

## Ко-гибридизация алгоритмов роя частиц

77-30569/355792

# 04, апрель 2012

Воробьева Е. Ю., Карпенко А. П., Селиверстов Е. Ю.

УДК 519.6

МГТУ им. Н.Э. Баумана

[vorobeva\\_ey@mail.ru](mailto:vorobeva_ey@mail.ru)

### Введение

Алгоритмы решения задач глобальной безусловной оптимизации можно разделить на три группы – детерминированные, стохастические и эвристические. Эвристические алгоритмы, в свою очередь, разделяют на эволюционные и поведенческие алгоритмы. Алгоритм роя частиц (*Particle Swarm Optimization, PSO*), рассматриваемый в данной работе, является поведенческим алгоритмом безусловной оптимизации [1]. Эффективность алгоритма *PSO* в значительной мере зависит от используемой алгоритмом топологии соседства частиц. Так, например, топология типа клика обеспечивает высокую эффективность алгоритма, но может привести к его преждевременной сходимости к некоторому локальному минимуму целевой функции. В случае топологии типа кольцо преждевременной сходимости можно избежать, но алгоритм оказывается эффективным лишь для сложных многоэкстремальных функций.

Целью данной работы является исследование эффективности ко-алгоритмической гибридизации алгоритмов *PSO* с различными топологиями соседства частиц [2]. Точнее говоря, рассматривается двухпопуляционный ко-алгоритм *CPSO* на основе алгоритмов *PSO*, использующих топологии соседства типа клика и кольцо.

### 1. Алгоритм роя частиц

Алгоритм *PSO* был опубликован в 1995 году. В основу алгоритма положена социально-психологическая поведенческая модель толпы. Агентами в алгоритме являются частицы в пространстве параметров задачи оптимизации, обладающие в этом пространстве положениями и скоростями. На каждой итерации при определении следующего положения частицы учитывается информация о наилучшей частице из числа

соседних частиц, а также информация о положении данной частицы на той итерации, когда этой частице соответствовало наилучшее значение функции цели. Известно большое число модификаций алгоритма, большинство из которых являются последовательными, хотя известны и параллельные варианты алгоритма [1].

Рассматриваем задачу глобальной безусловной минимизации целевой функции  $F(X)$  в  $|X|$ -мерном арифметическом пространстве  $R^{|X|}$ :

$$\min_{X \in R^{|X|}} F(X) = F(X^*) = F^*. \quad (1)$$

Множество частиц обозначим  $P = \{p_i, i \in [1:|P|]\}$ , где  $|P|$  – число частиц в рое (размер популяции). В момент времени  $t$  координаты частицы  $p_i$  определяются  $|X|$ -мерными векторами ее координат  $X_{i,t}$  и скорости  $V_{i,t}$ . Начальные координаты и скорости частицы  $p_i$  равны  $X_{i,0} = X_i^0$ ,  $V_{i,0} = V_i^0$  соответственно. Итерации метода *PSO* выполняются в соответствии с формулами

$$X_{i,t+1} = X_{i,t} + V_{i,t+1}, \quad (2)$$

$$V_{i,t+1} = \alpha V_{i,t} + U[0;\phi_1] \otimes (X_{i,t}^b - X_{i,t}) + U[0;\phi_2] \otimes (X_{g,t} - X_{i,t}). \quad (3)$$

Здесь  $U[a;b]$  –  $|X|$ -мерный вектор псевдослучайных чисел, равномерно распределенных в интервале  $[a;b]$ ;  $\otimes$  – символ покомпонентного умножения векторов;  $X_{i,t}^b$  – вектор координат частицы  $p_i$  с наилучшим (в смысле формулы (1)) значением целевой функции  $F(X)$  за все время поиска  $[0:\hat{t}]$ ;  $X_{g,t}$  – вектор координат соседней с данной частицы с наилучшим за то же время значением целевой функции;  $\alpha$ ,  $\phi_1$ ,  $\phi_2$  – свободные параметры алгоритма. Параметр  $\alpha$  определяет «инерционные» свойства частицы,  $\beta$  – ее «когнитивные» свойства,  $\gamma$  – «социальность» частицы [1]. Рекомендуемые значения параметров  $\alpha$ ,  $\beta$ ,  $\gamma$  равны, соответственно, 0,7298; 1,49618; 1,49618.

Соседство частиц в рое определяется топологией соседства, которую задают неориентированным графом. Вершины графа соответствуют частицам роя, а ребра связывают непосредственных соседей. В алгоритме *PSO* обычно используют следующие топологии соседства частиц:

- топология клика (глобальная оптимальная топология);
- топология кольцо (локальная оптимальная топология);

- топология двумерный тор (топология фон Неймана);
- топология кластер.

Две первые топологии, которые используются в алгоритме *CPSO*, иллюстрируют рисунки 1, 2. Топологии клика соответствует полносвязный граф (рисунок 1), в котором соседями каждой из частиц  $p_i$ ,  $i \in [1:|P|]$  являются остальные  $(|P|-1)$  частицы. Диаметр клика равен единице. В топологии кольцо (рисунок 2) соседями каждой из частиц являются всего две частицы, которые имеют соседние номера. Диаметр соответствующего графа равен  $O\left(\frac{|P|}{2}\right)$ .

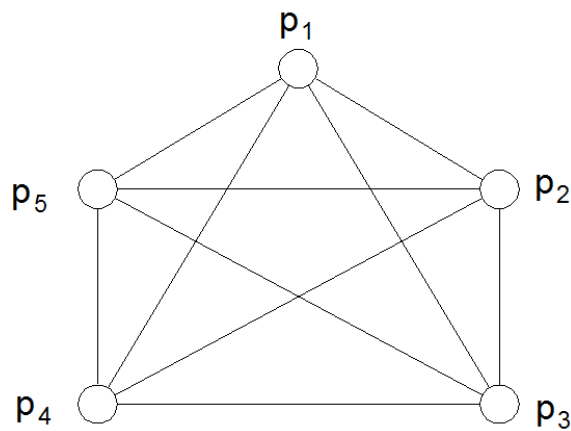


Рис. 1. Топология соседства типа клика:  $|P| = 5$

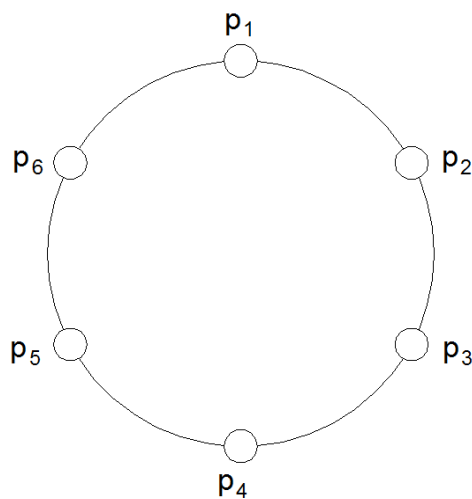


Рис. 2. Топология соседства частиц кольцо:  $|P| = 6$

Диаметр графа, соответствующего используемой роем топологии соседства частиц, определяет скорость распространения информации в рое. В рое с топологией соседства клика лучшее значение целевой функции, достигнутое той или иной частицей, сразу становится известным всем остальным частицам роя. Самую низкую скорость распространения информации обеспечивает топология кольцо.

Известно, что топология клика обеспечивает высокую скорость сходимости алгоритма *PSO*, но может привести к его преждевременной сходимости к некоторому локальному минимуму целевой функции. Так что данную топологию целесообразно использовать при оптимизации одноэкстремальных функций. Топология кольцо эффективна при оптимизации сложных многоэкстремальных функций. Причина этого заключается в том, что частицы роя, расположенные далеко друг от друга в кольце, слабо связаны друг с другом и могут эффективно исследовать различные части области поиска. Топология обеспечивает слабое «притяжение» частиц к локальным минимумам целевой функции, что позволяет избежать преждевременной сходимости.

## **2. Ко-алгоритмическая гибридизация алгоритмов роя частиц**

В основе ко-алгоритмической гибридизации популяционных алгоритмов оптимизации лежит следующая основная идея. Одновременно в пространстве поиска эволюционируют несколько субпопуляций, каждая из которых базируется на одном из популяционных алгоритмов и решает задачу оптимизации (1). Каждая из субпопуляций «борется» за вычислительные ресурсы, которые по окончании заданного числа итераций перераспределяют в пользу более эффективной из субпопуляций.

Вообще говоря, ко-алгоритм может быть построен на основе различных поисковых алгоритмов оптимизации. Классический ко-алгоритм предполагает использование в качестве субпопуляционных алгоритмов генетического алгоритма и называется, поэтому, генетическим коэволюционным алгоритмом. В этом алгоритме одинаковыми для всех субпопуляционных алгоритмов являются структура и значения свободных параметров, за исключением параметров, определяющих генетические операторы селекции и формирования нового поколения [2].

В данной работе в качестве субпопуляционных алгоритмов используются алгоритмы *PSO*, которые по сравнению с генетическим алгоритмом обладает рядом преимуществ [3, 4]:

- алгоритм *PSO* обладает «памятью», поскольку знание о лучшем из найденных решений рано или поздно становится известным всем частицам популяции

(в генетическом алгоритме лучшее решение может быть уничтожено при изменении популяции);

- алгоритм *PSO* подразумевает «сотрудничество» между частицами, так как частицы в рое делятся информацией между собой.

С другой стороны, в алгоритм *PSO*, также как и в генетическом алгоритме, возможна преждевременная сходимость.

Для генетического коэволюционного алгоритма рекомендуется использовать общее число субпопуляций  $|\mathbf{P}|$ , равное три - пять. В данной работе используем субпопуляции

$$P_1 = \{p_{1,i}, i \in [1:|P_1|]\}, \quad P_2 = \{p_{2,i}, i \in [1:|P_2|]\}$$

так что  $|\mathbf{P}| = 2$ .

Кроме числа субпопуляций, наиболее важными свободными параметрами ко-алгоритма являются:

- величина ресурса  $n_f$  – максимально допустимое число испытаний (вычислений значений целевой функции);

- интервал адаптации  $\Delta_a \in (0; T)$ , где  $T$  – максимально допустимое число итераций;

- величина штрафа  $n_p$ ;

- минимально допустимый размер субпопуляции  $|P|_{\min}$ .

На этапе инициализации ко-алгоритма величина ресурса  $n_f$  распределяется поровну между субпопуляциями. Условием окончания итераций ко-алгоритма может быть исчерпание этого ресурса.

Интервал адаптации  $\Delta_a$  – это число итераций, в течение которых субпопуляции не обмениваются информацией друг с другом, но итерируют обособленно. Чем меньше значение  $\Delta_a$ , тем меньше возможностей у субпопуляционных алгоритмов продемонстрировать особенности своего поведения. Все субпопуляции показывают в этом случае близкие результаты, и эффект гибридизации исчезает. При больших значениях величины  $\Delta_a$  эффективность ко-алгоритма также может быть низкой, поскольку в этом случае значительное число субпопуляций может оказаться в неперспективных подобластях пространства поиска.

Штраф назначается проигравшей субпопуляции (проигравшему субпопуляционному алгоритму). Величина штрафа  $n_p \in (0; 1)$  имеет смысл доли, на

которую сокращается размер проигравшей субпопуляции. Значение штрафа не следует назначать слишком маленьким, так как в таком случае субпопуляционные алгоритмы могут не ощутить на себе влияния штрафа. Также значение  $n_p$  не стоит выбирать слишком близким к единице, поскольку в этом случае проигравшие субпопуляционные алгоритмы окажутся неэффективными в силу малых мощностей своих популяций.

Минимально допустимый размер субпопуляции  $|P|_{\min}$  ограничивает снизу размер проигравших популяций. Значение этого параметра назначают, исходя из отмеченной выше неэффективности малых субпопуляций. Обычно используют не абсолютные значения величины  $|P|_{\min}$ , а ее относительные значения

$$\eta_i = \frac{|P|_{\min}}{|P_i|}, \quad i \in [1:|\mathbf{P}|].$$

Для коэволюционного генетического алгоритма рекомендованные значения указанных свободных параметров алгоритма приведены в таблице 1 [2, 5].

Таблица 1

Рекомендованные значения свободных параметров коэволюционного генетического алгоритма

| Параметр       | Рекомендованное значение |
|----------------|--------------------------|
| $ \mathbf{S} $ | $3 \div 5$               |
| $\Delta_a$     | $5 \div 7$               |
| $n_p$          | $0,02 \div 0,15$         |
| $ S _{\min}$   | $(0,05 \div 0,35)  S_i $ |

Блок-схема алгоритма CPSO имеет вид, представленный на рисунке 3.



Рис. 3. Общая схема ко-алгоритма

Рассмотрим основные шаги ко-алгоритма.

1) Задание параметров субалгоритмов и ко-алгоритма.

Основные свободные параметры, значения которых должны быть определены на данном шаге, перечислены выше.

2) Инициализация субпопуляций.

Инициализация субпопуляционного алгоритма *PSO* заключается в присвоение всем агентам всех субпопуляций начальных положений в пространстве поиска и начальных скоростей.

3) Выполнение каждой субпопуляцией  $\Delta_a$  независимых итераций.

Этот шаг включает в себя асинхронное независимое выполнение заданного числа итераций  $\Delta_a$  каждым из субпопуляционных алгоритмов  $P_i$ ,  $i \in [1:|\mathbf{P}|]$ .

4) Оценка эффективности субпопуляции.

Оценка эффективности субпопуляции  $P_i$  производится на основе оценки текущих значений ее функции пригодности:  $\varphi_i = \varphi_i(t)$ ;  $i \in [1:|\mathbf{P}|]$ ,  $t \in [1:T]$ . Пригодностью субпопуляции называется мера адаптации (приспособленности) этой субпопуляции к

задаче оптимизации (1). Пригодность тем выше, чем ближе лучшее решение, найденное данной субпопуляцией, к положению экстремума целевой функции.

Значение пригодности вычисляется, например, по формуле

$$\varphi_i(t) = \sum_{\tau=0}^{\Delta_a-1} \frac{\Delta_a - \tau}{\tau + 1} a_i(\tau), \quad i \in [1:|\mathbf{P}|] \quad (4)$$

где  $a_i(\tau) = 1$ , если субпопуляция  $P_i$  на итерации  $(t - \tau)$  включала в себя лучшего во всей популяции агента, и  $a_i(\tau) = 0$  в противном случае. Вес величины  $a_i(\tau)$  определяется отношением  $\frac{\Delta_a - \tau}{\tau + 1}$ , так что при  $\tau = 0$  (текущая итерация) этот вес максимален и равен интервалу адаптации  $\Delta_a$ . С уменьшением номера итерации  $\tau$  вес величины  $a_i(\tau)$  в сумме (4) быстро уменьшается.

#### 5) Проверка выполнения условия останова.

В случае выполнения условия останова, в качестве решения принимаем наилучшее решение, найденное популяцией, и прекращаем вычисления. В противном случае переходим к следующему шагу. В качестве условия останова может быть использовано условие достижение заданного числа итераций, стагнация вычислительного процесса в течение заданного числа итераций, исчерпание ресурса  $n_f$ .

#### 6) Перераспределение ресурсов.

Перераспределение ресурсов производим на основе приспособленности субпопуляций (4). Если текущее значение пригодности субпопуляции  $P_i$  максимально по сравнению с пригодностями остальных субпопуляций, то эта субпопуляция признается победившей, а остальные субпопуляции – проигравшими. Перераспределение ресурсов осуществляем путем сокращения размера каждой из проигравших субпопуляций на величину штрафа, определяемого величиной  $n_p$ , и увеличения размера победившей субпопуляции на число, равное сумме потерь проигравших. Таким образом, общий размер популяции не изменяется.

Если размер проигравшей субпопуляции оказывается меньше величины  $|P|_{\min}$ , то принимаем его равным этой величине. Таким образом, если  $|P_i|$  – размер субпопуляции, проигравшей на текущем интервале адаптации, то на следующем интервале размер этой субпопуляции будет равен величине

$$|P_i'| = \max(n_p |P_i|, |P|_{\min}), \quad i \in [1:|\mathbf{P}|].$$

Если же субпопуляция  $P_i$  выиграла на текущем интервале адаптации, то на следующем интервале размер этой субпопуляции будет равен

$$|P_i'| = |P_i| + \sum_j \max(n_p |P_i|, |P|_{\min}), \quad i, j \in [1:|\mathbf{P}|], \quad i \neq j.$$

### 3. Исследование эффективности ко-алгоритма

Исследование выполнено при варьировании значений свободных параметров алгоритма в следующих пределах:

- $\Delta_a = 5; 6; 7; 8; 9; 10;$
- $n_p = 0,05; 0,1; 0,15; 0,2; 0,25; 0,3;$
- $\eta = 0,15; 0,2; 0,25; 0,3; 0,35; 0,4.$

В качестве тестовых функций рассмотрены функции Розенброка (*Rosenbrock*), Химмельблау (*Himmelblau*) и Растригина (*Rastrigin*). Эффективность ко-алгоритма оцениваем средним достигнутым значением целевой функции  $\bar{F}$  по результатам мултистартов, а также аналогичными минимальным  $F^{best}$  и максимальными значениями  $F^{worst}$ .

#### 3.1. Сравнение ко-алгоритма с базовым алгоритмом PSO

Выполнено сравнение эффективности ко-алгоритма с эффективностью базовых алгоритмов *PSO* с топологиями клика и кольцо. Результаты исследования представлены в таблице 2. Исследование выполнено при следующих значениях свободных параметров: число итераций  $T$  равно 100; размеры популяций  $P_1$ ,  $P_2$  одинаковы и равны 16;  $n_p = 0,15$ ;  $\Delta_a = 9$ ;  $\eta_i = 0,25$ . Наряду с величинами  $\bar{F}$ ,  $F^{best}$ ,  $F^{worst}$  в таблице представлены также оценки среднеквадратичных отклонений этих величин  $\bar{\sigma}$ ,  $\sigma^{best}$ ,  $\sigma^{worst}$  соответственно.

Таблица 2

Сравнение эффективности ко-алгоритма с эффективностью базовых алгоритмов *PSO*

| Функция     |                  | Алгоритм            |                      |             |
|-------------|------------------|---------------------|----------------------|-------------|
|             |                  | <i>PSO</i><br>клика | <i>PSO</i><br>кольцо | ко-алгоритм |
| Розенброка  | $F^{best}$       | 1,1 е-004           | 2,8 е-004            | 4,6 е-011   |
|             | $\sigma^{best}$  | 3,1 е-008           | 6,7 е-008            | 1,5 е-020   |
|             | $\bar{F}$        | 0,01                | 0,10                 | 0,01        |
|             | $\bar{\sigma}$   | 2,0 е-004           | 0,006                | 4,5 е-004   |
|             | $F^{worst}$      | 0,08                | 0,90                 | 0,06        |
|             | $\sigma^{worst}$ | 0,01                | 1,16                 | 0,02        |
| Химмельблау | $F^{best}$       | 8,2 е-007           | 3,4 е-007            | 0           |
|             | $\sigma^{best}$  | 1,6 е-012           | 3,7 е-013            | 0           |
|             | $\bar{F}$        | 5,0 е-004           | 0,05                 | 0,02        |
|             | $\bar{\sigma}$   | 1,1 е-006           | 0,01                 | 8,4 е-004   |
|             | $F^{worst}$      | 0                   | 0,48                 | 0,08        |
|             | $\sigma^{worst}$ | 2,1 е-006           | 0,43                 | 0,02        |
| Растригина  | $F^{best}$       | 0,50                | 0                    | 0           |
|             | $\sigma^{best}$  | 0,25                | 6,5 е-005            | 0           |
|             | $\bar{F}$        | 0,64                | 1,32                 | 0,32        |
|             | $\bar{\sigma}$   | 0,14                | 0,09                 | 0,07        |
|             | $F^{worst}$      | 1,54                | 4,14                 | 0,88        |
|             | $\sigma^{worst}$ | 0,91                | 2,90                 | 0,77        |

Таблица 2 показывает, что ко-алгоритм обеспечивает более точную локализацию глобального минимума рассматриваемых тестовых функций, чем базовые алгоритмы *PSO*.

### 3.2. Влияние величины интервала адаптации

Выполнено исследование эффективности ко-алгоритма в зависимости от величины интервала адаптации  $\Delta_a$ . Рассмотрены значения этого параметра, равные 5; 6; 7; 8; 9; 10. Исследование выполнено при следующих значениях свободных параметров ко-алгоритма,

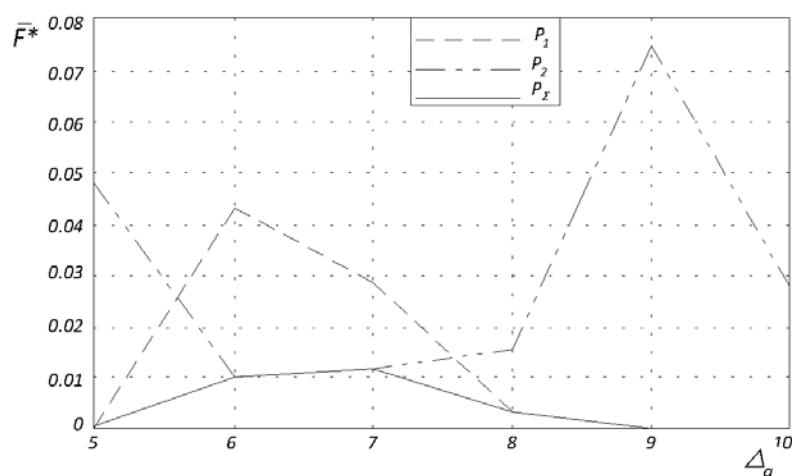
приведенных в п. 3.1. В качестве меры эффективности ко-алгоритма используем лучшее среднее по числу мултистартов значение целевой функции  $\bar{F}^*$ .

Результаты исследования иллюстрирует рисунок 4. Здесь и далее  $P_\Sigma = P_1 \cup P_2$  - популяция ко-алгоритма. Из рисунка 4а следует, что для функции Розенброка субпопуляция  $P_1$  побеждает субпопуляцию  $P_2$  при интервале адаптации, равном, 5; 8 - 10, а субпопуляция  $P_2$  побеждает субпопуляцию  $P_1$  при  $\Delta_a=6, 7$ . Оптимальными значениями величины  $\Delta_a$  для ко-алгоритма являются значения 5; 9; 10.

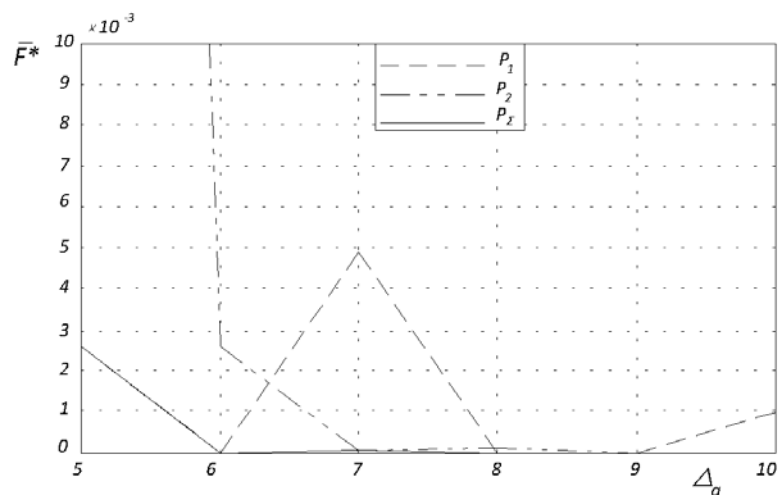
Для функции Химмельблау  $P_1$  побеждает  $P_2$  при  $\Delta_a=5; 8$ , а  $P_2$  побеждает  $P_1$  при  $\Delta_a=7; 9, 10$  (рисунок 4б). Оптимальными для ко-алгоритма являются  $\Delta_a=5; 9; 10$ .

Аналогично для функции Растргина (рисунок 4в) первая субпопуляция побеждает вторую при  $\Delta_a=5 - 7$ , вторая субпопуляция первую – при  $\Delta_a=8 - 10$ , и оптимальными для ко-алгоритма являются  $\Delta_a=5; 9; 10$ .

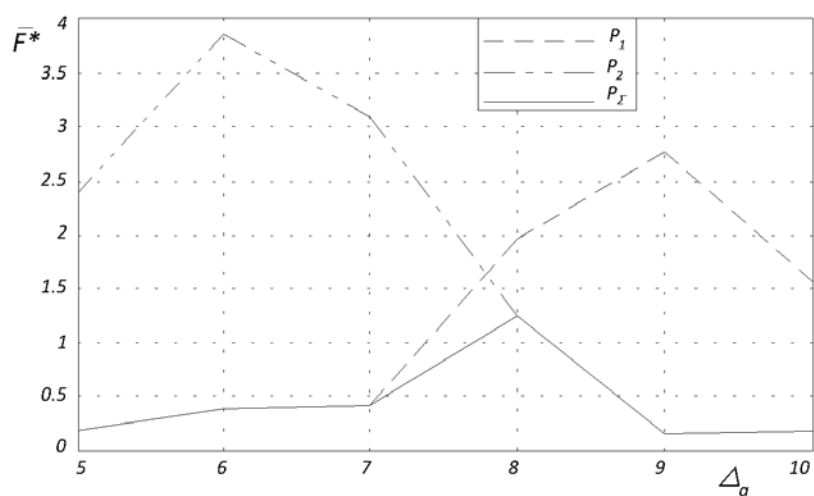
Таким образом, исходя из результатов исследования, в качестве оптимального интервала адаптации ко-алгоритма можно рекомендовать интервал, равный девяти.



а) функция Розенброка



б) функция Химмельблау



в) функция Растргина

Рис. 4. Лучшее среднее значение целевой функции  $\bar{F}^*$  в зависимости от величины интервала адаптации  $\Delta_a$

### 3.3. Влияние коэффициента штрафа

Исследование эффективности ко-алгоритма выполнено при значениях штрафного коэффициента  $n_p$ , равного 0,05; 0,1; 0,15; 0,2; 0,25; 0,3. Величина интервала адаптации  $\Delta_a$  принята равной своему оптимальному значению 9. Значения остальных свободных параметров алгоритма приведены в п. 3.1. Некоторые результаты исследования представлены на рисунке 5.

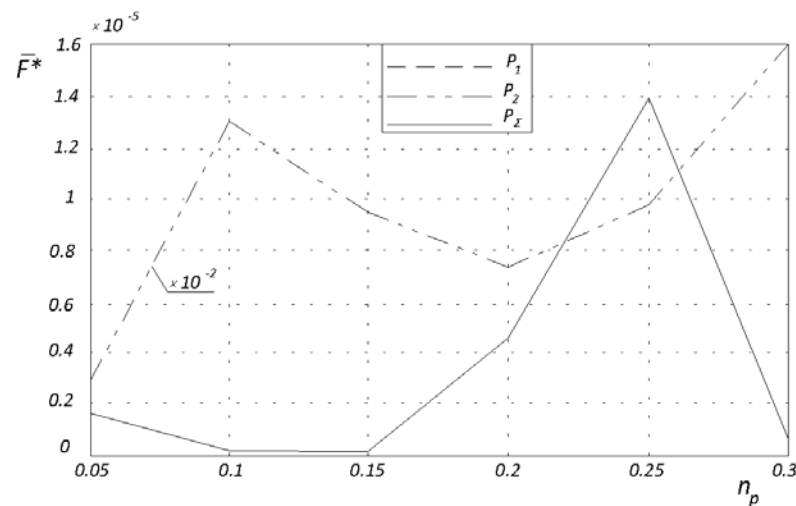
Рисунок 5а показывает, что для функции Розенброка при всех рассмотренных значений величины  $n_p$  победителем оказывается субпопуляция  $P_1$  (поэтому графики

зависимостей  $\bar{F}^*(n_p)$  для субпопуляций  $P_\Sigma$  и  $P_2$  совпадают). Оптимальными являются значения штрафного коэффициента, равные 0,1 и 0,15.

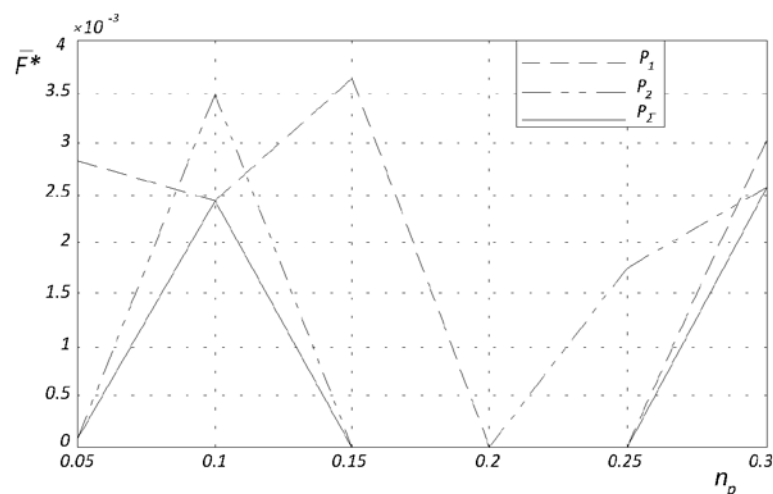
Из рисунка 5б следует, что для функции Химмельблау оптимальными являются значения коэффициента штрафа, равные 0,15 - 0,25.

Для функции Растригина (рисунок 5в) при всех исследуемых значениях коэффициента штрафа лучших значений  $\bar{F}^*$  достигал рой с топологией кольцо (поэтому графики зависимостей  $\bar{F}^*(n_p)$  для субпопуляций  $P_\Sigma$  и  $P_2$  совпадают). Оптимальным для функции Растригина является значение коэффициента штрафа, равное 0,15.

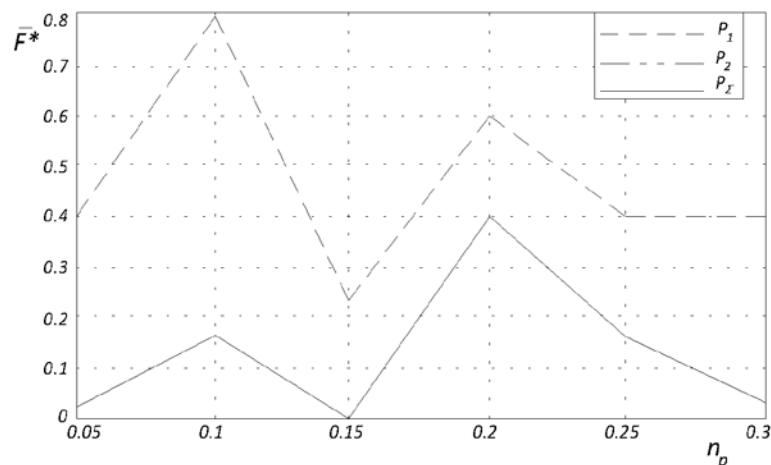
Таким образом, в качестве оптимального значения коэффициента штрафа можно рекомендовать значение, равное 0,15.



а) функция Розенброка



б) функция Химмельблау



в) функция Растригина

Рис. 5. Лучшее среднее значение целевой функции  $\bar{F}^*$  в зависимости от значений штрафного коэффициента  $n_p$

### 3.4. Влияние минимального размера субпопуляций

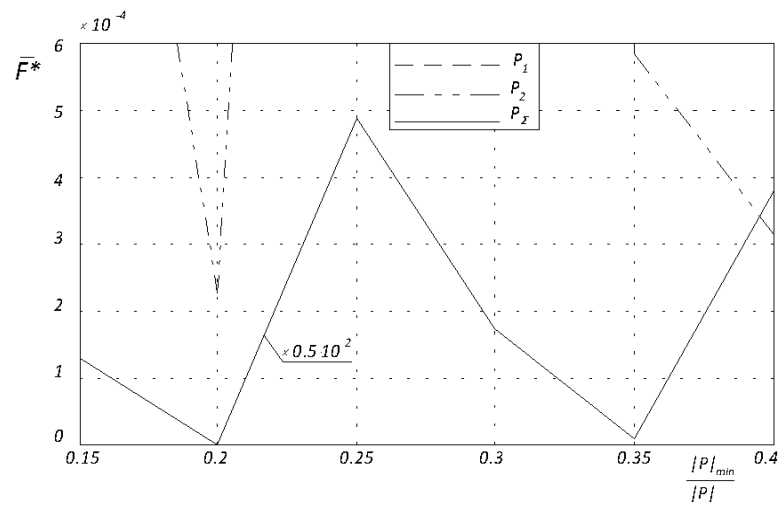
Рассматриваем следующие величины коэффициента минимального размера субпопуляции  $\eta_i$ : 0,15; 0,2; 0,25; 0,3; 0,35; 0,4. Значение коэффициента штрафа  $n_p$  принято равным 0,15. Значения остальных свободных параметров указаны выше. Результаты исследования иллюстрирует рисунок 6.

Рисунок 6а показывает, что для функции Розенброка при всех значениях параметра  $\eta$  победителем оказалась субпопуляция  $P_1$ ; оптимальные значения величины  $\eta$  равны 0,2; 0,35.

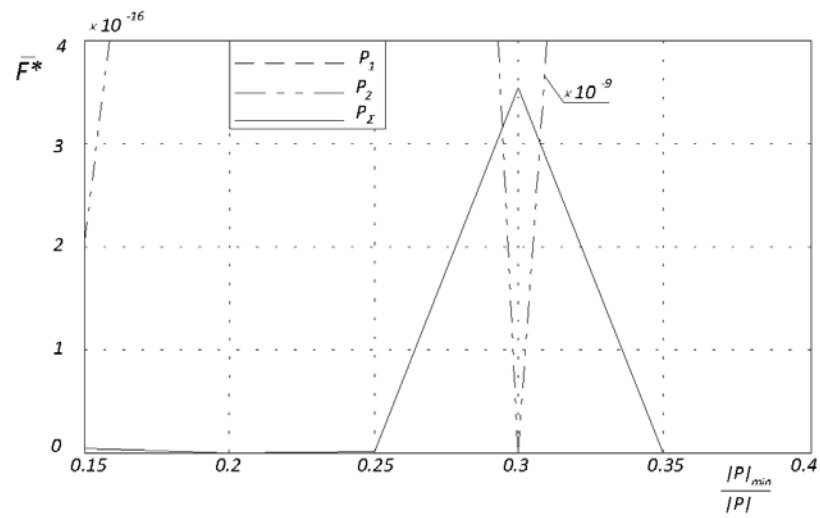
Из рисунка 6б видно, что для функции Химмельблау победителем также оказался рой с топологией клика; оптимальные значения величины  $\eta$  равны 0,2; 0,25; 0,35; 0,4.

Из рисунок 6в следует, что в случае функции Растригина при всех рассмотренных значениях параметра  $\eta$  победителем является субпопуляция  $P_2$ ; оптимальные значением  $\eta$  равны 0,2; 0,25; 0,35.

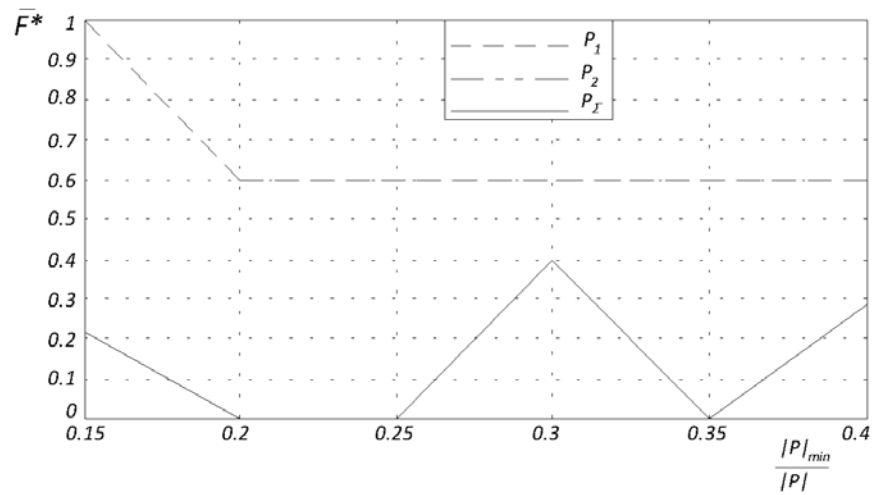
Таким образом, в качестве оптимального значения коэффициента минимального размера субпопуляции  $\eta$  можно рекомендовать величину 0,25.



а) функция Розенброка



б) функция Химмельблау

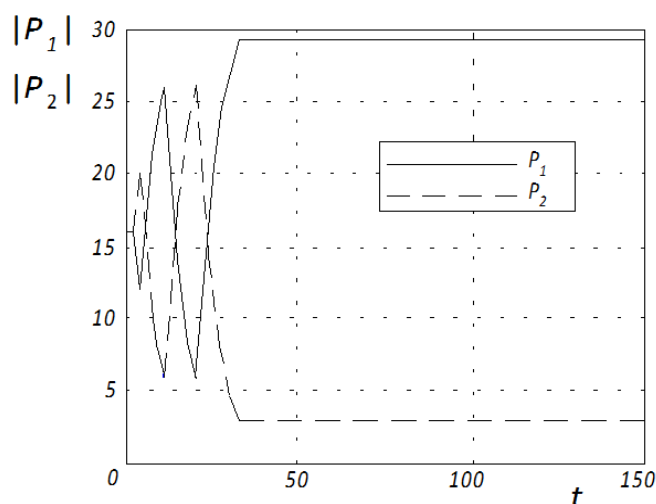


в) функция Растригина

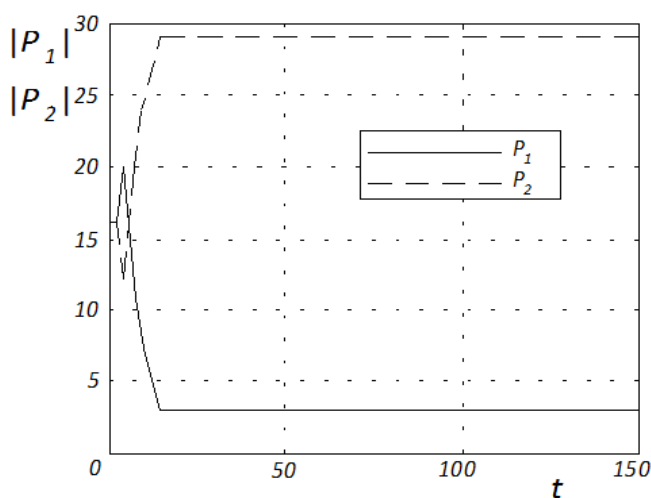
Рис. 6. Лучшее среднее значение целевой функции  $\bar{F}^*$  в зависимости от значений коэффициента минимального размера субпопуляции  $\eta$ .

### 3.5. Динамика размеров субпопуляций

Перераспределение ресурсов между субпопуляциями иллюстрирует рисунок 7. Приняты следующие значения свободных параметров алгоритма: начальные размеры субпопуляций  $|P_1| = |P_2| = 16$ ;  $\Delta_a = 9$ ;  $n_p = 0,15$ ;  $\eta = 0,2$ ;  $T = 250$ .



а) Функция Розенброка



а) Функция Растригина

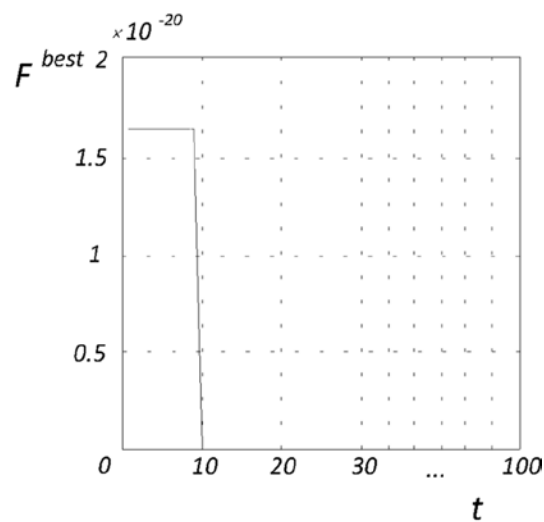
Рис. 7. Изменение размеров субпопуляций  $|P_1|$ ,  $|P_2|$  в процессе итераций

Рисунок 7а показывает, что для унимодальной функции Розенброка, как и следовало ожидать, в конечном счете, побеждает рой  $P_1$  с топологией клика. На первых 32-х итерациях рои с переменным успехом «борются» за вычислительные ресурсы. Для многоэкстремальной функции Растригина (рисунок 7б), опять же ожидаемо, побеждает

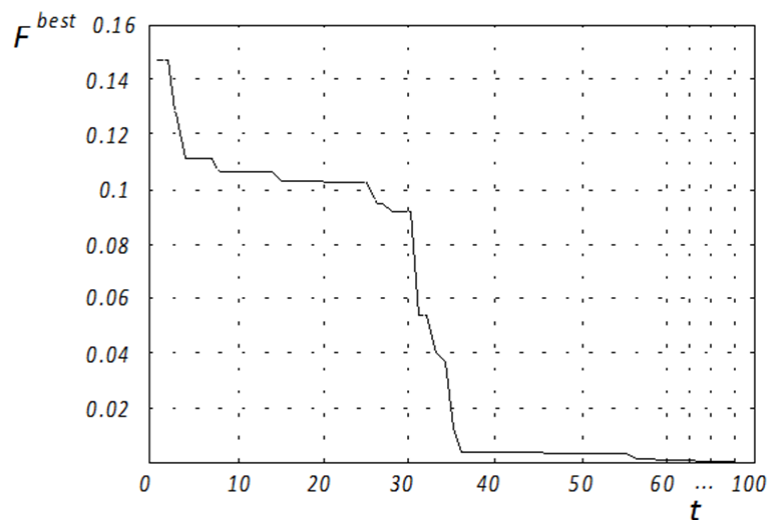
субпопуляция  $P_2$  с топологией кольцо. «Борьба» субпопуляций продолжается всего 14 итераций.

### 3.6. Сходимость ко-алгоритма

Эффективность ко-алгоритма особенно очевидна из сопоставления рисунков 8а, 8б, которые показывают характер изменения лучшего значения целевой функции  $F^{best}$  в процессе итераций для ко-алгоритма и базового алгоритма  $PSO$  соответственно. Исследование выполнено при следующих значениях свободных параметров алгоритма: начальные размеры субпопуляций  $|P_1| = |P_2| = 16$ ;  $\Delta_a = 9$ ;  $n_p = 0,15$ ;  $\eta = 0,2$ ;  $T = 100$ .



а) ко-алгоритм



б) базовый алгоритм  $PSO$

Рис. 8. Сходимость ко-алгоритма и базового алгоритма  $PSO$ : функция Розенброка

Отметим высокую скорость сходимости ко-алгоритма. По сути, решение задачи получается в течение всего одного интервала адаптации (девяти итераций). Базовый алгоритм *PSO* обеспечивает гораздо более низкую скорость сходимости – решение достигается в течение примерно 70 итераций.

### Заключение

В работе выполнено широкое исследование эффективности ко-алгоритма *PSO*, в котором коэволюционируют две субпопуляции с топологиями клика и кольцо. Показано, что ко-алгоритм обеспечивает более высокую эффективность по сравнению с базовыми алгоритмами *PSO* для всех тестируемых функций – унимодальной функции Розенброка, многоэкстремальной функции Растригина и функции Химмелблау, имеющей небольшое число экстремумов.

Для рассматриваемого набора тестовых функций найдены оптимальные значения основных свободных параметров ко-алгоритма - интервал адаптации, коэффициент штрафа и коэффициент минимально допустимого размера субпопуляции. Найденные оптимальные значения свободных параметров примерно в два раза повышают скорость сходимости ко-алгоритма. По истечении всего пятидесяти итераций ко-алгоритма с этими значениями параметров достигается точность локализации глобального экстремума, примерно равная точности, достигаемой тем же алгоритмом при исходных значениях этих параметров за 200 итераций.

В развитие работы предполагается исследование эффективности ко-алгоритмической гибридизации более двух алгоритмов *PSO*, а также эффективности гибридизации по той же схеме различных популяционных алгоритмов, например, алгоритм роя частиц и алгоритм искусственных иммунных систем, генетический алгоритм и алгоритм роя частиц и так далее.

### Литература

1. Zhou Y., Pei Sh. A hybrid co-evolutionary particle swarm optimization algorithm for solving constrained engineering design problems. // China Journal of computers, 2010, V. 5, N. 6, P. 965-972.
2. Shi Yu., Krohling R. A. Co-evolutionary particle swarm optimization to solve min-max problems // Evolutionary Computation, 2002, V. 2, P. 1682-1687.
3. Yongquan Zhou, Shengyu Pei. A hybrid co-evolutionary particle swarm optimization algorithm for solving constrained engineering design problems. // China: Journal of computers. June, 2010. V. 5. N. 6. P. 965-972.

4. Карпенко А.П., Селиверстов Е.Ю. Глобальная оптимизация методом роя частиц. Обзор. // Информационные технологии, 2010, № 2, С. 25-34.

5. Процыков Г.В., Семенкин Е.С., Томкин К.А. Об эффективности коэволюционного подхода в практических задачах оптимизации. // Вестник Красноярского государственного университета, 2005, №4, С. 233-239.

## Co-hybridization of PSO

77-30569/355792

# 04, April 2012

Vorob'eva E.Yu., Karpenko A.P., Seliverstov E.Yu.

Bauman Moscow State Technical University

[vorobeva\\_ey@mail.ru](mailto:vorobeva_ey@mail.ru)

This article deals with co-algorithmic hybridization of two Particle Swarm Optimization (PSO) algorithms, using click-type and ring-type neighborhood topologies of particles. Each of these topologies has their disadvantages which can be eliminated by using co-algorithmic hybridization. The authors compared the efficiency of canonical PSO and its co-algorithmic modification and found optimal values of free parameters.

---

**Publications with keywords:** [PSO](#), [co-algorithm](#), [hybridization](#)

**Publications with words:** [PSO](#), [co-algorithm](#), [hybridization](#)

---

### References

1. ZhouY., Pei Sh. A hybrid co-evolutionary particle swarm optimization algorithm for solving constrained engineering design problems. *Journal of computers (JCP)*. Academy Publisher , 2010, vol. 5, no. 6, pp. 965-972.
2. Shi Yu., Krohling R.A. Co-evolutionary particle swarm optimization to solve min-max problems. *Evolutionary Computation*, 2002, vol. 2, pp. 1682-1687.
3. ZhouY., Pei Sh. A hybrid co-evolutionary particle swarm optimization algorithm for solving constrained engineering design problems. *Journal of computers (JCP)*, 2010, vol. 5, no. 6, pp. 965-972.
4. Karpenko A.P., Seliverstov E.Iu. Global'naia optimizatsiia metodom roia chastits. Obzor. [Overall optimization by method of particles swarm. Review]. *Informatsionnye tekhnologii*, 2010, no. 2, pp. 25-34.
5. Protsykov G.V., Semenkin E.S., Tomkin K.A. Ob effektivnosti koevoliutsionnogo podkhoda v prakticheskikh zadachakh optimizatsii [The effectiveness of co-evolutionary approach in practical optimization problems]. *Vestnik Krasnoyarskogo gosudarstvennogo universiteta* [Bulletin of KSU], 2005, no.4, pp. 233-239.