

Далее в диссертации описаны перечисленные алгоритмы, а также результаты проведенных исследований. Для описания работы методов предположим, что необходимо решить задачу безусловной оптимизации с вещественными переменными.

1.1. Метод роя частиц (Particle Swarm Optimization, PSO)

Стайный алгоритм – метод численной оптимизации, базирующийся на моделировании поведения популяции частиц в пространстве оптимизации, для использования которого не требуется знать точного градиента оптимизируемой функции [101]. Данный метод привлекателен простотой реализации, он может использоваться для решения многих задач, включая обучение нейронных сетей.

Первоначально PSO был создан для задач с вещественными переменными. Пусть $f(x)$ целевая функция, которую необходимо минимизировать, причем $x = (x_1, \dots, x_d, \dots, x_D)$. Работа алгоритма начинается с создания популяции частиц (то есть множества потенциальных решений x_i , $i = \overline{1, N}$, где N – размер популяции) случайным образом. Частицы представляют собой вектор координат точки в пространстве оптимизации (вещественных чисел) $x_i = (x_{i1}, \dots, x_{id}, \dots, x_{iD})$, $i = \overline{1, N}$. Каждая частица передвигается по поверхности графика функции с какой-то скоростью $v_i = (v_{i1}, \dots, v_{id}, \dots, v_{iD})$, $i = \overline{1, N}$. Частицы изменяют свою скорость и координаты, основываясь на собственном опыте и опыте других частиц. Таким образом, моделируется многоагентная система, где агенты-частицы двигаются к оптимальным решениям, обмениваясь при этом информацией с соседями. Скорость и координаты частиц обновляются по следующим формулам:

$$v_{id}^{t+1} = \omega \cdot v_{id}^t + c_1 \cdot rand() \cdot (p_{id} - x_{id}^t) + c_2 \cdot Rand() \cdot (p_{gd} - x_{id}^t),$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}.$$

В этих формулах v_{id}^{t+1} , v_{id}^t – скорости частицы на $(t+1)$ -ой и t -ой итерациях соответственно, x_{id}^{t+1} , x_{id}^t – координаты частицы на $(t+1)$ -ой и t -ой итерациях соответственно, p_{id} – лучшая позиция, найденная i -ой частицей за t предыдущих

поколений, p_{gd} – лучшая позиция, найденная всей стаей за все время работы алгоритма, $rand()$ и $Rand()$ – случайные числа из отрезка $[0;1]$, c_1 , c_2 – коэффициенты обучения из отрезка $[0;2]$, ω – инерционный вес.

Константа c_1 называется когнитивным (познавательным) параметром, она позволяет учитывать «собственный опыт» (историю) частицы. Константа c_2 называется социальным параметром и позволяет частице учитывать «опыт всей стаи». Таким образом, c_2 управляет воздействием глобального лучшего положения, а c_1 управляет воздействием личного лучшего положения на скорость каждой частицы.

Работу алгоритма также определяет топология соседства частиц, а именно, каким образом выбирается лучший индивид для каждой частицы. Наиболее известными являются следующие топологии: кольцо, клика, двумерный тор, кластер [112].

Итак, идея алгоритма заключается в том, что частицы, которые сначала равномерно распределены по поверхности отклика функции, с течением времени (от поколения к поколению) начинают группироваться («сбиваться в стаи») около локальных минимумов, причем наибольшая стая собирается около глобального оптимума. При этом почти всегда имеются частицы, находящиеся в стороне от таких стай, а также частицы, выскакивающие за границы допустимой области.

В настоящее время существуют различные модификации стайного алгоритма оптимизации, разработанные для решения различных задач оптимизации. Одна из первых модификаций была предложена Кеннеди и Эберхартом в 1997 году для решения задач однокритериальной безусловной оптимизации с бинарными переменными [102]. В 1998 году Ши и Эберхарт опубликовали работу, в которой описывалась методика автоматизированной настройки параметров c_1 , c_2 , ω алгоритма [127]. В 1999 году в статье Кеннеди [100] рассматривалось влияние выбранной топологии на результат работы алгоритма при решении той или иной задачи оптимизации. В том же году была впервые предложена модификация алгоритма для решения задач многокритериальной оптимизации [117]. В последующие годы были разработаны

различные методы на основе роевого алгоритма, например, [138], [136]. Также стайный алгоритм применялся для обучения нейронных сетей [82] и решения различных практических задач, например, в работах [80], [89].

Ниже приведена схема стайного алгоритма оптимизации.

Particle Swarm Optimization

инициализация популяции $P = \{x_{i1}, \dots, x_{id}, \dots, x_{iD}\}, i = 1, \dots, N\}$ случайным образом

для каждой частицы x_i из популяции P

$$p_{id} = x_{id}, d = 1, \dots, D$$

конец цикла

для каждой частицы x_i из популяции P

$$v_{id} = 0, d = 1, \dots, D$$

конец цикла

определение $p_{gd}, d = 1, \dots, D$

пока не выполнится критерий остановки

для каждой частицы x_i из популяции P

если $f(x_i) < f(p_i)$

$$p_{id} = x_{id}, d = 1, \dots, D$$

конец цикла

обновить значение $p_{gd}, d = 1, \dots, D$

конец цикла

В ходе выполнения диссертационной работы было проведено исследование эффективности метода роя частиц, результаты опубликованы в статьях [6], [9] и [12]. В статье [15] представлены результаты исследования эффективности эвристики PSO для решения задач оптимизации с бинарными переменными, а также ее практическое приложение. Разработка, реализация и исследование эффективности метода роя частиц, модифицированного для решения задач условной оптимизации, были темой публикаций [7], [5] и [13]. Разработка, реализация, а также исследование эффективности распараллеленного стайного алгоритма продемонстрированы в работе [10]. Кроме того, в статьях [4], [11], [16] и [14] можно найти описание решения задачи формирования оптимального инвестиционного портфеля предприятия с помощью различных разработанных модификаций алгоритма PSO. В монографии [25] представлено подробное описание всех исследований, связанных со стайным алгоритмом, описание всех реализованных его модификаций, а также представлены все полученные

результаты (как при решении тестовых задач, так и при решении практических). Краткое описание проведенных исследований опубликовано в работе [21].

1.2. Алгоритм поиска стай волков (Wolf Pack Search, WPS)

Алгоритм поиска стай волков – метаэвристический алгоритм, разработанный в 2007 году, идея которого почерпнута из социального поведения стай волков [75].

Для волков типичен семейный образ жизни: они живут стаями – семейными группами, состоящими из пары вожаков, их родственников, а также пришлых одиноких волков. Внутри стаи наблюдается строго обозначенная иерархия, на вершине которой находится вожак стаи, направляющий остальных особей на поиск добычи. Волки «исследуют» местность на наличие добычи, когда кто-то из них учует запах жертвы, начинается ее поиск. Чем сильнее ощущается запах, тем ближе волки к жертве. Таким образом, они перемещаются в направлении усиления запаха добычи. Причем волки разделяются на небольшие группы, и каждая группа осуществляет поиск в каком-то определенном направлении, отличном от направлений других групп. В итоге, когда один из волков найдет жертву, он подает сигнал вожаку и остальным, чтобы поделиться добычей с волками из стаи.

Метод поиска стай волков копирует процесс их охоты. Предположим местность, на которой волки охотятся – это поисковая область в смысле оптимизации, а частицы – это волки. Пусть изначально сгенерировано N «волков» в евклидовом пространстве размерности D , то есть каждый волк представлен в виде вектора $x_i = (x_1, \dots, x_D)$, определяющего его координаты в пространстве. Таким образом, стая (популяция) представляет собой множество потенциальных решений, координаты которых так же, как и для стайного алгоритма оптимизации, обновляются на каждой итерации, пока не будет найдено оптимальное решение или не будет выполнено максимально заданное количество вычислений целевой функций.