Гаецький Богдан Богданович, Курсант
КВВУ им. Генерала-армии С.М. Штеменко

Скворцов Дмитрий Юрьевич, Курсант
КВВУ им. Генерала-армии С.М. Штеменко

Акишин Андрей Владимирович, к.т.н.
КВВУ им. Генерала-армии С.М. Штеменко

АДАПТИВНЫЙ ВАРИАНТ АЛГОРИТМА РОЯ ЧАСТИЦ ДЛЯ ОПТИМИЗАЦИИ ЗАДАЧ С МНОЖЕСТВЕННЫМИ ОГРАНИЧЕНИЯМИ

Аннотация. В статье рассматривается применение алгоритма оптимизации роем частиц (PSO) к задачам с множественными ограничениями смешанного типа. Предложен адаптивный вариант алгоритма CPSO-A, в котором управление параметрами осуществляется на основе эволюционного фактора, а обработка ограничений реализована через комбинацию динамической штрафной функции и процедуры коррекции недопустимых решений. Приведены результаты сравнительного анализа с классическим PSO, генетическим алгоритмом и методом имитации отжига.

Ключевые слова: Алгоритм роя частиц, PSO, метаэвристика, оптимизация с ограничениями, адаптивный алгоритм, эволюционный фактор.

1. Введение

Задачи оптимизации с множественными ограничениями встречаются в проектировании технических систем, управлении ресурсами, конфигурировании защитных комплексов и ряде других прикладных областей. Одновременное присутствие нелинейных, целочисленных ограничений и ограничений на совместимость переменных исключает применение классических методов математического программирования и делает задачу вычислительно сложной.

Алгоритм оптимизации роем частиц (Particle Swarm Optimization, PSO), предложенный Кеннеди и Эберхартом в 1995 году, продемонстрировал высокую эффективность на широком классе задач нелинейной оптимизации [1]. Вместе с тем в задачах с плотными ограничениями стандартный PSO сталкивается с серьёзными трудностями: значительная часть популяции оказывается в недопустимой области, что замедляет сходимость и снижает качество найденных решений. В настоящей статье предлагается адаптивный вариант алгоритма CPSO-A, направленный на преодоление указанных ограничений.

2. Стандартный PSO и проблемы при наличии ограничений

В PSO рой из N частиц перемещается в D-мерном пространстве поиска. На каждой итерации t скорость и позиция каждой частицы i обновляются по формулам:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (pbest_i - x_i(t)) + c_2 \cdot r_2 \cdot (gbest - x_i(t)),$$

$$x_i(t+1) = x_i(t) + v_i(t+1),$$

где w – коэффициент инерции; c_1, c_2 – ускорительные коэффициенты; r_1, r_2 – случайные числа из [0,1]; $pbest_i$ – персонально лучшая позиция частицы i; gbest – глобально лучшая позиция всего роя [1].

В задачах с ограничениями стандартный PSO сталкивается с тремя ключевыми проблемами. Во-первых, случайная инициализация популяции нередко приводит к тому, что большинство частиц начинают поиск в недопустимой области. Во-вторых, выбор коэффициента штрафа критически влияет на поведение алгоритма: малое значение позволяет игнорировать ограничения, большое – фактически блокирует перемещение через недопустимую область. В-третьих, фиксированные параметры w, c_1, c_2 не позволяют алгоритму автоматически адаптировать стратегию поиска к текущему состоянию роя [2]. Устранение перечисленных проблем составляет основу предлагаемого алгоритма CPSO-A.

3. Алгоритм CPSO-A: адаптация параметров

Адаптация параметров в CPSO-A основана на вычислении эволюционного фактора EF – численной меры текущего состояния роя, предложенной Чжа и соавторами [3]:

$EF = (\bar{d} - d_{min}) / (d_{max} - d_{min})$,

где $\bar{d}$ – среднее расстояние от каждой частицы до глобально лучшей позиции gbest; d_{min} и d_{max} – минимальное и максимальное значения этого расстояния по всему рою. Значение $EF \in [0, 1]$: EF близкое к 1 означает, что рой рассредоточен и ведёт активное исследование пространства; EF близкое к 0 – что рой сконцентрирован вблизи gbest и находится в стадии конвергенции.

На основании значения EF состояние роя классифицируется в одну из четырёх фаз: исследование ($EF \in [0{,}75;\ 1{,}0]$), эксплуатация ($EF \in [0{,}50;\ 0{,}75)$), конвергенция ($EF \in [0{,}25;\ 0{,}50)$) и выход из локального оптимума ($EF \in [0;\ 0{,}25)$). Каждой фазе соответствует свой диапазон значений параметров: в фазе исследования w повышается до 0,9, $c_1 = 2{,}5$, $c_2 = 0{,}5$, что стимулирует широкий охват пространства; в фазе конвергенции w снижается до 0,4, $c_1 = 0{,}5$, $c_2 = 2{,}5$, что концентрирует рой вблизи найденного оптимума. Такая схема адаптации устраняет необходимость ручного подбора параметров для каждой конкретной задачи.

4. Обработка ограничений: штраф и коррекция

Для обработки ограничений в CPSO-A применяется двухуровневая стратегия. Первый уровень – адаптивная штрафная функция. Штрафная добавка к целевой функции для частицы i вычисляется как:

$P(x_i, t) = p(t) \cdot \Sigma_j \max(0, g_j(x_i))^2$,

где $g_j(x_i)$ – нарушение j-го ограничения; $p(t)$ – динамический коэффициент штрафа. На ранних итерациях $p(t)$ мало, что позволяет частицам пересекать недопустимую область и исследовать пространство решений. С ростом номера итерации $p(t)$ увеличивается, постепенно направляя поиск в допустимую область. Скорость нарастания штрафа адаптируется в зависимости от текущей доли допустимых решений в популяции: при низкой доле нарастание замедляется во избежание преждевременной блокировки исследования [4].

Второй уровень – процедура прямой коррекции – применяется к частицам, нарушающим структурные ограничения, которые не могут быть обработаны штрафным методом: ограничения совместимости бинарных переменных вида $s_i + s_j \leq 1$, обязательные переменные и ограничения на диапазон значений. Процедура выполняется после каждого обновления позиций и восстанавливает допустимость путём минимально необходимого изменения компонент вектора решения. При конфликте совместимости случайно исключается одна из двух конфликтующих переменных – с вероятностью, обратно пропорциональной её вкладу в целевую функцию.

Для задач со смешанным пространством переменных (непрерывные и бинарные компоненты) вычисление EF требует единой метрики расстояния. В CPSO-A применяется взвешенная комбинация евклидовой нормы по непрерывным компонентам и нормализованного расстояния Хэмминга по бинарным:

$d(x_i, x_j) = \gamma \cdot dE(x_i, x_j) + (1-\gamma) \cdot dH(x_i, x_j)$,

где $\gamma \in [0,1]$ – весовой коэффициент, отражающий долю непрерывных переменных в задаче. Такая метрика обеспечивает корректную классификацию состояния роя независимо от соотношения типов переменных.

5. Вычислительный эксперимент и результаты

Для оценки эффективности CPSO-A проведён вычислительный эксперимент на наборе тестовых задач с различными типами ограничений. Использовались задачи G1–G6 из стандартного набора CEC 2006 [5], характеризующиеся нелинейными ограничениями неравенства и равенства, а также три прикладных задачи: оптимизация конфигурации системы технической защиты информации, распределение ресурсов с бюджетным ограничением и задача размещения с ограничениями совместимости. Алгоритм реализован на Python 3.10; каждый из четырёх методов запускался 30 раз на каждой задаче.

В качестве методов сравнения выбраны: стандартный PSO с фиксированными параметрами ($w = 0{,}729$, $c_1 = c_2 = 1{,}494$), генетический алгоритм (GA) с турнирной селекцией и однородным кроссовером, а также метод имитации отжига (SA) с экспоненциальным графиком охлаждения. Бюджет вычислений для всех методов одинаков – 50 000 вычислений целевой функции на один запуск.

CPSO-A обеспечил наилучшее среднее качество решений на 7 из 9 тестовых задач. На задачах G1–G4 преимущество над стандартным PSO по значению целевой функции составило 12–31 %. На задачах с бинарными переменными – 18–45 %. На задачах G5–G6 с высокой плотностью активных ограничений CPSO-A уступил GA на 3–7 % по среднему качеству, однако потребовал в 1,6–2,1 раза меньше времени вычисления. Доля допустимых решений в финальной популяции составила 87–98 % против 41–73 % у стандартного PSO, что подтверждает эффективность двухуровневой стратегии обработки ограничений.

6. Заключение

В статье предложен адаптивный алгоритм CPSO-A для решения задач оптимизации с множественными ограничениями смешанного типа. Алгоритм сочетает три элемента: адаптацию параметров на основе эволюционного фактора EF, динамическую штрафную функцию с адаптивным коэффициентом и прямую коррекцию недопустимых решений. Комбинированная метрика расстояния обеспечивает корректную работу механизма адаптации в задачах со смешанным дискретно-непрерывным пространством переменных.

Вычислительный эксперимент подтвердил превосходство CPSO-A над стандартным PSO на большинстве тестовых задач и его конкурентоспособность с генетическим алгоритмом при значительно меньших вычислительных затратах. Полученные результаты открывают перспективы применения алгоритма в задачах проектирования систем защиты информации, оптимизации размещения ресурсов и других прикладных областях с множественными разнотипными ограничениями. Дальнейшие исследования целесообразно направить на автоматическую настройку весового коэффициента γ, многокритериальные постановки с построением фронта Парето, а также распараллеливание вычислений для задач высокой размерности

Список литературы:

1. Kennedy J., Eberhart R. Particle swarm optimization // Proceedings of the IEEE International Conference on Neural Networks. – 1995. – P. 1942–1948.

2. Mezura-Montes E., Coello C. A. C. Constraint-handling in nature-inspired numerical optimization: Past, present and future // Swarm and Evolutionary Computation. – 2011. – Vol. 1, № 4. – P. 173–194.

3. Zhan Z., Zhang J., Li Y., Chung H. Adaptive Particle Swarm Optimization // IEEE Transactions on Systems, Man, and Cybernetics. – 2009. – Vol. 39, № 6. – P. 1362–1381.

4. Deb K. An efficient constraint handling method for genetic algorithms // Computer Methods in Applied Mechanics and Engineering. – 2000. – Vol. 186, № 2–4. – P. 311–338.

5. Liang J. J. et al. Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization. – Nanyang Technological University, 2006. – 24 p.

6. Clerc M., Kennedy J. The particle swarm – explosion, stability, and convergence in a multidimensional complex space // IEEE Transactions on Evolutionary Computation. – 2002. – Vol. 6, № 1. – P. 58–73.

7. Coello C. A. C. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art // Computer Methods in Applied Mechanics and Engineering. – 2002. – Vol. 191, № 11–12. – P. 1245–1287