ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ
Т. 64, № 5 ФИЗИКА 2021

* *
*

УДК 53.072:681.3 DOI: 10.17223/00213411/64/5/94

YA BI[1,2], ANTHONY LAM[3], HUIQUN QUAN[1], HUI LIU[1], CUNFA WANG[4,5]

ОПТИМИЗАЦИЯ МНОЖЕСТВА ЧАСТИЦ ДЛЯ ОБЕСПЕЧЕНИЯ ИХ АКТИВНОСТИ [*]

Алгоритм оптимизации множества частиц имеет ряд недостатков, поэтому предложена стратегия комплексного улучшения, представляющая собой простую оптимизацию множества частиц с динамической адаптивной гибридизацией экстремальных возмущений и кроссов (алгоритм ecds-PSO). Предложенный новый комплексный улучшенный алгоритм множества частиц отбрасывает их скорость и уменьшает PSO от второго порядка до разностного уравнения первого порядка. Эволюционный процесс управляется только переменными положения частиц. Операция гибридизации, заключающаяся в увеличении возмущения экстремума и введении генетического алгоритма, может ускорить частицы до выхода за пределы локального экстремума. Математический вывод и множество сравнительных экспериментов подтверждают, что улучшенная оптимизация множества частиц – это простой и эффективный алгоритм оптимизации, который может повысить точность алгоритма, вязкость сходимости и способность избегать локального экстремума, а также эффективно снизить сложность расчета.

Ключевые слова: *алгоритм оптимизации множества частиц, динамическая адаптивность, локальный экстремум, активность частиц.*

Введение

Алгоритм оптимизации множества частиц (PSO) [1] возник как результат моделирования группового кормового поведения птиц и был типичным параллельным алгоритмом глобальной оптимизации. Поскольку алгоритм PSO может быть использован для крупномасштабных, многопиковых, нелинейных и недифференцируемых сложных задач, а операция проста и легка в реализации, поэтому он широко применяется.

Подобно генетическому алгоритм PSO в эволюционном процессе опирается, в основном, на две ключевые характеристики: множество и приспособленность. Каждая частица представляет собой одно возможное решение задачи, которое характеризуется параметрами положения и скорости, в то время как пригодность используется для измерения плюсов и минусов частиц. В отличие от генетического алгоритма, каждая эволюция оптимизации PSO не только зависит от ценности отдельных частиц, но и опирается на взаимосвязь и конкуренцию между множествами. Таким образом, хотя скорость сходимости велика и эффективность очень высока на ранней стадии эволюции множества частиц, но на поздней стадии скорость сходимости и точность алгоритма быстро падают и алгоритм легко попадает в локальный экстремум.

Более того, при решении многомерных экстремальных задач точность сходимости алгоритма невелика. В этой связи в данной работе предложен алгоритм «динамической самоадаптирующейся и простой оптимизации множества частиц с возмущенным экстремумом и кроссовером» путем двух стратегий совершенствования: 1) упрощения уравнения скорости и динамической адаптивной регулировки инерционного веса множества частиц; 2) введения возмущенного экстремума и кроссовера, далее именуемого ecds-PSO.

Базовый алгоритм множества частиц и его совершенствование

Алгоритм процесса базового PSO (bPSO [2]) заключается в следующем: сначала инициализируется группа случайных частиц, затем в процессе итерационной эволюции частицы обновляют свою скорость и положение в соответствии с индивидуальным экстремумом и глобальным экстремумом, пока не находится оптимальное решение. Обновленные уравнения для положения и скорости можно записать как

$$\vec{v}_i^{\,t+1} = \vec{v}_i^{\,t} + c_1 \cdot \mathrm{rand}_1(\,) \cdot (\overrightarrow{p\,\mathrm{best}}_i^{\,t} - \vec{x}_i^{\,t}) + c_2 \cdot \mathrm{rand}_2(\,) \cdot (\overrightarrow{g\,\mathrm{best}}^{\,t} - \vec{x}_i^{\,t}); \quad (1)$$

[*] Работа поддержана Национальным фондом естественных наук Китая (№ 70160376), Национальным фондом социальных наук (№ 20ZDA038) и Исследовательским центром логистического развития Хубэй, спонсируемым Проектом.

$$\vec{x}_i^{\,t+1} = \vec{x}_i^{\,t} + \vec{v}_i^{\,t+1}, \tag{2}$$

где в итерации t положение частицы i обозначено как $\vec{x}_i^{\,t}$, скорость – $\vec{v}_i^{\,t}$, индивидуальное экстремальное положение частицы – $\overrightarrow{p\,\mathrm{best}}_i^{\,t}$, глобальное экстремальное положение – $\overrightarrow{g\,\mathrm{best}}^{\,t}$.

На поздней стадии алгоритма частицы могут быть собраны в локальном экстремальном положении и не могут вырваться из него, что приводит к низкой скорости поиска или точности.

Попытки решения этих проблем можно разделить на два подхода. Первый заключается в увеличении и регулировке глобального коэффициента для повышения скорости оптимизации и точности. Например, в [3] добавили инерционный вес w к скорости в (1) с целью ускорить оптимизацию. Модифицированное уравнение выглядит так:

$$\vec{v}_i^{\,t+1} = w \cdot \vec{v}_i^{\,t} + c_1 \cdot \mathrm{rand}_1(\,) \cdot (\overrightarrow{p\mathrm{best}}_i^{\,t} - \vec{x}_i^{\,t}) + c_2 \cdot \mathrm{rand}_2(\,) \cdot (\overrightarrow{g\,\mathrm{best}}^{\,t} - \vec{x}_i^{\,t}). \tag{3}$$

В (2) добавили ограничивающий фактор к скорости [4] с целью предотвращения чрезмерной дивергенции частиц, а модифицированное уравнение выглядит как

$$\vec{x}_i^{\,t+1} = \vec{x}_i^{\,t} + \partial \vec{v}_i^{\,t+1}. \tag{4}$$

Было доказано, что большее w отклоняет множество от первоначального направления, что приводит к дивергенции поиска, в то время как меньшее w мало помогает для поиска скачка из локального экстремума. Фактор ограничения ∂ оказывает определенное влияние на оптимизацию, но в данном случае не подходит.

В [5] добавили четыре вида новых частиц в алгоритм оптимизации PSO, скорректировали режим обновления и ввели динамическую регулировку инерционного веса изменения скорости фокусированного расстояния, чтобы преодолеть случайность алгоритма и решить проблемы слепого поиска на ранней стадии и медленной скорости поиска на поздней стадии алгоритма. В [6] предложили элитный алгоритм обратного обучения PSO, основанный на возмущении, и приняли метод нелинейного уменьшения для обновления весов, что значительно улучшило производительность алгоритма по скорости сходимости и точности.

Усовершенствованный алгоритм оптимизации множества, основанный на генетическом кроссовере и стратегии мультихаоса, был предложен авторами работы [7]. Алгоритм значительно улучшил оптимизационные возможности за счет хаотической обработки инерционного веса, локального и глобального поиска. Согласно адаптивной оценке превосходного коэффициента и обновлению информации о положении глобальных оптимальных частиц, авторы [8] далее углубились в локальные решения с помощью стратегии 3-opt и применили улучшенный алгоритм оптимизации к дискретным задачам TSP. Приведенные исследования повысили скорость поиска и точность улучшенного алгоритма PSO, но идеи заключались лишь в оптимизации скоростных элементов bPSO. Такие стратегии значительно улучшили оптимальную скорость, но не решали проблему разнообразия частиц, поэтому на позднем этапе эволюции алгоритм легко попадал в локальную ловушку. Особенно в сложных задачах нелинейной оптимизации роль этих идей в повышении точности весьма ограничена [9].

Другой подход заключается в оптимизации положения частицы, а именно в динамической корректировке положения отдельных частиц. За счет увеличения разнообразия частиц повышаются их способность выскакивать из локального экстремума и точность. Например, получаем алгоритм оптимизации kPSO путем группировки и кластеризации частиц [10] и формирования на этой основе различных усовершенствованных алгоритмов. В [11] изучена топология пяти полей и обнаружено, что после введения кластеризации алгоритм может значительно повысить точность сходимости. С точки зрения топологии и адаптивного множества авторы [12] предложили адаптивный алгоритм оптимизации с гетерогенными характеристиками кластеризации и изучена способность обеспечения успеха кластеризации и чувствительности параметров. После вывода о том, что включение алгоритма в выборку признаков может повысить точность классификации [13], в [14] была использована стратегия классификации признаков с коэффициентом штрафа для управления процессом кластеризации. Было доказано, что этот метод оптимизации имеет лучший эффект кластеризации и глобальную сходимость, что очень подходит для задач уменьшения размерности для многомерных данных. Авторы [15] предложили многоцелевой алгоритм оптимизации, основанный на декомпозиции и дифференциальной эволюции, причем этот алгоритм генерировал

набор однородных векторов направления через угол направления и обеспечивал равномерность распределения частиц. С помощью неявной политики удержания «элиты» и механизма коррекции дифференциальной эволюции исключалось попадание алгоритма в локальный экстремум, а с помощью стратегии сброса обеспечивалось разнообразие роев частиц. Большое количество исследований показало, что такого рода стратегии могут лучше решить проблему сходимости алгоритмов и разнообразия частиц, а также значительно улучшить оптимизационную способность. Однако достижение кластеризации, будучи очень сложным процессом, принесет много временных затрат и вычислительной сложности, поэтому при решении многомерных задач многоцелевой оптимизации возникают большие трудности.

Динамическая самоадаптация и простая оптимизация множества частиц с возмущенным экстремумом и кроссовером

Упрощение уравнения скорости и динамическая адаптация инерционного веса

Предложена стратегия «упрощения уравнения скорости множества частиц и динамической регулировки адаптивного инерционного веса» с целью повышения скорости поиска и точности. Суть алгоритма заключается в процессе приближения к оптимальному значению через эволюционное положение частицы. В [9] было доказано, что скорость частиц не имеет ничего общего с оптимальным решением и был выдвинут упрощенный алгоритм PSO (sPSO) и упрощенное итерационное уравнение bPSO в виде

$$\vec{x}_i^{\,t+1} = w \cdot \vec{x}_i^{\,t} + c_1 \cdot \mathrm{rand}_1(\,) \cdot (\overrightarrow{p\,\mathrm{best}}_i^{\,t} - \vec{x}_i^{\,t}) + c_2 \cdot \mathrm{rand}_2(\,) \cdot (\overrightarrow{g\,\mathrm{best}}^{\,t} - \vec{x}_i^{\,t}). \quad (5)$$

Алгоритм sPSO устраняет концепцию скорости частицы, упрощает итерационное уравнение оптимизации множества (PSO), уменьшает влияние искусственного параметра ∂ или $v_{\max}$ на процесс оптимизации и конечный результат, снижает затраты и сложность расчета для повышения точности и скорости. Но каждая частица по-прежнему использует одну и ту же итеративную формулу эволюции, что ограничивает возможности глобальной оптимизации. Однако на среднепоздней стадии алгоритма остается застойное множествование из-за концентрированного расположения частиц.

Поэтому в данной работе на основе (5) вводится динамический адаптивный инерционный вес, а именно для плюсов и минусов каждого текущего местоположения частицы динамически придается разный инерционный вес с целью повышения способности и скорости частиц в глобальной оптимизации. Стратегия может быть описана следующим образом. По характеристикам частицы приходят к заключению о пространственном соотношении между оптимальным и текущим положениями. Для частиц, близких к оптимальному положению, уменьшается их инерционный вес, чтобы улучшить поисковую способность в пределах небольшой области вблизи оптимального положения. Для частиц, удаленных от оптимального положения, увеличивается инерционный вес, чтобы улучшить их способность приближаться к оптимальному положению. Вводится итерационное уравнение множества частиц динамического адаптивного инерционного веса в форме

$$\vec{x}_i^{\,t+1} = w^{t+1}{}_i \cdot \vec{x}_i^{\,t} + c_1 \cdot \mathrm{rand}_1(\,) \cdot (\overrightarrow{p\,\mathrm{best}}_i^{\,t} - \vec{x}_i^{\,t}) + c_2 \cdot \mathrm{rand}_2(\,) \cdot (\overrightarrow{g\,\mathrm{best}}^{\,t} - \vec{x}_i^{\,t}); \quad (6)$$

$$w^{t+1}{}_i = \begin{cases} w^t{}_i (f_i / f_{\mathrm{average}_g} - 1), \\ \text{если } (f_i \geq f_{\mathrm{average}_g}), \\ w^t{}_i (f_i / f_{\mathrm{average}_g} + 1), \\ \text{если } (f_i < f_{\mathrm{average}_g}), \end{cases} \quad (7)$$

где f_i – индивидуальное значение адаптационной способности; f_{average_g} – среднее значение для множества. Если $f_i \geq f_{\mathrm{average}_g}$, то частица находится в хорошем положении, близком к оптимальному. В этом случае уменьшается инерционный вес в следующей итерации, пропускаются оптимальные решения из-за слишком быстрого обновления положения. Если $f_i < f_{\mathrm{average}_g}$, то частица

находится в плохом положении, которое расположено далеко от оптимального. В этом случае увеличивается инерционный вес в следующей итерации и улучшается скорость оптимизации.

Введение экстремального возмущения и кроссовера

Предложенная стратегия «введения возмущенного экстремума и кроссовера» направлена на то, чтобы множество оставалось разнообразным и жизнеспособным на средней и поздней стадиях алгоритма, что играет важную роль в улучшении поисковой способности и предотвращении попадания алгоритма в ловушку локального экстремума.

Рассматривая сходимость уравнения (6), предполагаем, что $\text{rand}_1(\,)$ и $\text{rand}_2(\,)$ равномерно распределены. На основе [16–19] можно вывести:

$$\begin{aligned}\lim_{t\to\infty}\vec{x}^{\,t} &= \frac{c_1\,\text{rand}_1(\,)}{c_1\,\text{rand}_1(\,)+c_2\,\text{rand}_2(\,)}\overrightarrow{p\,\text{best}}^{\,t}+\frac{c_2\,\text{rand}_2(\,)}{c_1\,\text{rand}_1(\,)+c_2\,\text{rand}_2(\,)}\overrightarrow{g\,\text{best}}^{\,t}= \\ &= \frac{c_1\overrightarrow{p\,\text{best}}^{\,t}+c_2\overrightarrow{g\,\text{best}}^{\,t}}{c_1+c_2}=(1-\alpha)\overrightarrow{p\,\text{best}}^{\,t}+\alpha\overrightarrow{g\,\text{best}}^{\,t}.\end{aligned} \tag{8}$$

Выражение (8) показывает, что результаты оптимизации $\lim\limits_{t\to\infty}x^t$ всех частиц после итерации будут получены совместно с ее экстремумом $\overrightarrow{p\,\text{best}_i}^{\,t}$ и экстремумом множества $\overrightarrow{g\,\text{best}}^{\,t}$.

То есть, если все частицы в процессе оптимизации не находят лучшего места, чем $\overrightarrow{p\,\text{best}_i}^{\,t}$ и $\overrightarrow{g\,\text{best}}^{\,t}$, то частица не может отклоняться, поэтому поиск останавливается. Исходя из этого, предлагается: 1) наложить экстремальное возмущение на оптимальное положение $\overrightarrow{g\,\text{best}}^{\,t}$; 2) провести операцию кроссовера генетического алгоритма для всех частиц $\vec{x}_i$. А именно одновременно скорректировать положение отдельных частиц и положение экстремумов роев, заставить все частицы перемещаться в новое место, чтобы открыть новые пути поиска, а также избавиться от локального экстремума и найти лучшее решение. По сравнению со стратегией кластеризации kPSO при том же разнообразии частиц выдвинутая в данной работе стратегия о введении возмущенного экстремума и кроссовера имеет значительно меньшую вычислительную сложность и временные затраты. Это связано с тем, что на каждом шаге итерации отсутствуют сравнение, расчет и кластеризация положения частиц [20–24].

В соответствии с изложенным выражение (6) преобразуем к виду

$$\vec{x}_i^{\,t+1}=w^{t+1}\vec{x}_i^{\,t+1}+c_1\cdot\text{rand}_1(\,)\cdot(\overrightarrow{p\,\text{best}_i}^{\,t}-\vec{x}_i^{\,t})+c_2\cdot\text{rand}_2(\,)\cdot(\beta^{T>3}\overrightarrow{g\,\text{best}}^{\,t}-\vec{x}_i^{\,t}); \tag{9}$$

$$\beta^{T>3}=\begin{cases}1, & T\le 3,\\ \text{rand}(0,1), & T>3;\end{cases} \tag{10}$$

$$T=\begin{cases}1, & if\quad T>3,\\ T+1, & if\quad \overrightarrow{g\,\text{best}}^{\,t+1}=\overrightarrow{g\,\text{best}}^{\,t};\end{cases} \tag{11}$$

$$\begin{aligned}\text{child}_1(x_i)&=p_i\cdot\text{parent}_1(x_i)+(1-p_i)\cdot\text{parent}_2(x_i),\\ \text{child}_2(x_i)&=p_i\cdot\text{parent}_2(x_i)+(1-p_i)\cdot\text{parent}_1(x_i).\end{aligned} \tag{12}$$

Здесь $\beta^{T>3}$ – элемент экстремального возмущения, а его роль состоит в том, чтобы нарушить глобальный экстремум множества. Выражения (10) и (11) указывают, что если глобальное оптимальное значение множества не обновляется более трех раз, то дается случайное возмущение его экстремуму. Выражение (12) используется для изменения положения каждой частицы, так что на каждом шаге итерационного процесса сначала выбирается множество в соответствии с определенной множественностью, а затем обменивается позициями между частицами по пересечению p_i.

Когда операция скрещивания завершается, сохраняются результаты и отбрасываются исходные значения для гарантии того, что рассматривается прежнее множество.

Эксперименты и анализ результатов

Для сравнения и анализа различных характеристик алгоритма ecds-PSO были разработаны три вида экспериментальных планов:

Эксперимент 1. Анализ эффективности алгоритма по скорости сходимости и точности. Выбраны три типичные контрольные функции для фиксированного числа итераций с целью сравнения и оценки скорости сходимости и точности алгоритмов bPSO, sPSO, kPSO и ecds-PSO. Цель – проверить влияние стратегии улучшения на скорость сходимости и точность.

Эксперимент 2. Анализ разнообразия множества. Выбрана многопиковая функция Швефеля. При условии одинакового начального значения сравнивается и оценивается состояние распределения множества алгоритмов bPSO, kPSO и ecds-PSO. Цель – проверить влияние стратегии улучшения на разнообразие частиц и способность избавиться от локального экстремума.

Эксперимент 3. Статистический анализ производительности алгоритма. Выбраны три типичные бенчмарковые функции и проведен бенчмарковый тест на производительность алгоритмов bPSO, sPSO, kPSO и ecds-PSO. Цель – комплексная оценка различных статистических характеристик алгоритма ecds-PSO. В ходе эксперимента были выбраны три часто используемые тестовые функции бенчмарка и связанные с ними параметры (табл. 1).

Таблица 1

Бенчмарковые функции, используемые для тестирования методов улучшения

Код	Характеристика	Формула	Диапазон	Цель	Оптимум	Инструкция
Сфера	Унимодальный	$f(x)=\sum_{i=1}^{n} x_i^2$	$[-100, 100]^{30}$	10^{-5}	0	Обычные однопиковые функции, гладкая поверхность, используются для проверки регулярной работы алгоритма
Розеброк	Унимодальный	$f(x)=\sum_{i=1}^{n}\left(100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right)$	$[-100, 100]^{30}$	10^{2}	0	Однопиковые функции, небольшой наклон на полюсе, используются для проверки возможности выполнения алгоритма
Растригин	Мультимодальный	$f(x)=\sum_{i=1}^{n}(x_i^2-10\times\cos(2\pi x_i)+10)$	$[-100, 100]^{30}$	10^{-5}	0	Многопиковые функции, многополюсные функции, используются для проверки глобальной поисковой и оптимизационной способности алгоритма

Согласно [3, 4], размер множества алгоритма bPSO взят равным 30, $c_1=c_2=1.5$. Величина w принимает случайные значения в диапазоне $[0.4, 0.95]$, $v_{\max}=30$. На основании [9] размер множества sPSO составляет 15, $c_1=c_2=2$. Согласно [12], размер множества kPSO принимается равным 30, $c_1=c_2=0.5$, $w=0.65$, $v_{\max}=30$, коэффициент кластеризации $k=5$. Заданный размер множества алгоритма ecds-PSO равен 30, $c_1=c_2=0.5$, $w^0=0.65$, вероятность выбора генетического алгоритма принимает случайные значения в диапазоне [0.2, 0.4], а вероятность кроссовера – случайные значения в интервале [0.2, 0.8].

Скорость сходимости и точность

Используя три эталонные функции таблицы, в случае фиксированной эволюционной алгебры сравнивались точность сходимости и скорость алгоритмов bPSO, sPSO, kPSO и ecds-PSO. Результаты показаны на рис. 1–3.

Согласно кривым сходимости рис. 1, 2 и 3, алгоритмы sPSO и ecds-PSO по сравнению с bPSO и kPSO показали преимущества в скорости сходимости, что демонстрирует обоснованность и эффективность стратегии. Способность к сходимости ecds-PSO и kPSO по сравнению с bPSO и sPSO была значительно выше, т.е. стратегия имеет очень хорошие оптимизационные характеристики в способности избавляться от локального экстремума на средней-поздней стадии эволюции алгоритма, а также в точности сходимости.

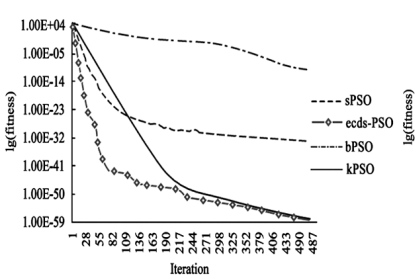

Рис. 1. Сфера

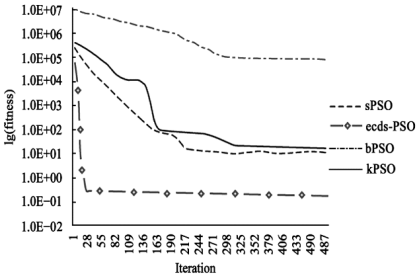

Рис. 2. Розеброк

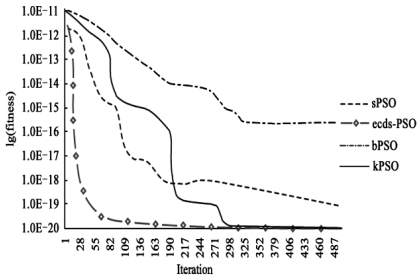

Рис. 3. Растригин

Разнообразие множеств частиц

В этом эксперименте выбрана функция Швефеля для сравнения и оценки способности алгоритмов sPSO, kPSO и ecds-PSO поддерживать разнообразие частиц в процессе эволюции. Функция Швефеля – типичная «обманчивая» функция с одной глобальной и тремя локальными оптимальными точками. Четыре точки распределены по четырем углам 3D-графика на больших расстояниях. Застряв в локальном оптимуме, трудно выпрыгнуть и достичь точки глобального оптимального решения. Эта функция используется, в основном, для тестирования алгоритма глобальной оптимизации и разнообразия множества, а конкретные графики представлены на рис. 4. Для того чтобы показать пиковую производительность функции более наглядно, в данной работе была реализована небольшая деформация и формула после деформации $\sum_{i=1}^{D} \mathrm{sum}(xi \cdot \sin(\mathrm{fabs}(xi)))$. Результаты приведены на рис. 5–7.

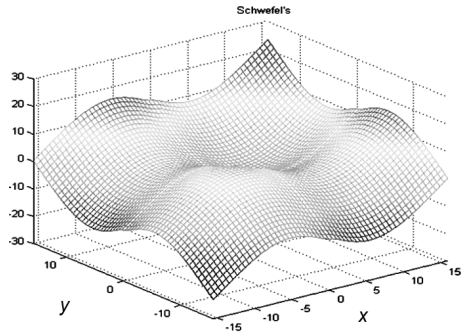

Рис. 4. Функция Швефеля

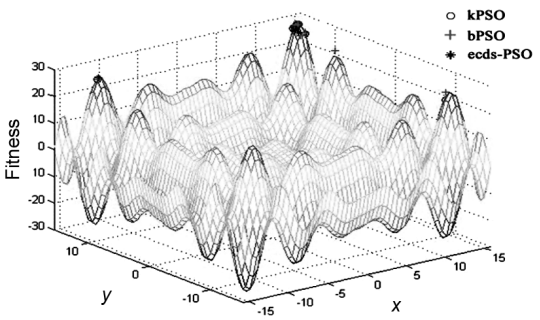

Рис. 5. Итоговое итерационное состояние для трех алгоритмов

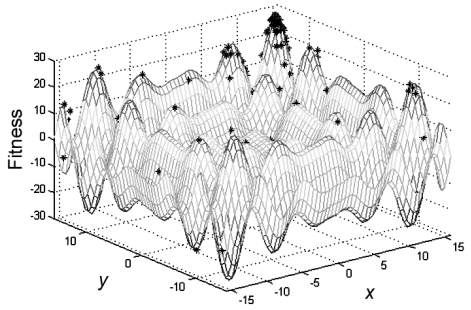

Рис. 6. Распределение множества алгоритма PSO

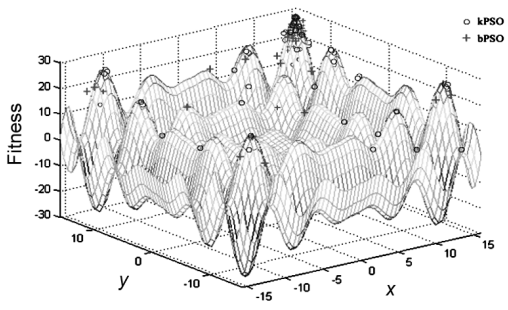

Рис. 7. Распределение множества алгоритма bPSO и kPSO

На рис. 5 показан график наложения оптимальных результатов трех алгоритмов при одинаковых начальных значениях после десятикратного запуска алгоритмов bPSO, kPSO и ecds-PSO (задана одинаковая точность оптимизации, но нет ограничений на количество итераций). Из рис. 5 следует, что bPSO находил оптимальное решение 7 раз, kPSO – 8, ecds-PSO – 10 со скоростью успеха 100%. На рис. 6 показано распределение частиц (размер множества 50) после того, как ecds-PSO нашел оптимальное решение, на рис. 7 – bPSO и kPSO нашли оптимальное решение. При сравнении рис. 6 и 7 видно, что на средне-позднем этапе эволюции алгоритма ecds-PSO разнообразие частиц оставалось на значительном уровне. По сравнению с kPSO, который характеризовался сохранением разнообразия частиц, ecds-PSO был хорош, и, исходя из теоретического анализа, ecds-PSO должен быть более случайным, чем kPSO по распределению частиц.

Статистический анализ производительности алгоритма

С помощью трех эталонных функций (см. табл. 1), проведены анализ и сравнение различных статистических элементов алгоритмов bPSO, sPSO, kPSO и ecds-PSO. Каждый алгоритм проходил независимое выполнение в течение 20 раз, а фиксированная итерация выполнялась 500 раз для каждого. Статистические параметры включали в себя: наихудшее значение (худшее), оптимальное значение (лучшее), среднее оптимальное значение (среднее лучшее) и стандартное отклонение (SD). Результаты приведены в табл. 2.

Из табл. 2 следует: 1) в функциональных тестах сферы из сравнений алгоритмов bPSO и sPSO, kPSO и ecds-PSO можно значительно улучшить точность алгоритма поиска; 2) в функциональных тестах Розеброка алгоритмы kPSO и sPSO имели одинаковый оптимизационный эффект, в то время как точность оптимизации ecds-PSO была на четыре-пять порядков выше; 3) в функциональных тестах Растригина, поскольку алгоритмы kPSO и ecds-PSO могли обеспечить более богатое разнообразие частиц и жизнеспособность, они имели значительно более высокую точность поиска, чем bPSO и sPSO; 4) благодаря производительности алгоритма ecds-PSO он значительно улучшил решающую способность по сравнению с bPSO, sPSO, kPSO, а также отразил большее преимущество в точности поиска.

Таблица 2

Статистика производительности различных улучшенных алгоритмов PSO

Функция	Алгоритм	Наихудший	Лучший	Средний лучший	SD
Сфера	bPSO	$7.69\cdot10^{-4}$	$2.67\cdot10^{-7}$	$2.01\cdot10^{-6}$	$1.42\cdot10^{-6}$
	kPSO	$1.79\cdot10^{-18}$	0	$3.62\cdot10^{-51}$	$4.25\cdot10^{-50}$
	sPSO	$4.26\cdot10^{-15}$	0	$5.38\cdot10^{-32}$	$4.68\cdot10^{-32}$
	ecds-PSO	$8.94\cdot10^{-57}$	0	$4.55\cdot10^{-57}$	$2.87\cdot10^{-57}$
Розеброк	bPSO	$7.67\cdot10^{5}$	$1.07\cdot10^{5}$	$4.07\cdot10^{5}$	$2.77\cdot10^{5}$
	kPSO	$1.49\cdot10^{6}$	$2.14\cdot10^{2}$	$8.48\cdot10^{4}$	$5.82\cdot10^{4}$
	sPSO	$5.84\cdot10^{5}$	$8.03\cdot10^{1}$	$6.87\cdot10^{3}$	$3.47\cdot10^{3}$
	ecds-PSO	$8.87\cdot10^{-1}$	$6.69\cdot10^{-1}$	$2.89\cdot10^{-1}$	$1.79\cdot10^{-1}$
Растригин	bPSO	$3.17\cdot10^{-14}$	$1.74\cdot10^{-15}$	$9.16\cdot10^{-15}$	$8.62\cdot10^{-15}$
	kPSO	$1.42\cdot10^{-18}$	0	$5.59\cdot10^{-20}$	$4.62\cdot10^{-20}$
	sPSO	$2.56\cdot10^{-14}$	$3.84\cdot10^{-17}$	$7.53\cdot10^{-16}$	$2.63\cdot10^{-16}$
	ecds-PSO	0	0	0	0

Заключение

С учетом проблем скорости сходимости и точности алгоритма множества частиц принята стратегия улучшения, которая основывается на динамической адаптации инерционного веса и упрощении уравнения скорости. Применительно к задачам оптимизации множества частиц (алгоритм PSO), которые легко попадали в локальные экстремумы, одобрена стратегия возмущенного экстремума и кроссовера. На основе объединения двух этих стратегий предложена динамическая самоадаптация и простая оптимизация множества частиц с возмущенным экстремумом и кроссовером (ecds-PSO). Разработаны три набора оптимизационных вычислительных экспериментов, которые доказали, что алгоритм оптимизации ecds-PSO обладает более высокой точностью, более высокой скоростью сходимости и большей способностью избавляться от локального экстремума.

СПИСОК ЛИТЕРАТУРЫ

1. Wang D.F. and Feng L. // Acta Automat. Sinica. – 2016. – V. 42. – No. 10. – P. 1552–1561.
2. Kennedy J. and Eberhart R. // Proc. ICNN'95 Int. Conf. on Neural Networks. – Perth, Australia, 1995. – P. 1942–1948.
3. Shi Y. and Eberhart R. // IEEE Int. Conf. on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence. – Anchorage, USA, 1998. – P. 69–73.
4. Clerc M. // Proc. the Congress on Evolutionary Computation-CEC99. – Washington, USA, 1999. – P. 1951–1957.
5. Jiang F.L., Zhang Y., and Wang Y.G. // Appl. Res. Comput. – 2017. – V. 34. – No. 12. – P. 3599–3602.
6. Li J., Wang C., Li B., and Fang G. // Appl. Res. Comput. – 2016. – V. 33. – No. 9. – P. 2584–2587, 2591.
7. Tan Y., Tan G.Z., and Deng S.Z. // Appl. Res. Comput. – 2016. – V. 33. – No. 8. – P. 6–12.
8. Cheng B.Y., Lu H.Y., Huang Y., and Xu K.B. // J. Comput. Appl. – 2017. – V. 37. – No. 3. – P. 750–754, 781.
9. Hu W. and Li Z.S. // J. Software. – 2007. – V. 18. – No. 4. – P. 861–868.
10. Ni Q.J., Zhang Z.Z., Wang Z.Z., and Xing H.C. // J. Software. – 2009. – V. 20. – No. 2. – P. 339–349.
11. Kennedy J. // Proc. Congress on Evolutionary Computation. – La Jolla, USA, 2000. – P. 1507–1512.
12. Li W.F., Liang X.L., and Zhang Y. // Acta Electron. Sinica. – 2012. – V. 40. – No. 11. – P. 2194– 2199.
13. Li C., Wang B.Y., and Gao H. // Comp. Technol. Development. – 2017. – V. 27. – No. 4. – P. 89–93.
14. Unler A. and Murat A. // Eur. J. Operation. Res. – 2010. – V. 206. – No. 3. – P. 528–539.
15. Li F., Liu J.C., Shi H.T., and Zi Y. // Control and Decision. – 2017. – V. 3. – No. 3. – P. 403–410.
16. Clerc M. and Kennedy J. // IEEE Tran. Evolution Comp. – 2002. – V. 6. – No. 1. – P. 58–73.
17. Esteban M., Núñez E.P., and Torres F. // Appl. Math. Nonlinear Sci. – 2017. – V. 2. – 449–464.
18. Ge S., Liu Z., Furuta Y., and Peng W. // Saudi J. Biol. Sci. – 2017. – V. 24. – P. 1370–1374.
19. Imam M.H., Tasadduq I.A., Ahmad A., and Aldosari F. // Eurasia J. Math. Sci. Technol. Education. – 2017. – V. 13. – P. 3069–3081.
20. Maddi B., Viamajala S., and Varanasi S. // ACS Sustainable Chem. Eng. – 2018. – V. 6. – P. 237–247.
21. Bruzón M.S. and Garrido T.M. // Discrete and Continuous Dynam. Systems. – 2018. – V. 11. – P. 631–641.
22. Shen Y., Zhao N., Xia M., and Du X. // Polish Maritime Res. – 2017. – V. 24. – P. 102–109.
23. Sun X., Chen F., and Hewings G.J.D. // Emerging Markets Finance & Trade. – 2017. – V. 53. – No. 5. – P. 2063–2081.
24. Cai L., Chen J., Peng X., and Chen B. // Int. J. Technol. Management. – 2016. – V. 72. – No. 1–3. – P. 171–191.

Поступила в редакцию 17.11.2020.

[1] School of Business Administration, Hubei University of Economics, Wuhan, China
[2] College of Public Administration, Huazhong University of Science & Technology, Wuhan, China
[3] Faculty of Economics and Business, KU Leuven, Leuven, Belgium
[4] School of Management, Wuhan University of Technology, Wuhan, China
[5] Fujian Zhuozhi Project Investment Consulting Co., LTD, Wuhan, China

Ya Bi, Ph.D., Professor School of Business Administration, Hubei University of Economics, College of Public Administration, Huazhong University of Science & Technology, e-mail: biya@hbue.edu.cn;
Anthony Lam, Ph.D., Professor Faculty of Economics and Business, KU Leuven, e-mail: 21259537@qq.com;
Huiqun Quan, Ph.D., Professor School of Business Administration, Hubei University of Economics, e-mail: oh2020987456@163.com;
Hui Liu, Associate Professor, Professor School of Business Administration, Hubei University of Economics, e-mail: liuhui@hbue.edu.cn;
Cunfa Wang, Master, Engineer School of Management, Wuhan University of Technology, Fujian Zhuozhi Project Investment Consulting Co., LTD, e-mail: 912832894@qq.com.