УДК 519.6 DOI: 10.26102/2310-6018/2019.26.3.006

Б.С. Ермаков

МЕТОД РОЯ ЧАСТИЦ С АДАПТИВНЫМИ СОЦИАЛЬНОЙ И КОГНИТИВНОЙ КОМПОНЕНТАМИ

Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, Россия

Эффективность решения оптимизационных задач с помощью метода роя частиц в значительной степени зависит от выбранных исследователем значений социальной и когнитивной компонент. На данный момент нет возможности однозначно определять такие значения этих параметров, которые бы обеспечивали максимальную эффективность поиска решения для конкретной задачи. В целях устранения этого недостатка, в данной статье предлагается модификация метода роя частиц, в которой социальная и когнитивная компоненты алгоритма адаптируются в процессе оптимизации к исследуемой задаче, избавляя таким образом исследователя от необходимости подбирать значения этих компонент вручную. В основе адаптации лежат принципы, сходные с генетическими алгоритмами: осуществляются отбор наиболее эффективных частиц, скрещивание – передача их значений социальной и когитивной компонент другим частицам, и мутация – случайные модификации значений компонент. Для оценки эффективности полученного алгоритма была проведена серия экспериментов по нахождению минимумов нескольких тестовых функций. Найденные минимумы, усредненные по каждой группе тестов, сравнивались с минимумами, найденными каноническим методом роя частиц. На основе полученных результатов, была выдвинута и подтверждена статистическая гипотеза о превосходстве адаптивного методя роя частиц над каноническим. Проведенное исследование свидетельствует об эффективности применения представленного адаптивного метода для решения практических задач.

Ключевые слова: оптимизация, метод роя частиц, адаптация, генетические алгоритмы.

Введение.

Многие практические задачи в исследовании технических, экономических и социальных систем можно свести к задаче оптимизации – поиску таких значений параметров системы, которые обеспечат наибольшее (или наименьшее) значение рассматриваемого критерия. Для непрерывных, дифференцируемых и выпуклых функций существует развитый аппарат детерминированных методов оптимизации (в первую очередь основанных на вычислении градиента функции). Однако на практике исследователь далеко не всегда располагает априорной информацией о виде функции, что ограничивает применение детерминированных методов. Поэтому стоит задача разработки таких подходов, которые позволят эффективно находить экстремумы функций любого вида.

Метод роя частиц (МРЧ) – метод оптимизации, широко использующийся в различных приложениях. Одно из важных его преимуществ – это прямой метод, т.е. не требуется вычислять производные целевой функции (ЦФ), соответственно не требуется информация о гладкости функции. Благодаря этому обстоятельству, МРЧ является универсальным методом – требуется лишь возможность вычислять значения ЦФ в конкретной точке.

Недостатком МРЧ является необходимость подбора социального и когнитивного параметров, от их значений зависит то, насколько эффективно рой сможет находить оптимальные решения. При этом для функций различного вида наилучший результат может достигаться с различными значениями этих параметров. И нет общепризнанного подхода для их подбора, как правило, он осуществляется эмпирически, в ходе множественных экспериментов. А это может существенно увеличивать временные и вычислительные ресурсы, требуемые на решение задачи.

На данный момент предложено большое число модификаций канонического МРЧ, каждая из них нацелена на совершенстовавание и развитие различных аспектов поиска. В статье [1] произведен обзор 17 модификаций МРЧ, отличающихся топологией частиц и способами обмена информацией между ними. Было также предложено большое количество методов, объединяющих МРЧ с другими методами, например: с градиентным спуском [2], с генетическими алгоритмами [3][4], и даже модель состоящая из трех алгоритмов сразу [5] – МРЧ, генетический алгоритм и метод восхождения к вершине, которые применяются поочередно. Многие из этих модификаций нацелены на конкретные типы задач, либо требуют определенной априорной информации об исследуемой проблеме. В данной статье предложена модификация МРЧ, нацеленная на улучшение эффективности алгоритма в общем, и не налагающая требований к виду функции. В соответствии с ней, частицы эволюционируют, обмениваясь информацией о значениях социальной и когнитивной компонент наиболее эффективных частиц, таким образом происходит адаптация к исследуемой проблеме.

Канонический метод роя частиц.

Метод роя частиц изначально [6] задумывался как модель поведения стаи птиц, однако проявил себя как эффективный метод оптимизации. В основе его лежит представление о множестве частиц, перемещающихся в пространстве поиска. Начальные положения и скорости частиц задаются случайно. Движение каждой частицы определяется лучшим найденным этой частицей положением, и лучшим найденным роем положением по следующим формулам:

$$V_{di}(t+1) = w * V_{di}(t) + c_1 * r_1 * (P_{di} - x_{di}(t)) + c_2 * r_2 * (G_d - x_{di}(t)) \quad (1)$$

$$x_{di}(t+1) = x_{di}(t) + V_{di}(t+1) \quad (2)$$

Где $i \in (1 \dots n)$ - номер частицы, $d \in (1 \dots D)$ – номер пространственного измерения, V – скорость частицы, x – координата частицы, t – номер итерации, P – координаты лучшего найденного частицей положения, G – координаты лучшего найденного всем роем положения, c_1 и c_2 – когнитивная и социальная компоненты соответственно, r_1 и r_2 – случайные числа, равномерно распределенные в промежутке [0; 1]. w – коэффициент инерции, был предложен в статье [7] и его использование в МРЧ стало уже классическим подходом.

Таким образом, на каждой итерации положение каждой частицы изменяется на случайный вектор, образованный из векторов направленных в сторону лучших положений, найденных самой частицей и всем роем. Такой подход обеспечивает эффективный поиск как на начальном этапе, когда частицы с большими скоростями свободно перемещаются в поисковом пространстве, исследуя его на области экстремумов, так и на конечном этапе, когда движение частиц сосредоточено вокруг лучшего найденного решения, с возможностью найти вблизи него еще более точное положение экстремума.

Эффективность этого поиска обуславливается в первую очередь значениями социальной и когнитивной компоненты, которые определяют в какой степени частицы будут стремиться к лучшим частным и к лучшему общему найденным положениям. При высоких значениях когнитивной компоненты, частицы будут за короткое время застревать около начальных найденных ими субоптимальных значений, при высоких значениях социальной компоненты частицы быстро сойдутся к одному субоптимальному значению (которое может быть далеко от глобального экстремума). При малых значениях социальной и когнитивной компоненты найденные оптимумы будут слабо влиять на движение частицы, оно будет более случайным, обусловленным в первую очередь начальным (случайным) значением скорости частицы, что тоже затрудняет поиск эффективных решений [7][8].

Адаптивный МРЧ.

Для решения проблемы подбора значений социальной и когнитивной компонент в данной статье предлагается модификация канонического МРЧ (в дальнейшем эта модификация обозначена как АМРЧ), 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 [9]. 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.

Начальные значения социальной и когнитивной компонент задаются случайными числами, равномерно распределенными в некотором интервале. Через равные промежутки времени оценивается эффективность поиска каждой частицы – ее приспособленность. На основе степени приспособленности каждой частицы (ее эффективности) производятся операции селекции, скрещивания и мутации: чем большую эффективность продемонстрировали частицы с определенными значениями компонент, тем больше вероятность что эти значения компонент будут переданы другим частицам. Мутация обеспечивает необходимое разнообразие значений компонент.

Оценка приспособленности частицы.

Оценку приспособленности и генетические операции кажется уместным проводить на каждой итерации. Однако эксперименты показали что такой подход может приводить к существенному ухудшению результатов: если значения функции в исследуемой области сильно различаются между собой (как например у функции Розенброка, значения которой при незначительном изменении координат могут различаться на много порядков), то частица, случайно оказавшаяся в области в которой значения функции значительно меньше (например в ходе начальной инициализации частиц), захватит весь или почти весь генетический пул частиц, несмотря на то что значения ее социальной и когнитивной компонент могут не обеспечивать каких либо преимуществ в плане поиска экстремума данной функции.

Чтобы обойти данную проблему, был применен прием, который условно можно назвать «генетическим сглаживанием»: эволюция частиц производится не на каждой итерации поиска, а с интервалом в k итераций. Эксперименты показали, что уместной является эволюция каждые 3-5 итераций, все значения частоты эволюции из этого интервала показали близкие результаты, которые были значительно лучше, чем при эволюции на каждой итерации. Бо́льшие значения представляются неуместными, т.к. чем ниже частота эволюции, тем в меньшей степени выражены преимущества ее использования.

В качестве функции приспособленности используется сумма значений целевой функции в точках, в которых побывала частица за время, прошедшее с прошлого этапа эволюции. В случае если значения ЦФ требуется минимизировать, в качестве функции приспособленности следует использовать значение, обратное этой сумме:

$$fitness = 1/\sum_{k} f \tag{3}$$

Таким образом, меньшим значениям ЦФ будут соответствовать большие значения функции приспособленности.

Селекция.

Для селекции частиц используется т.н. метод рулетки [10]: каждой частице, исходя из ее функции приспособленности, присваивается вероятность что ее выберут, в аналогии с рулеткой – сектор определенной ширины, чем он шире тем больше вероятность. Затем происходит отбор частиц: рулетку «вращают» n раз (по количеству частиц в рое) – т.е. n раз случайным образом отбирается частица, причем вероятность выбора конкретной частицы определяется следующей формулой:

$$p_i = fitness_i \Big/ \sum_i fitness_i \qquad (4)$$

Таким образом, будет образован массив «родителей», причем частицы, которым значения их социальной и когнитивной компонент обеспечивают бóльшую эффективность, буду присутствовать в этом массиве чаще.

Скрещивание.

Когда родительский массив сформирован, начинается его скрещивание с исходным массивом частиц. Используется однородный кроссовер (uniform crossover) – каждый ген (социальная или когнитивная компонента) частицы заменяется на соответствующий ген частицы из «родительского» массива с вероятностью 50% [11]. Таким образом, значения социальной и когнитивной компонент, обеспечивающие лучший результат, с течением времени захватывают всё большую часть генетического пула частиц – происходит адаптация поведения частиц к исследуемой задаче.

Мутация.

Над полученными в ходе скрещивания частицами выполняется операция мутации: с некоторой вероятностью к гену каждой частицы может быть добавлена случайная величина, нормально распределенная около нуля. Мутации позволяют вносить разнообразие в значения социальной и когнитивной компонент, не ограничивая их только изначально сгенерированными числами. Эксперименты показали, что эффективные результаты достигаются при частоте мутации в диапазоне от 0.1 до 0.25. Также, хорошо себя зарекомендовал подход, при котором среднеквадратичное отклонение (СКО) величины, добавляемой при мутации, линейно уменьшается с течением времени.

Для начальных значений рекомендуется использовать СКО в диапазоне 0.1-0.3, конечные ее значения должны быть близки к нулю – порядка 0.01-0.05. Таким образом, на начальных этапах поиска, когда значения социальной и когнитивной компонент частиц в значительной степени еще случайны, они будут активно варьироваться, позволяя протестировать различные значения компонент; однако на завершающих этапах влияние мутации будет сведено к минимуму – к этому моменту, в ходе эволюции, уже будут выработаны удачные генотипы частиц (значения их компонент).

После операций скрещивания и мутации производится проверка на соответствие значений социальной и когнитивной компонент всех частиц граничным условиям. Это сделано для предотвращения бесконтрольного их уменьшения или увеличения, которые могут сделать эффективный поиск невозможным.

Схематично, процесс эволюции частиц представлен на Рисунке 1:

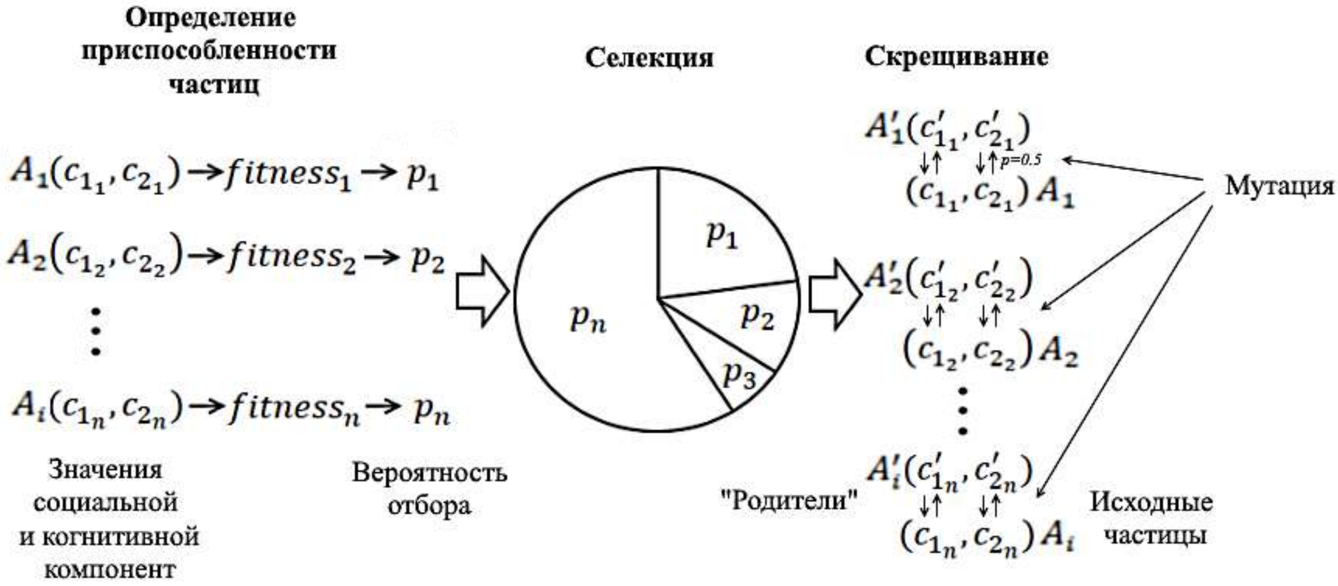

Рисунок 1 – Процесс создания следующего поколения частиц

Весь алгоритм можно описать следующими шагами:

1. Инициализация начальных положений частиц x_i, начальных скоростей частиц V_i и начальных значений социальных и когнитивных компонент каждой частицы $A_i[c_{1_i}; c_{2_i}]$ случайными числами.

2. Определение первых значений P_i лучших найденных частицами значений ЦФ, и G - лучшего имеющегося положения из всего роя.

3. Изменение скоростей частиц по формуле (1), причем вместо фиксированных c_1 и c_2 используются значения из массива $A_i[c_{1_i}; c_{2_i}]$ – у каждой частицы свои значения социальной и когнитивной компоненты. Проверка на соответствие скоростей частиц граничным условиям. Изменение положений частиц по формуле (2).

4. Вычисление ЦФ в полученных точках, и проверка улучшились ли частное P_i и общее G решения.

5. Полученное значение ЦФ для i-ой частицы прибавляется к значениям $F_i := F_i + f_i$ (изначально F_i инициализируется нулем). Это требуется для упомянутого выше «генетического сглаживания».

6. На каждой k-ой итерации происходит эволюция частиц. Найденные к этому моменту значения F_i используются для вычисления приспособленности каждой частицы. В случае если решается задача максимизации, приспособленность частицы равна F_i, в случае если решается задача минимизации – используются обратные F_i значения, в соответствии с формулой (3). По найденным значениям приспособленности частиц вычисляется вероятность отбора частицы в соответствии с формулой (4). Значения F_i обнуляются.

7. Отбор по методу рулетки: формируется родительский массив из n частиц, отобранных исходя из найденных вероятностей. В нем в большей степени присутствуют частицы, показавшие хорошие результаты на предыдущих k итерациях.

8. Скрещивание: значения социальной или когнитивной компоненты каждой исходной частицы с вероятностью 50% заменяются соответствующими значениями компонент частицы с тем же номером из родительского массива.

9. К полученным значениям компонент применяется операция мутации: с вероятностью m к компоненте будет прибавлена случайная величина, нормально распределенная, с математическим ожиданием равным 0, и среднеквадратичным отклонением σ. Как было указано выше, σ с течением времени изменяется от больших значений к меньшим по линейному закону: $\sigma = \sigma_{max} - \frac{t}{T} * (\sigma_{max} - \sigma_{min})$, где σ_{max} и σ_{min} – максимальное и минимальное значения, которые принимает σ в начале и в конце поиска соответственно, а t и T – текущий номер итерации и общее количество итераций поиска соответственно.

10. Проверка соответствия полученных социальных и когнитивных компонент граничным условиям.

11. Повторение пунктов 3-10 до достижения условия остановки (количества итераций).

Эксперименты.

Сравнение алгоритмов осуществлялось по результатам минимизации трех разных функций:

1. функция сферы (5) – выпуклая функция с единственным минимумом равным 0 в точке (0, 0, ... 0);

2. функция Растригина (6), которая состоит из большого количества локальных минимумов, глобальный минимум равен 0 и находится в точке (0,0, ... 0);

3. функция Розенброка (7), которая имеет сложный рельеф вблизи глобального минимума, равного 0 в точке (1,1, ... 1), и резко возрастает при удалении от него.

Формулы, по которым вычисляются значения функций, приведены ниже:

$$f(x) = \sum_{d=1}^{D} x_d^2 \tag{5}$$

$$f(x) = 10D + \sum_{d=1}^{D} [x_d^2 - 10\cos(2\pi x_d)] \tag{6}$$

$$f(x) = \sum_{d=1}^{D-1} [100(x_{d+1} - x_d^2)^2 + (x_d - 1)^2] \tag{7}$$

Для каждой из этих функций были проведены серии экспериментов, в которых менялась размерность задачи, количество частиц, участвующих в поиске, и количество итераций поиска. Также, в некоторых сериях количество испытаний было уменьшено, в силу значительных временных ресурсов, необходимых на решение задачи.

Начальные положения частиц равномерно распределены в промежутке [-100;100], причем на область поиска в дальнейшем никаких ограничений не накладывается. Начальные значения скоростей частиц равномерно распределены в промежутке [-10;10], этими же значениями ограничены скорости частиц в ходе поиска.

Параметры канонического МРЧ: коэффициент инерции $w = 0.9$, когнитивная и социальная компоненты $c_1 = c_2 = 0.5$.

Параметры адаптивного МРЧ: коэффициент инерции $w = 0.9$, частота мутации $m = 0.15$, начальное значение среднеквадратичного отклонения при мутации $\sigma_{max} = 0.2$, конечное значение среднеквадратичного отклонения при мутации $\sigma_{min} = 0.05$, частота «генетического сглаживания» $k = 5$ – эволюция частиц происходит каждые 5 итераций. Значения социальной и когнитивной компонент ограничены интервалом [0; 1].

В Таблице 1 представлены результаты экспериментов. В Таблице приведены функция, значения которой минимизировались в конкретной серии испытаний; N – количество испытаний в этой серии; D – размерность функции; P – количество частиц, участвовавших в поиске; T – количество итераций поиска; средние значения найденных минимумов для АМРЧ и МРЧ; среднее время в секундах, затрачиваемое каждым алгоритмом на поиск решения и Δ - отношение результата МРЧ к АМРЧ (чем больше значение Δ, тем больше эффективность АМРЧ в сравнении с МРЧ).

Таблица 1 – Результаты экспериментов

№	Функция	N	D	P	T	АМРЧ результат	АМРЧ время	МРЧ результат	МРЧ время	Δ
1	Сфера	100	50	50	500	0.03	1.25	0.13	1.22	4.3
2	Сфера	100	100	100	1000	0.1	9.98	4.57	9.7	45.7
3	Сфера	50	200	200	1000	24.8	40.3	209.8	38.5	8.5
4	Сфера	50	200	200	2000	1.51	79.2	44.2	78.3	29.3
5	Сфера	50	200	400	1000	0.29	79.3	19.4	78.2	66.9
6	Сфера	30	500	500	1000	1270	251	4801	247	3.8
7	Растригин	100	10	10	1000	24	0.12	35	0.11	1.5
8	Растригин	100	50	50	1000	428	2.78	692	2.62	1.6
9	Растригин	100	50	50	2000	408	5.64	711	5.14	1.7
10	Растригин	100	50	100	1000	287	5.55	415	5.26	1.4
11	Растригин	100	100	100	1000	1388	10.4	2355	10.17	1.7
12	Растригин	100	100	200	1000	1025	20.9	1489	20.23	1.5
13	Растригин	50	200	200	1000	4514	41.4	7363	40.5	1.6
14	Растригин	30	500	500	1000	19395	263	27812	262	1.4
15	Розенброк	100	10	10	1000	17.7	0.117	18.6	0.108	1.1
16	Розенброк	100	50	50	1000	105	2.7	117	2.69	1.1
17	Розенброк	100	50	50	2000	63	5.3	79	5.2	1.3
18	Розенброк	100	50	100	1000	75	5.4	70	5.3	0.9
19	Розенброк	100	100	100	1000	388	10.8	721	10.7	1.9
20	Розенброк	100	100	100	2000	222	23.3	282	21.4	1.3
21	Розенброк	100	100	200	1000	200	23	226	20.9	1.1
22	Розенброк	50	200	200	1000	5 039	43.4	38 904	42	7.7
23	Розенброк	50	200	200	2000	1 051	90.9	4 958	85.1	4.7
24	Розенброк	50	200	400	1000	809	90.7	2 187	85.9	2.7
25	Розенброк	30	500	500	1000	1 807 267	270	15753889	267	8.7

Оценка результатов.

Из Таблицы 1 видно, что АМРЧ показал лучший результат во всех испытаниях, кроме испытания №18, в котором оба метода показали очень близкие результаты. Причем если для функции Растригина АМРЧ находил в 1.4-1.7 раз меньший минимум, то для функции Розенброка, при больших размерностях, прирост эффективности составил от 3 до 9 раз; а для функции сферы прирост эффективности в целом составил от 4 до 67 раз.

Произведем оценку полученных результатов. Нулевая гипотеза состоит в том что эффективность обоих алгоритмов одинакова, т.е. $\Delta = 1$.

Альтернативная гипотеза состоит в том что эффективность АМРЧ выше, чем эффективность МРЧ, т.е. $\Delta > 1$.

Для проверки гипотезы будет использоваться t-критерий Стьюдента, на уровне значимости 5%:

$$t = \frac{\bar{\Delta} - m}{\sigma/\sqrt{n}} \tag{8}$$

где $\bar{\Delta}$ – среднее по выборке значение Δ, m=1 – значение мат. ожидания из нулевой гипотезы, σ – стандартное отклонение по выборке, n=25 – количество элементов в выборке.

По данным Таблицы 1 можно найти значение критерия Стьюдента: t=2.24. Известное критическое значение критерия для уровня доверия 95% и 24 степеней свободы $t_{0.95;24}$= 2.06. Так как $t > t_{0.95;24}$, нулевая гипотеза отвергается – эффективность АМРЧ выше, чем эффективность МРЧ.

Заключение.

В статье была предложена модификация метода роя частиц, обеспечивающая заметный (до двух порядков) прирост эффективности в поиске экстремума. На уровне доверия в 95% было доказано превосходство предложенного АМРЧ по сравнению с каноническим МРЧ.

Важно отметить, что в большинстве испытаний увеличение времени, требуемого АМРЧ на решение задачи по сравнению с МРЧ, составляло порядка 1-4%, максимальное увеличение составило 10%. В подавляющем большинстве практических приложений такое увеличение затрачиваемого времени можно считать несущественным.

Несмотря на то, что в АМРЧ используется больше управляющих параметров (m, σ_{max}, σ_{min}, k), подбор их значений не так принципиален, как подбор значений c_1 и c_2 в каноническом МРЧ, т.к. в случае с АМРЧ параметры влияют на процесс поиска только косвенно, и в ходе эволюции частицы в любом случае будут адаптироваться.

Приведенные соображения позволяют утверждать, что использование описанного АМРЧ предпочтительней по сравнению с использованием канонического МРЧ, т.к. результаты, обеспечиваемые АМРЧ, в целом не хуже, а во многих случаях значительно лучше, чем результаты МРЧ; причем на решение затрачивается почти такое же время.

ЛИТЕРАТУРА

1. А.П. Карпенко. Обзор методов роя частиц для задачи глобальной оптимизации / А.П. Карпенко, Е.Ю. Селиверстов // Машиностроение и компьютерные технологии. – 2009 – №3 – 26 с.
2. Jing-Ru Zhang. Hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training / Jing-Ru Zhang, Jun Zhang, Tat-Ming Lok, Michael R. Lyu // Applied Mathematics and Computation. – 2007 – №2 – pp. 1026-1037.
3. V. Miranda. EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems / V. Miranda, N. Fonseca // Transmission and Distribution Conference and Exhibition. – 2002 – №2 – pp. 740-750.
4. Chia-Feng Juang. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design / Chia-Feng Juang // Transactions on systems, man, and cybernetics – Part B: Cybernetics. – 2004 – №2 – pp. 997 – 1006.
5. T. Krink. The LifeCycle model: Combining Particle Swarm Optimisation, Genetic Algorithms and HillClimbers» / T. Krink, M. Lovbjerg // Parallel Problem Solving from Nature — PPSN VII. – 2002 – pp. 621-630.
6. J. Kennedy. Particle swarm optimization / J. Kennedy, R. C. Eberhart // Proceedings of the 1995 IEEE International Conference on Neural Networks. – 1995 – pp. 1942-1948.
7. Shi Y. A modified particle swarm optimizer / Shi Y., Eberhart R. // Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence. – 1998 – pp. 69-73.
8. Ratnaweera, A. «Self-organizing Hierarchical Particle Swarm Optimizer with Time-varying Acceleration Coefficients» / Ratnaweera, A., Halgamuge, S., Watson, H. //IEEE Transactions on Evolutionary Computation. – 2004 – pp. 240-255.
9. J.H. Holland. Adaptation in natural and artificial systems / J.H. Holland. – Cambridge: MIT Press, 1992. – 225 pp.
10. М.В. Бураков. Генетический алгоритм: теория и практика: учеб. пособие / М. В. Бураков. – СПб.: ГУАП, 2008. – 164 с.
11. Z. Michalewicz. A Note on Usefulness of Geometrical Crossover for Numerical Optimization Problems» / Z. Michalewicz, G. Nazhiyath, M. Michalewicz // Proceedings of the 5th Annual Conference on Evolutionary Programming. – 1996 – pp. 305-312.

B.S. Ermakov

PARTICLE SWARM OPTIMIZATION WITH ADAPTIVE SOCIAL AND COGNITIVE COMPONENTS

Saint Petersburg State University of Aerospace Instrumentation,
Saint-Petersburg, Russia

Efficiency of solution finding by particle swarm optimization depends significantly on specific values of social and cognitive components used by a researcher. There is no known way currently to determine whether specific values of the components would provide maximal search efficiency in a particular case, or not. In order to eliminate this flaw, this article provides a modification of particle swarm optimization with adaptive social and cognitive components, which allows to fit particles movement to a particular problem during optimization process, thus removing the need of adjusting components manually. This adaption is based on genetic algorithms principles: it starts with a selection of the best performing particles, then crossover of their social and cognitive components with other particles, then mutation to provide some fluctuations of components. To evaluate algorithm's performance a series of experiments on minimizing few test functions has been made. Minimums found by adaptive and canonical algorithms were averaged out and compared. Based on results, a statistical hypothesis that adaptive algorithm has better performance than canonical algorithm was confirmed. Provided research proves efficiency of adaptive particle swarm.

Keywords: mathematical optimization, particle swarm optimization, adaptation, genetic algorithms.

REFERENCES

1. A.P. Karpenko. Overview of particle swarm optimizaion methods for global optimization / A.P. Karpenko, E.Y. Seliverstov // Mechanical engineering and computer technologies. – 2009 – №3 – 26 pp.
2. Jing-Ru Zhang. Hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training / Jing-Ru Zhang, Jun Zhang, Tat-Ming Lok, Michael R. Lyu // Applied Mathematics and Computation. – 2007 – №2 – pp. 1026-1037.
3. V. Miranda. EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems / V. Miranda, N. Fonseca // Transmission and Distribution Conference and Exhibition. – 2002 – №2 – pp. 740-750.
4. Chia-Feng Juang. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design / Chia-Feng Juang // Transactions on systems, man, and cybernetics – Part B: Cybernetics. – 2004 – №2 – pp. 997 – 1006.
5. T. Krink. The LifeCycle model: Combining Particle Swarm Optimisation, Genetic Algorithms and HillClimbers» / T. Krink, M. Lovbjerg // Parallel Problem Solving from Nature — PPSN VII. – 2002 – pp. 621-630.
6. J. Kennedy. Particle swarm optimization / J. Kennedy, R. C. Eberhart // Proceedings of the 1995 IEEE International Conference on Neural Networks. – 1995 – pp. 1942-1948.

7. Shi Y. A modified particle swarm optimizer / Shi Y., Eberhart R. // Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence. – 1998 – pp. 69-73.
8. Ratnaweera, A. «Self-organizing Hierarchical Particle Swarm Optimizer with Time-varying Acceleration Coefficients» / Ratnaweera, A., Halgamuge, S., Watson, H. //IEEE Transactions on Evolutionary Computation. – 2004 – pp. 240-255.
9. J.H. Holland. Adaptation in natural and artificial systems / J.H. Holland. – Cambridge: MIT Press, 1992. – 225 pp.
10. M.V. Burakov. Genetic algorithm: theory and practice. / M.V. Burakov. – Saint-Petersburg: SUAI, 2008. – 164 pp.
11. Z. Michalewicz. A Note on Usefulness of Geometrical Crossover for Numerical Optimization Problems» / Z. Michalewicz, G. Nazhiyath, M. Michalewicz // Proceedings of the 5th Annual Conference on Evolutionary Programming. – 1996 – pp. 305-312.