычислений хорошо вписывается в PSO. Для подходов parallelized PSO (PPSO) за-

дачи каждой параллельной подгруппы могут обрабатываться с использованием графического процессора (GPU) или даже многоядерного центрального процессора (CPU), потенциально обеспечивая некоторый механизм для синхронного или асинхронного обмена информацией между ними. При синхронном обмене информацией подрой полагаются друг на друга (т. е. частицы в подрое не переходят к следующей итерации до тех пор, пока не закончатся частицы в других подроях), что дает тот же результат, что и при последовательном выполнении. но с параллельной обработкой. С другой стороны, асинхронный обмен позволяет частицам в подрое перемещаться в следующую позицию на основе информации, предоставленной другими параллельными подроями в конце итерации, независимо от текущего состояния выполнения (выполняется или завершено) этих подроев. Обмен информацией можно контролировать с помощью различных архитектур, некоторые из которых: ведущий-ведомый, в котором один процессор управляет выполнением других блоков; мелкозернистый, когда подрой большого роя организованы в двумерную сетку, что позволяет общаться только между соседями каждого подроя; и крупнозернистые, в которых подрой большого роя независимы друг от друга, но они, однако, периодически обмениваются между собой частицами [26]. В ранней работе, где был реализован первый PPSO, Гис и Рахмат-Самии [27] использовали систему с 10 параллельными узлами, которая сообщила о восьмикратном увеличении производительности по сравнению с последовательным узлом. Основываясь на эти результаты, Баскар и Сугантан [28] представили новый параллельный подход, названный согласованный PSO или concurrent PSO (CONPSO).

4 Гибридизация алгоритма PSO с другими алгоритмами

При объединении алгоритма ω -PSO с другими оптимизаторами создается вариант PSO, который называется гибридным. Основная цель гибридизации состоит в том, чтобы конкретно повысить качество поиска частиц роя, а также в целом повысить как эффективность, так и производительность алгоритма. Алгоритм ω -PSO и некоторые его варианты известны своей склонностью к потенциальной ловушке локальных оптимумов, которая препятствует исследованию всего пространства поиска. Поэтому алгоритм PSO комбинируется с другими алгоритмами, чтобы преодолеть эту проблему, а также подавить недостатки других алгоритмов, с которыми он гибридизирован.

4.1 PSO с эволюционными алгоритмами

В 1998 г., подобно большинству традиционных эволюционных алгоритмов (evolutionary algorithms - EAs) в статье [29] был представлен то, что считается первым гибридным алгоритмом PSO, включив в PSO механизм выбора. Этот механизм работает путем начисления балла за наименьшее соответствие между текущими значениями приспособленности каждой частицы и текущими значениями приспособленности других частиц, чтобы эта оценка использовалась позже для

соответствующей сортировки роя. После этого позиции и скорости лучшей половины используются для замены текущих позиций и скоростей худшей половины, сохраняя индивидуальную лучшую позицию. Таким образом, частицы с низким баллом благодаря процессу выбора сбрасываются в пространстве поиска в места, которые дали лучшие результаты. В большинстве протестированных функций введение этого механизма выбора в PSO значительно улучшило производительность алгоритма. С тех пор многим исследователям, удалось комбинировать PSO с различными стратегиями изменчивости (например, мутацией [30] и операторами кроссинговерами [31]), при этом получая хорошую производительность.

Мутация — это генетический оператор, который изменяет значение следующей позиции частицы или значение g_{best} по сравнению с ее текущим состоянием с определенной вероятностью в надежде найти лучшее решение при сохранении разнообразия роя. Эта операция предотвращает преждевременную сходимость к локальному оптимуму, а также предоставляет рою широкие возможности исследования и эксплуатации. В свою очередь, кроссинговер (воспроизведение/рекомбинация/размножение) представляет собой двухэтапный процесс, включающий: 1) объединение двух частиц, случайно выбранных при заданной вероятности размножения из числа отобранных для размножения частиц, и 2) выполнение операция кроссинговера, при которой две новые частицы генерируются на основе характеристик их родителей, заменяя родителей этими новыми частицами. В этом контексте в [31] был предложен арифметический оператор кроссинговера, чтобы вычислить положение каждой новой дочерней частицы. Хотя операторы эволюционных вычислений (ЕА) обычно снижают эффективность алгоритма, они могут давать лучшие результаты, особенно в сравнении с мультимодальными функциями и многоцелевой оптимизацией. Миранда и Фонсека [32] в 2002 году объединили концепции ЕС с PSO, создав новый подход, получивший название evolutionary PSO (EPSO). В их алгоритме применяются операции мутации (по когнитивным, социальным и IW-параметрам), репликации (где замена каждой частицы происходит r раз; обычно $r = 1$), отбора и скрещивания (перед оценкой) для включения выживания и распространения наиболее приспособленной частицы, а также для создания разнообразия.

4.2 PSO с генетическими алгоритмами

Подобно PSO, генетический алгоритм (genetic algorithms - GA) имеет множество потенциальных решений-кандидатов, что приводит к PSO в сочетании с GA. В GA в каждом элементе роя есть хромосомы, мутировавшие с определенной вероятностью, для улучшения решения за счет поддержания разнообразия роя. Каждая итерация называется поколением, и процесс естественного отбора отражается алгоритмом, в котором выбираются наиболее приспособленные особи для создания следующего поколения, которое лучше, чем предыдущее. ω -PSO известен своей неспособностью эффективно избежать попадания в ловушку локальных оптимумов в области поиска. Таким образом, эту слабость можно уменьшить, используя алгоритм GA вместе с его операторами. Однако GA демонстрирует более медлен-

ное поведение сходимости, чем PSO [33]. Руководствуясь этими достоинствами и недостатками, исследователям удалось внедрить новые гибридные алгоритмы оптимизации, объединив PSO с GA.

Гибридизация GA с PSO может быть осуществлена с использованием таких популярных подходов, как последовательное или параллельное использование двух подходов или привлечение операторов GA в рамках PSO. Робинсон и др. [34] были первыми, кто представил гибридный подход PSO и GA, чтобы оптимизировать профилированную гофрированную рупорную антенну. В их подходе либо сначала используется PSO, а затем GA (PSO-GA), либо GA, а затем PSO (GA-PSO), используя результаты одного алгоритма в качестве входных данных для другого. Если один из алгоритмов не демонстрирует дальнейшего улучшения по сравнению с его текущим решением, этот алгоритм переключается между PSO и GA. PSO в сочетании с GA также использовался в некоторых приложениях, в которых операции скрещивания и мутации, как и в GA, используются для создания особей в новом поколении, запуская в то же время экземпляр PSO. Однако и GA, и PSO, в отличие от описанного выше подхода, используют один и тот же рой. После вычисления значений приспособленности в конце каждого поколения нижние 50% элементов отбрасываются, а верхняя половина помечается для созревания. Алгоритм PSO связан с обработкой метода созревания, который используется для дальнейшего улучшения наиболее эффективных элементов, вместо прямого воспроизведения следующего поколения. Затем используется турнирная селекция для выбора родителей, чтобы следующее потомство могло быть получено путем скрещивания и мутации.

Авторами статьи [35] предложена модификация алгоритма PSO названная H-алгоритом (hydra algorithm), позволяющая повысить скорость сходимости по сравнению с ω -PSO. Суть модификации состоит в том, что для определения направления движения выполняется несколько пробных шагов, для одного из которых используется метод анализа иерархий. Схема разработанного алгоритма напоминает поведение пресноводных гидр. Значение целевой функции можно интерпретировать как благоприятность условий среды. Гидра перемещается в направлении улучшения условий среды. Если в результате очередного «шага» оказывается, что эти условия ухудшились, то гидра выбирает новое направление движения. Если гидра не может найти перспективное направление движения или наступают неблагоприятные условия, то приступает к размножению половым путем, выпуская в воду яйцо и при этом погибая. Яйцо переносится водой, из яйца развивается новая особь гидры.

4.3 PSO с имитацией отжига

Имитация отжига (simulated annealing - SA), еще один метаэвристический алгоритм оптимизации, имитирует термодинамический процесс отжига, при котором микроструктура металлического материала изменяется посредством медленного и контролируемого охлаждения для улучшения его основных свойств пластичности, твердости и прочности. Когда материал достигает состояния минимальной

энергии, этот процесс заканчивается. SA, как и другие метаэвристики, не делает предположений о дифференцируемости, гладкости целевой функции. Однако, в отличие от PSO, SA принимает плохие решения с определенной вероятностью, чтобы улучшить процесс поиска за счет сохранения разнообразия [36]. Из-за популярной неспособности PSO избежать оптимальных локальных решений были предложены гибридные варианты PSO-SA с учетом того, что SA известен тем, что убегает от этих решений, совершая движения в гору, тем самым также избегая преждевременной сходимости. Однако гарантированная сходимость к глобальному минимуму не всегда обеспечивается этим алгоритмом. Кроме того, эти гибридные алгоритмы могут также иметь низкую вычислительную эффективность и производительность.

Ян и др. [36] представил алгоритм PSOSA, в котором алгоритмы PSO и SA выполняются одновременно; то есть индивидуальное наилучшее положение каждой частицы подвергается операции гауссовой мутации. Если новое значение пригодности хуже, чем предыдущее, то решение может оставаться приемлемым на основе заданной вероятности; в противном случае оно заменяется этим новым значением. Было отмечено, что производительность этого метода эффективна для набора общих эталонных функций.

Недавно Джавидрад и Назари [37] представили гибридный PSO-SA; то есть на протяжении итераций PSO может неоднократно показывать отсутствие улучшений в отношении глобальной лучшей частицы. Затем SA вмешивается, чтобы обновлять эту глобальную лучшую частицу снова и снова, пока не будет принято решение об отклонении по критерию Метрополиса. Затем PSO снова иницирует процесс поиска с новым глобальным наилучшим положением, полученным из этой новой информации. Этот процесс продолжается до тех пор, пока не будет достигнуто удовлетворительное поведение сходимости.

5 Приложения PSO

5.1 Искусственные нейронные сети с PSO

Искусственные нейронные сети (artificial neural network - ANN) подходят для обучения с использованием ряда выбранных входных и выходных векторов, преобразованных в набор данных, которые необходимо обучить для построения стандартной модели для проверки оставшихся данных [38]. В двух статьях, где впервые был представлен PSO [1],[2] Эберхарт и Кеннеди провели первый эксперимент по обучению весов ANN с использованием PSO. Они утверждали, что успешно использовали PSO для обучения многослойной ANN персептрона с прямой связью для классификации набора данных Fisher's Iris и решения проблемы исключающего ИЛИ (XOR), показывая те же результаты, что и алгоритм обратного распространения, а иногда и лучше. Эберхарт и Ху [39] адаптировали PSO для обучения многослойной ANN персептрона с прямой связью, используя сигмоидальную функцию активации для классификации типов тремора при болезни Паркинсона. Хотя этот набор данных невелик, ANN была успешно обучена с использованием PSO с высо-

кой производительностью и низкой ошибкой. При обучении ANN с использованием алгоритма обратного распространения, в отличие от PSO, требует дифференцируемую функцию, склонен к попаданию в локальные минимумы и имеет медленную сходимость [40].

5.2 Разные приложения PSO

За последние два десятилетия, параллельно с исследованиями в области алгоритмов, исследователи PSO также уделяли особое внимание исследованиям, связанным с PSO, для конкретных приложений. Каждая область применения PSO влечет за собой некоторые проблемы, которые необходимо устранить, чтобы найти успешные решения, которые затем сделают PSO более эффективным и действенным для применения в реальных условиях.

В статье [41] предлагается структура для диагностики болезни Альцгеймера, которая состоит из предварительной обработки МРТ-изображений, выделения признаков, анализа главных компонент и модели машины опорных векторов (support vector machine - SVM). В частности, для оптимизации параметров SVM предлагается новый алгоритм PSO с задержкой переключения (Switching-Delayed-PSO - SDPSO). Разработанная структура, основанная на модели SDPSO-SVM, успешно применяется для классификации болезни Альцгеймера и легких когнитивных нарушений с использованием МРТ-сканов из набора данных ADNI (Alzheimer's Disease Neuroimaging Initiative). Разработанный алгоритм может обеспечить точность классификации для 6 типичных случаев. Кроме того, результаты эксперимента показывают, что предложенный алгоритм превосходит несколько моделей SVM, а также два других современных метода со встроенным глубоким обучением, тем самым выступая в качестве эффективного метода диагностики болезни Альцгеймера.

В биоинформатике, чтобы определить последовательность длинной цепи ДНК, сначала необходимо разделить ее на подмножества фрагментов. Затем алгоритмы сборки генома рассматривают полученные фрагменты одновременно, находя их перекрытия (overlap), объединяя их по перекрытиям (layout) и исправляя ошибки в объединенной строке (consensus) (overlap-layout-consensus). В статье [42] предлагается меметический алгоритм PSO, основанный на двух операторах инициализации и операторе локального поиска, для решения проблемы ДНК с использованием модели overlap-layout-consensus для максимизации измерения перекрывающихся показателей. Результаты, основанные на 19 наборах данных фрагментов ДНК с покрытием, показывают, что алгоритм PSO, сочетающий поиск табу и локальный поиск с имитацией переменного поиска на основе отжига, может достичь наилучших показателей перекрытия.

В работе [43] с помощью модифицированного метода ω -PSO PSO-M2 решена 28-мерная задача молекулярного докинга для молекул протеазы HIV1 и дарунавиром 3U7S в качестве молекул-рецептора и лиганда соответственно. Результаты вычислительного эксперимента показали, что метод PSO-M2 успешно находит по-

ложение лиганда, близкое к нативному и может быть рекомендован для решения задач молекулярного докинга как альтернатива генетическому алгоритму.

Системы музыкальных рекомендаций - важная область исследований из-за легкой доступности и использования онлайн-музыки. В работе [44] рассматривается гибридная система музыкальных рекомендаций (hybrid music recommender system - HMRS), которая работает на основе контекста и совместных подходов. Извлекается свойства музыки из истории прослушивания пользователя и оценивается эффективность системы рекомендации музыки на основе контекстных предпочтений пользователя. Реализован интеллектуальный алгоритм оптимизации для создания оптимизированного ранжированного списка рекомендуемых результатов с помощью PSO. PSO автоматически оптимизирует качество работы музыкальных рекомендательных систем, используя возможности роевого интеллекта, поскольку она обеспечивает наилучшую эффективность поиска. Тестирование HMRS проводилось на наборе данных - Last.fm, который содержит 19 150 868 журналов прослушивания музыки 992 пользователями (до 4 мая 2009 г.), чтобы продемонстрировать эффективность алгоритма.

Задача оптимального управления генерирующим потребителем с возобновляемыми источниками энергии в интеллектуальной сети с распределенной генерацией и двусторонними потоками энергии рассмотрена в статье [45]. Для управления генерирующим потребителем использована модель выбора управляющих действий на основе логических правил, что приводит к необходимости решения задачи структурно-параметрической оптимизации списка правил и значений их численных параметров. Решение данной задачи оптимизации выполнено алгоритмом PSO и другими роевыми алгоритмами, а также алгоритмом градиентного спуска. Вычислительные эксперименты проведены для моделей ветроэнергетических систем на дальневосточных островах Русский и Попова. Полученные результаты показали высокую эффективность роевых алгоритмов, которые продемонстрировали надежную и быструю сходимость к глобальному экстремуму задачи оптимизации при различных вариантах взаимодействия и различных параметрах генерирующих потребителей.

В работе [46] для решения проблемы балансировки нагрузки в сети рассматривался PSO. В ходе исследования выявили, что при невысокой нагрузке сети нецелесообразно использовать PSO. Поэтому предлагается модификация алгоритма PSO, уменьшающая вычислительную нагрузку при низкой и средней нагрузке и время, необходимое на обработку задачи.

В работе [47] рассматриваются модификации роевых алгоритмов, включая экспоненциальный алгоритм роя частиц (exponential PSO - EPSO) для решения задачи сегментации изображений. Сегментацией называется разбиение изображения на некоторые области на основе их спектральных или пространственных характеристик. Главной идеей предлагаемого авторами гиперэвристического метода является применение нескольких эвристик, каждая из которых имеет свои слабые и сильные места, а затем их использования в зависимости от текущего состояния решения. Разработанный гиперэвристический роевой метод сегментации программно

реализован и исследован на множестве тестовых изображений. Тестирование проводилось на эталонных изображениях из базы Ossirix, выборка из 150 изображений с различными начальными условиями – хорошее качество (отсутствуют шумы и прочие артефакты); с наличием шума; контрастные изображения, размытые изображения. В ходе тестирования было выявлено, что данный метод сегментации имеет полиномиальную временную сложность.

Для разработки методики обнаружения DDoS-атак на основе нейронных сетей авторами статьи [48] проведены исследования эффективности работы ряда методов роевого интеллекта. По результатам исследований для настройки параметров нейронной сети выбран метод роя частиц, который показал высокую скорость и точность вычислений. На основе реализованной нейронной сети проводилась настройка параметров метода роя частиц, который был использован как алгоритм обучения нейронной сети совместно с методом обратного распространения ошибки для повышения эффективности процесса обучения.

Исследуются эффективность управления с прогнозирующими моделями и PSO в улучшении оперативного управления системой перекачки воды Зеррине-Руд в Иране в работе [49]. Методы оптимизации обладают значительной способностью управлять задачами в сложных и обширных системах, таких как линии электропередач, и способны удовлетворить все условия задачи.

Базовые параметры водохранилища, насосной станции и данные о ежедневном потреблении являются входными данными. В этом исследовании разрабатывается модель с двумя целями, включая минимизацию отвода безопасного объема в резервуарах для преодоления критических условий, минимизацию внезапных колебаний расхода через насосные станции и различные ограничения используются в качестве граничных условий. Полученные результаты показали, что алгоритм PSO позволил снизить расход насосов за период эксплуатации на 10%. Тем не менее алгоритм не мог уменьшить флуктуацию при имеющихся данных и условиях, что было одной из целей, которые учитывались в целевой функции.

Новый алгоритм PSO предлагается в статье [50] для повышения точности традиционных подходов к кластеризации с приложениями для анализа данных о посещаемости пациентов в режиме реального времени из отделения неотложной помощи в местной больнице Великобритании. В предложенном алгоритме PSO с распределенной задержкой RODDPSO (randomly occurring distributedly delayed PSO) со случайным возникновением четыре эволюционных состояния (эксплуатация, исследование, сходимости и выпрыгивание) определяются на основе поисковых характеристик частиц. Скорость и положение частиц обновляются в зависимости от эволюционного состояния соответственно.

Для оценки предложенного алгоритма RODDPSO используются восемь известных эталонных функций. Чтобы еще больше проиллюстрировать потенциал приложения, алгоритм RODDPSO применяется в задаче кластеризации пациентов для анализа данных в местном отделении неотложной помощи в Западном Лондоне. Результаты эксперимента показывают, что метод кластеризации на основе RODDPSO превосходит два других известных алгоритма кластеризации.

Заключение

В заключение можно сказать, что PSO продемонстрировал преимущество во многих областях применения по сравнению со многими другими алгоритмами оптимизации. Некоторые недостатки, выявленные в алгоритме, были уменьшены различными модификациями канонического ω -PSO. Данный метод оптимизации является универсальным, что еще больше подчеркивает его эффективность. Такая гибкость сделала PSO надежным оптимизатором в самых разных, но сложных сценариях оптимизации.

Конкурирующие интересы. Конфликтов интересов в отношении авторства и публикации нет.

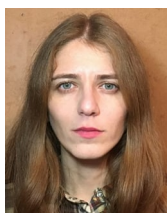
Авторский вклад и ответственность. Автор участвовал в написании статьи и полностью несет ответственность за предоставление окончательной версии статьи в печать.


Список литературы

1. Eberhart R., Kennedy J. Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, 4, *IEEE*, 1995 DOI: 10.1109/ICNN.1995.488968, pp. 1942–1948.
2. Eberhart R., Kennedy J. A new optimizer using particle swarm theory, *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, *Ieee*, 1995 DOI: 10.1109/MHS.1995.494215, pp. 39-43.
3. Cleghorn C.W., Engelbrecht A.P. Particle swarm convergence: an empirical investigation, *2014 IEEE Congress on Evolutionary Computation (CEC)*, *IEEE*, 2014 DOI: 10.1007/978-3-319-09952-112, pp. 2524-2530.
4. Banks A., Vincent J., Anyakoha C. A review of particle swarm optimization, *Part I: background and development*, *Nat. Comput.*, 2007 DOI: 10.1007/s11047-007-9049-5. vol. 4, no. 6, pp. 467-484.
5. Карпенко А. П., Селиверстов Е. Ю. Обзор методов роя частиц для задачи глобальной оптимизации (Particle Swarm Optimization), *Машиностроение и компьютерные технологии*, 2009. №3, С. 2.
6. Houssein E.H., Saad M.R., Hashim F.A., Shaban H., Hassaballah M. Levy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems, *Eng. Appl. Artif. Intell.*, 2020 DOI: 10.1016/j.engappai.2020.103731. vol. 94, pp. 103731.
7. Cazzaniga P., Nobile M.S., Besozzi D. The impact of particles initialization in PSO: parameter estimation as a case in point, *2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, *IEEE*, 2015 DOI:10.1109/CIBCB.2015.7300288. vol. 94, pp. 1-8.
8. Farooq M.U., Ahmad A., Hameed A. Opposition-based initialization and a modified pattern for inertia weight (IW) in PSO, *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, *IEEE*, 2017 DOI: 10.1109/INISTA.2017.8001139, pp. 96-101.
9. Djellali H., Ghoualmi N. Improved chaotic initialization of particle swarm applied to feature selection, *2019 International Conference on Networking and Advanced Systems (ICNAS)*, *IEEE*, 2019 DOI: 10.1109/ICNAS.2019.8807837, pp. 1-5.
10. Li Q., Liu S.-Y., Yang X.-S. Influence of initialization on the performance of metaheuristic optimizers, *Appl. Soft Comput.*, 2020 DOI: 10.1016/j.asoc.2020.106193, pp. 106193.
11. Liang X., Li W., Zhang Y., Zhong Y., Zhou M. Recent advances in particle swarm optimization via population structuring and individual behavior control, *2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 2013 DOI: 10.1109/ICNSC.2013.6548790, pp. 503-508.
12. Engelbrecht A. Particle swarm optimization: velocity initialization, *2012 IEEE Congress on Evolutionary Computation*, *IEEE*, 2012 DOI: 10.1109/CEC.2012.6256112, pp. 1-8.
13. Gunasundari S., Janakiraman S., Meenambal S. Velocity bounded boolean particle swarm optimization for improved feature selection in liver and kidney disease diagnosis, *Expert Syst. Appl.*, 2016 DOI: 10.1016/j.eswa.2016.02.042Get. vol. 56, pp. 28-47.

14. Marandi A., Afshinmanesh F., Shahabadi M., Bahrami F. Boolean particle swarm optimization and its application to the design of a dual-band dual-polarized planar antenna, *2006 IEEE International Conference on Evolutionary Computation, IEEE*, 2006 DOI: 10.1109/CEC.2006.1688716, pp. 3212-3218.
15. Shi Y., Eberhart R.C. Parameter selection in particle swarm optimization, in: International conference on evolutionary programming, *Springer*, 1998 DOI:10.1007/BFb0040810, pp. 591-600.
16. Qu B. Y., Suganthan P. N., Das S. A distance-based locally informed particle swarm model for multimodal optimization, *IEEE Trans. Evol. Comput.*, 2012 DOI: 10.1109/TEVC.2012.2203138. vol. 17, no. 3, pp. 387-402.
17. Shi Y., Liu H., Gao L., Zhang G. Cellular particle swarm optimization, *Inf. Sci.*, 2011 DOI: 10.1016/j.ins.2010.05.025. vol. 181, no. 20, pp. 4460-4493.
18. Alba E., Talbi E., Luque G., Melab N. Meta-heuristics and parallelism, *In book: Parallel Meta-heuristics: A New Class of Algorithms*, 2005, pp. 79-103 DOI: 10.1002/0471739383.ch4.
19. Houssein E.H. et al. Major advances in particle swarm optimization: theory, analysis, and application, *Swarm and Evolutionary Computation*, 2021 DOI:10.1016/j.swevo.2021.100868. vol. 63, pp. 100868.
20. Hu X., Eberhart R.C. Adaptive particle swarm optimization: detection and response to dynamic systems, *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, 2, *IEEE*, 2002 DOI: 10.1109/CEC.2002.1004492, pp. 1666-1670.
21. Xie X.-F., Zhang W.-J., Yang Z.-L. Adaptive particle swarm optimization on individual level, *6th International Conference on Signal Processing, 2002, 2, IEEE*, 2002 DOI: 10.1109/ICOSP.2002.1180009, pp. 1215-1218.
22. Zhan Z.-H., Zhang J., Li Y., Chung H.S.-H. Adaptive particle swarm optimization, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2009 DOI: 10.1016/j.engappai.2020.103731. vol. 39, no. 6, pp. 1362-1381.
23. Ермаков В.С. Метод роя частиц с адаптивными социальной и когнитивной компонентами, *Моделирование, оптимизация и информационные технологии*, 2019 DOI: 10.26102/2310-6018/2019.26.3.006. Т. 7, №3, С. 6.
24. Sun J., Feng B., Xu W. Particle swarm optimization with particles having quantum behavior, *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, 1, *IEEE*, 2004 DOI: 10.1109/CEC.2004.1330875, pp. 325-331.
25. Qian Q., Wu J., Wang Z. Optimal path planning for two-wheeled self-balancing vehicle pendulum robot based on quantum-behaved particle swarm optimization algorithm, *Pers. Ubiquitous Comput*, 2019 DOI:10.1007/s00779-019-01216-1. vol. 23, no. 3-4, pp. 393-403.
26. Lalwani S., Sharma H., Satapathy S. C., Deep K. , Bansal J. C. A survey on parallel particle swarm optimization algorithms, *Arab. J. Sci. Eng.*, 2019 DOI:10.1007/s13369-018-03713-6. Т. 44, №4, С. 2899-2923.
27. Gies D., Rahmat-Samii Y. Reconfigurable array design using parallel particle swarm optimization, *IEEE Antennas and Propagation Society International Symposium. Digest. Held in conjunction with: USNC/CNC/URSI North American Radio Sci. Meeting (Cat. No. 03CH37450)*, 2003 DOI: 10.1109/APS.2003.1217429. vol. 1, pp. 177-180.
28. Baskar S., Suganthan P. N. A novel concurrent particle swarm optimization, *Proceedings of the 2004 Congress on Evolutionary Computation*, 2004 DOI: 10.1109/CEC.2004.1330940. vol. 1, pp. 792-796.
29. Angeline P. J. Using selection to improve particle swarm optimization, in: *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*, 1998, pp. pp. 84-89 DOI: 10.1109/ICEC.1998.699327.
30. Higashi N., Iba H. Particle swarm optimization with gaussian mutation, *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03*, 2003 10.1109/SIS.2003.1202250, pp. 72-79.
31. Løvbjerg M., Rasmussen T.K., Krink T. Hybrid particle swarm optimiser with breeding and subpopulations, *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc.*, 2001, pp. 469-476.
32. Miranda V., Fonseca N. EPSO-best-of-two-worlds meta-heuristic applied to power system problems, *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*, 2002 10.1109/CEC.2002.1004393. vol. 2, pp. 1080-1085.

33. Yang B., Chen Y., Zhao Z. A hybrid evolutionary algorithm by combination of PSO and GA for unconstrained and constrained optimization problems, *International Conference on Control and Automation, IEEE*, 2007 10.1109/ICCA.2007.4376340, pp. 166–170.
34. Robinson J., Sinton S., Rahmat-Samii Y. Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna, *IEEE Antennas and Propagation Society International Symposium*, 2002 10.1109/APS.2002.1016311. vol. 1, pp. 314–317.
35. Королев С. А., Майков Д. В. Модификация алгоритма роя частиц на основе метода анализа иерархий, *Вестник ВГУ. Серия: Системный анализ и информационные технологии*, 2019 DOI: 10.17308/sait.2019.4/2679. №4, С. 36-46.
36. Yang G., Chen D., Zhou G. A new hybrid algorithm of particle swarm optimization, *International Conference on Intelligent Computing, Springer*, 2006 DOI: 10.1007/118161026, pp. 50-60.
37. Javidrad F., Nazari M. A new hybrid particle swarm and simulated annealing stochastic optimization method, *Appl. Soft Comput.*, 2017 DOI: 10.1016/j.asoc.2017.07.023. vol. 60, pp. 634–654.
38. Villarrubia G., De Paz J.F., Chamoso P., De la Prieta F. Artificial neural networks used in optimization problems, *Neurocomputing*, 2018 DOI: 10.1016/j.neucom.2017.04.075. vol. 272, pp. 10–16.
39. Eberhart R.C., Hu X. Human tremor analysis using particle swarm optimization, *Proceedings of the 1999 congress on evolutionary computation-CEC99*, 1999 DOI:10.1109/CEC.1999.785508. vol. 3, pp. 1927–1930.
40. Hamada M., Hassan M. Artificial neural networks and particle swarm optimization algorithms for preference prediction in multicriteria recommender systems, *Informatics, 5, Multidisciplinary Digital Publishing Institute*, 2018 DOI: 10.3390/informatics5020025. vol. 5, no. 2, pp. 25.
41. Zeng N., et al. A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer's disease, *Neurocomp.*, 2018 DOI: 10.1016/j.neucom.2018.09.001. vol. 320, pp. 195-202.
42. Huang K.-W., Chen J.-L., Yang C.-S., Tsai C.-W. A memetic particle swarm optimization algorithm for solving the dna fragment assembly problem, *Neural Comput. Appl.*, 2015 DOI:10.1007/s00521-014-1659-0. vol. 26, no. 3, pp. 495-506.
43. Карпенко А. П., Матвеева К. О., Буланов В. А. Решение задачи молекулярного докинга модифицированным методом роя частиц, *Машиностроение и компьютерные технологии*, 2014 DOI: 10.7463/0414.0707258. №4, С. 339-353.
44. Katarya R., Verma O.P. Efficient music recommender system using context graph and particle swarm, *Multimed. Tools Appl.*, 2018 DOI:10.1007/s11042-017-4447-x. vol. 77, no. 2, pp. 2673–2687.
45. Манусов В. З., Матренин П. В., Насрулло Х. Применение алгоритмов роевого интеллекта в управлении генерирующим потребителем с возобновляемыми источниками энергии, *Сист. анал. и обработ. данных*, 2019 DOI: 10.17212/1814-1196-2019-3-115-134. Т. 76, №3, С. 115-134.
46. Гадасин Д.В., Смальков Н.А., Кузин И.А. Использование метода роя частиц для балансировки нагрузки в сетях Интернета вещей, *Системы синхронизации, формирования и обработки сигналов*, 2022. Т. 13, №2, С. 17-23.
47. Эль Хатиб С. А., Скобцов Ю. А., Родзин С. И. Гиперэвристический роевый метод сегментации медицинских изображений, *Информатизация и связь*, 2021 DOI: 10.34219/2078-8320-2021-12-2-22-29. №2, С. 22-29.
48. Частикова В. А., Власов К. А., Картамышев Д. А. Обнаружение DDoS-атак на основе нейронных сетей с применением метода роя частиц в качестве алгоритма обучения, *Фундаментальные исследования*, 2014. Т. 4, №8, С. 829-832.
49. Javan Salehi M., Shourian M. Comparative Application of Model Predictive Control and Particle Swarm Optimization in Optimum Operation of a Large-Scale Water Transfer System, *Water Resour Manage*, 2021 DOI:10.1007/s11269-020-02755-6. Т. 35, С. 707-727.
50. Liu W., Wang Z., Liu X., Zeng N., Bell D. A novel particle swarm optimization approach for patient clustering from emergency departments., *IEEE Transactions on Evolutionary Computation*, 2018 DOI:10.1109/TEVC.2018.2878536. Т. 23, №4, С. 632-644.



Казакова Елена Мусовна ✉ – младший научный сотрудник отдела Нейроинформатики и машинного обучения, Институт прикладной математики и автоматизации, Кабардино-Балкарская Республика, г. Нальчик, Россия,  ORCID 0000-0002-5819-9396.

A Concise Overview of Particle Swarm Optimization Methods


E. M. Kazakova

Institute of Applied Mathematics and Automation KBSC RAS,
360000, Nalchik, Shortanova st., 89a, Russia

E-mail: shogenovae@inbox.ru


Particle Swarm Optimization (PSO) is a meta-heuristic method of global, inferred, proposed by Kennedy and Eberhart in 1995. It is currently one of the most commonly used search methods. This review provides a brief overview of PSO research in recent years - swarm and rate initialization methods in PSO, modifications, neighborhood topologies, hybridization, and an overview of various PSO applications.

Key words: optimization, particle swarm optimization, meta-heuristic algorithm.

 DOI: 10.26117/2079-6641-2022-39-2-156-180

Original article submitted: 07.07.2022

Revision submitted: 25.08.2022

For citation. Kazakova E. M. A Concise Overview of Particle Swarm Optimization Methods. *Vestnik KRAUNC. Fiz.-mat. nauki.* 2022, **39**: 2, 156-180.  DOI: 10.26117/2079-6641-2022-39-2-156-180

Competing interests. The authors declare that there are no conflicts of interest regarding authorship and publication.

Contribution and Responsibility. All authors contributed to this article. Authors are solely responsible for providing the final version of the article in print. The final version of the manuscript was approved by all authors.

The content is published under the terms of the Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/deed.ru>)

© Kazakova E. M., 2022

References

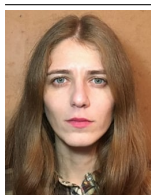
- [1] Eberhart R., Kennedy J. Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, 4, IEEE, 1995, pp. 1942–1948. DOI: 10.1109/ICNN.1995.488968.
- [2] Eberhart R., Kennedy J. A new optimizer using particle swarm theory, MHS'95. Proceedings of the Sixth International Symp. on Micro Machine and Human Sci., Ieee, 1995, pp. 39-43. DOI:10.1109/MHS.1995.494215.


Funding. The work was done without financial support

- [3] Cleghorn C. W., Engelbrecht A. P. Particle swarm convergence: an empirical investigation, 2014 IEEE Congress on Evolut. Comput. (CEC), IEEE, 2014, pp. 2524-2530. DOI: 10.1007/978-3-319-09952-112.
- [4] Banks A., Vincent J., Anyakoha C. A review of particle swarm optimization, Part I: background and development, *Nat. Comput.*, 2007, vol. 4, no. 6, pp. 467-484. DOI: 10.1007/s11047-007-9049-5
- [5] Karpenko A. P., Seliverstov E. Ju. A review of particle swarm methods for the global optimization problem, *Mashinost. i komp'yuternye tehnol.*, 2009, no. 3, pp. 2. (In Russian).
- [6] Houssein E. H., Saad M. R., Hashim F. A., Shaban H., Hassaballah M. Levy flight distribution: a new metaheuristic algorithm for solving engineering optimization problems, *Eng. Appl. Artif. Intell.*, 2020, vol. 94, pp. 103731. DOI: 10.1016/j.engappai.2020.103731.
- [7] Cazzaniga P., Nobile M. S., Besozzi D. The impact of particles initialization in PSO: parameter estimation as a case in point, 2015, IEEE Conferen. on Comput. Intellig. in Bioinform. and Comp. Biology (CIBCB), 2015, vol. 94, pp. 1-8. DOI:10.1109/CIBCB.2015.7300288.
- [8] Farooq M. U., Ahmad A., Hameed A. Opposition-based initialization and a modified pattern for inertia weight (IW) in PSO, 2017 IEEE International Confer. on Innovat. in Intel. Sys. and Appl. (INISTA), 2017, pp. 96-101. DOI: 10.1109/INISTA.2017.8001139.
- [9] Djellali H., Ghoulmi N. Improved chaotic initialization of particle swarm applied to feature selection, 2019 Intern. Conf. on Network. and Advanc. Sys. (ICNAS), IEEE, 2019, pp. 1-5. DOI: 10.1109/ICNAS.2019.8807837.
- [10] Li Q., Liu S.-Y., Yang X.-S. Influence of initialization on the performance of metaheuristic optimizers, *Appl. Soft Comput.*, 2020, pp. 106193. DOI: 10.1016/j.asoc.2020.106193.
- [11] Liang X., et al. Recent advances in particle swarm optimization via population structuring and individual behavior control, 2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC), 2013, pp. 503-508. DOI: 10.1109/ICNSC.2013.6548790.
- [12] Engelbrecht A. Particle swarm optimization: velocity initialization, 2012 IEEE Congress on Evolutionary Computation, IEEE, 2012, pp. 1-8. DOI: 10.1109/CEC.2012.6256112.
- [13] Gunasundari S., Janakiraman S., Meenambal S. Velocity bounded boolean particle swarm optimization for improved feature selection in liver and kidney disease diagnosis, *Expert Syst. Appl.*, 2016, vol. 56, pp. 28-47. DOI: 10.1016/j.eswa.2016.02.042Get
- [14] Marandi A., et al. Boolean particle swarm optimization and its application to the design of a dual-band dual-polarized planar antenna, 2006 IEEE International Conference on Evolutionary Computation, IEEE, 2006, pp. 3212-3218. DOI: 10.1109/CEC.2006.1688716
- [15] Shi Y., Eberhart R. C. Parameter selection in particle swarm optimization, in: International conference on evolutionary programming, 1998, pp. 591-600. DOI:10.1007/BFb0040810.
- [16] Qu B. Y., Suganthan P. N., Das S. A distance-based locally informed particle swarm model for multimodal optimization, *IEEE Trans. Evol. Comput.*, 2012, vol. 17, no. 3, pp. 387-402. DOI: 10.1109/TEVC.2012.2203138.
- [17] Shi Y., Liu H., Gao L., Zhang G. Cellular particle swarm optimization, *Inf. Sci.*, 2011, vol. 181, No 20, pp. 4460-4493. DOI: 10.1016/j.ins.2010.05.025.
- [18] Alba E., et al. Meta-heuristics and parallelism, In book: *Parallel Metaheuristics: A New Class of Algorithms*, 2005, pp. 79-103 DOI: 10.1002/0471739383.ch4.
- [19] Houssein E. H., et al. Major advances in particle swarm optimization: theory, analysis, and application, *Swarm and Evol. Comp.*, 2021, vol. 63, pp. 100868. DOI:10.1016/j.swevo.2021.100868.
- [20] Hu X., Eberhart R. C. Adaptive particle swarm optimization: detection and response to dynamic systems, *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02*, 2, IEEE, 2002, pp. 1666-1670. DOI: 10.1109/CEC.2002.1004492.

- [21] Xie X.-F., Zhang W.-J., Yang Z.-L. Adaptive particle swarm optimization on individual level, 6th International Conference on Signal Processing, 2002, 2, IEEE, 2002, pp. 1215–1218. DOI: 10.1109/ICOSP.2002.1180009.
- [22] Zhan Z.-H., Zhang J., Li Y., Chung H.S.-H. Adaptive particle swarm optimization, IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 2009, vol. 39, no. 6, pp. 1362–1381. DOI: 10.1016/j.engappai.2020.103731.
- [23] Ermakov B. S. Particle swarm method with adaptive social and cognitive components, Modelirovanie, optimizacija i informacionnye tehnologii, 2019, vol. 7, no. 3, pp. 6. DOI:10.26102/2310-6018/2019.26.3.006 (In Russian).
- [24] Sun J., Feng B., Xu W. Particle swarm optimization with particles having quantum behavior, Proceedings of the 2004 Congress on Evolutionary Computation, 1, IEEE, 2004, pp. 325–331. DOI: 10.1109/CEC.2004.1330875.
- [25] Qian Q., Wu J., Wang Z. Optimal path planning for two-wheeled self-balancing vehicle pendulum robot based on quantum-behaved particle swarm optimization algorithm, Pers. Ubiquitous Comput, 2019, vol. 23, no. 3-4, pp. 393–403. DOI:10.1007/s00779-019-01216-1.
- [26] Lalwani S., Sharma H., Satapathy S. C., Deep K., Bansal J. C. A survey on parallel particle swarm optimization algorithms, Arab. J. Sci. Eng., 2019, vol. 44, no. 4, pp. 2899–2923. DOI:10.1007/s13369-018-03713-6.
- [27] Gies D., Rahmat-Samii Y. Reconfigurable array design using parallel particle swarm optimization, IEEE Antennas and Propagation Society International Symposium. Digest. Held in conjunction with: USNC/CNC/URSI North American Radio Sci. Meeting, 2003, vol. 1, pp. 177–180. DOI: 10.1109/APS.2003.1217429.
- [28] Baskar S., Suganthan P. N. A novel concurrent particle swarm optimization, Proc. of the 2004 Congr. on Evol. Comp., 2004, vol. 1, pp. 792–796. DOI: 10.1109/CEC.2004.1330940.
- [29] Angeline P. J. Using selection to improve particle swarm optimization, in: 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence, 1998, pp. 84–89. DOI: 10.1109/ICEC.1998.699327
- [30] Higashi N., Iba H. Particle swarm optimization with gaussian mutation, Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03, 2003 10.1109/SIS.2003.1202250, pp. 72–79. DOI: 10.1109/SIS.2003.1202250.
- [31] Lovbjerg M., Rasmussen T.K., Krink T. Hybrid particle swarm optimiser with breeding and subpopulations, Proc. of the 3rd Ann. Conf. on Gen. and Evolut. Comput., 2001, pp. 469–476.
- [32] Miranda V., Fonseca N. EPSO-best-of-two-worlds meta-heuristic applied to power system problems, Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02, 2002, vol. 2, pp. 1080–1085. DOI: 10.1109/CEC.2002.1004393
- [33] Yang B., Chen Y., Zhao Z. A hybrid evolutionary algorithm by combination of PSO and GA for unconstrained and constrained optimization problems, International Conference on Control and Automation, IEEE, 2007, pp. 166–170. DOI: 10.1109/ICCA.2007.4376340.
- [34] Robinson J., Sinton S., Rahmat-Samii Y. Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna, IEEE Anten. and Propagat. Society Internat. Symp., 2002, vol. 1, pp. 314–317. DOI: 10.1109/APS.2002.1016311.
- [35] Korolev S. A., Majkov D. V. Modification of the particle swarm algorithm based on the hierarchy analysis method, Vestnik VGU. Serija: Sistem. anal. i inform. tehnol., 2019, no. 4, pp. 36-46. DOI: 10.17308/sait.2019.4/2679 (in Russian).
- [36] Yang G., Chen D., Zhou G. A new hybrid algorithm of particle swarm optimization, Internat. Confer. on Intelli. Comp., 2006, pp. 50-60. DOI: 10.1007/118161026

- [37] Javidrad F., Nazari M. A new hybrid particle swarm and simulated annealing stochastic optimization method, *Appl. Soft Comput.*, 2017, vol. 60, pp. 634–654. DOI: 10.1016/j.asoc.2017.07.023.
- [38] Villarrubia G., De Paz J.F., Chamoso P., De la Prieta F. Artificial neural networks used in optimization problems, *Neurocomp.*, 2018, vol. 272, pp. 10–16. DOI: 10.1016/j.neucom.2017.04.075.
- [39] Eberhart R.C., Hu X. Human tremor analysis using particle swarm optimization, *Proceedings of the 1999 congress on evolutionary computation-CEC99*, 1999, vol. 3, pp. 1927–1930. DOI:10.1109/CEC.1999.785508
- [40] Hamada M., Hassan M. Artificial neural networks and particle swarm optimization algorithms for preference prediction in multicriteria recommender systems, *Inform.*, 2018, vol. 5, no. 2, pp. 25. DOI: 10.3390/informatics5020025
- [41] Zeng N., et al. A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer's disease, *Neurocomp.*, 2018, vol. 320, pp. 195-202. DOI:10.1016/j.neucom.2018.09.001.
- [42] Huang K.W., Chen J.L., Yang C.S., Tsai C.W. A memetic particle swarm optimization algorithm for solving the dna fragment assembly problem, *Neural Comput. Appl.*, 2015, vol. 26, no. 3, pp. 495-506. DOI:10.1007/s00521-014-1659-0.
- [43] Karpenko A.P., Matveeva K.O., Bulanov V.A. Solving the problem of molecular docking by a modified particle swarm method, *Mashinostroenie i komp'yuternye tehnologii*, 2014, no. 4, pp. 339-353. DOI: 10.7463/0414.0707258 (In Russian).
- [44] Katarya R., Verma O.P. Efficient music recommender system using context graph and particle swarm, *Multimed. Tools Appl.*, 2018, vol. 77, no. 2, pp. 2673–2687. DOI:10.1007/s11042-017-4447-x
- [45] Manusov V.Z., Matrenin P.V., Nasrullo H. Application of swarm intelligence algorithms in the management of a generating consumer with renewable energy sources, *Sistem. anal. i obrabot. dannyh*, 2019, vol. 76, no. 3, pp. 115-134. DOI: 10.17212/1814-1196-2019-3-115-134 (In Russian).
- [46] Gadasin D.V., Smal'kov N.A., Kuzin I.A. Using the Particle Swarm Method for Load Balancing in Internet of Things Networks, *Sistemy sinhronizacii, formirovaniya i obrabotki signalov*, 2022, vol. 13, no. 2, pp. 17-23. (In Russian).
- [47] El-Khatib S., Skobtsov Y.A., Rodzin S.I. Hyper heuristic particle swarm optimization method for medical images segmentation, *Informatizaciya i svyaz'*, 2021, no. 2, pp. 22-29. DOI: 10.34219/2078-8320-2021-12-2-22-29(In Russian).
- [48] Chastikova V.A., Vlasov K.A., Kartamyshev D.A. Obnaruzhenie DDoS-atak na osnove nejronnyh setej s primeneniem metoda roya chastic v kachestve algoritma obucheniya, *Informatizaciya i svyaz'*, 2014, vol. 4, no. 8, pp. 829-832. (In Russian).
- [49] Javan S. M., Shourian M. Comparative Application of Model Predictive Control and Particle Swarm Optimization in Optimum Operation of a Large-Scale Water Transfer System. *Water Resour Manage*, 2021, vol. 35, pp. 707–727 DOI: 10.1007/s11269-020-02755-6
- [50] Liu W., et al. A novel particle swarm optimization approach for patient clustering from emergency departments, *IEEE Transactions on Evolutionary Computation*, 2018, vol. 23, no. 4, pp. 632-644. DOI: I10.1109/TEVC.2018.2878536



Kazakova Elena Musovna ✉ – Junior Researcher of the Department of Neural Networks and Machine Learning, Institute of Applied Mathematics and Automation, Kabardino-Balkarian Republic, Nalchik, Russia,  ORCID 0000-0002-5819-9396.