

## ЭВОЛЮЦИОННЫЕ ОПЕРАТОРЫ ПОПУЛЯЦИОННЫХ АЛГОРИТМОВ ГЛОБАЛЬНОЙ ОПТИМИЗАЦИИ. ОПЫТ СИСТЕМАТИЗАЦИИ

Карпенко Анатолий Павлович

Д.ф.-м.н., профессор, зав. кафедрой «Системы автоматизированного проектирования»,  
ФГБОУ «Московский Государственный технический университет им. Н.Э. Баумана»,  
105005 г. Москва, 2-я Бауманская ул. д. 5,  
e-mail: akarpenko@bmstu.ru

**Аннотация.** В связи с большим и постоянно увеличивающимся числом популяционных алгоритмов (П-алгоритмов) глобальной оптимизации актуальной является проблема систематизации их выразительных средств. Рассмотрена одна из составляющих этой проблемы – проблема систематизации эволюционных операторов П-алгоритмов. Представлены постановка задачи глобальной оптимизации и общая схема П-алгоритмов ее решения. Рассмотрены следующие основные эволюционные операторы: инициализация популяции; окончание поиска; кодирование особей; рандомизация; селекция; скрещивание; управление популяцией; локальный поиск. Раскрыта суть этих операторов, приведены варианты использования в различных П-алгоритмах.

**Ключевые слова:** глобальная оптимизация, популяционные алгоритмы, эволюционные операторы.

**Цитирование:** Карпенко А.П. Эволюционные операторы популяционных алгоритмов глобальной оптимизации. Опыт систематизации // Информационные и математические технологии в науке и управлении. 2018. № 2(10). С. 22–30. DOI:10.25729/2413-0133-2018-2-02

**Введение.** В разных публикациях *популяционные алгоритмы* глобальной поисковой оптимизации (П-алгоритмы) называют поведенческими, интеллектуальными, метаэвристическими, вдохновленными (инспирированными) природой, роевыми, многоагентными и т.д. [2]. Известно большое число примеров успешного решения с помощью П-алгоритмов сложных практических задач глобальной оптимизации, например, задач автоматизированного проектирования, синтеза сложных химических соединений, оптимального управления динамическими системами и т.д. (см., например, [7]).

П-алгоритмы многочисленны и весьма разнообразны. Так, в книге [4] представлено более 130 таких алгоритмов, и продолжают появляться новые алгоритмы. Многообразие П-алгоритмов порождает проблему обобщения и систематизации используемых ими базовых принципов, сущностей, в том числе, используемых операторов и операций и т.д. Наш общий подход к систематизации сущностей П-алгоритмов представлен в работе [1]. Данная работа посвящена систематизации важнейших сущностей П-алгоритмов – их *эволюционных операторов*. Все П-алгоритмы, которые упоминаются в работе, можно найти в [2, 4].

### 1. Постановка задачи глобальной оптимизации и основные обозначения.

Рассмотрим *детерминированную задачу глобальной безусловной минимизации*

$$\min_{X \in R^{X1}} f(X) = f(X^*) = f^*, \quad (1)$$

где  $|X|$  - размерность вектора варьируемых параметров  $X = (x_i, i \in [1:|X|])$ ;  $f(X)$  - целевая функция (Ц-функция) со значениями в пространстве  $R^1$ ;  $X^*$ ,  $f^*$  - искомые оптимальные решение и значение Ц-функции соответственно. Полагаем, что поиск решения задачи (1) (то есть инициализация популяции) начинается в  $|X|$ -мерном параллелепипеде  $\Omega_X$ . Другими словами, полагаем, что в этой области осуществляется инициализация популяции.

Рассмотрим также аналогичную задачу глобальной условной минимизации

$$\min_{X \in \Omega \subset R^{|X|}} f(X) = f(X^*) = f^*, \quad (2)$$

где  $\Omega$  - выпуклое множество допустимых значений вектора  $X$ .

Полагаем, что фитнес-функция  $\varphi(X)$  (Ф-функция) П-алгоритма также подлежит минимизации.

В общем виде оператор над популяцией  $S$ , полагаем, имеет вид

$$O(Q, P) \Big|_{A_{identification}^{system.control}} = S',$$

где  $O$  - наименование оператора;  $A$  - алгоритм, реализующий данный оператор;  $Q, P$  - наборы, возможно пустые, аргументов и свободных параметров соответственно. Полагаем, что набор  $Q$  включает в себя информацию, идентифицирующую популяцию  $S$ . Строки *system*, *control* определяют оператор на двух верхних классификационных уровнях. Строка *identification*, в общем случае являющаяся составной, идентифицирует оператор с необходимым уровнем детализации. Строка *system* определяет, является ли данный оператор детерминированным (*deterministic*) или стохастическим (*stochastic*), а строка *control* - статическим (*static*), динамическим программным (*program*) или динамическим адаптивным (*adaptive*) соответственно. Говоря более строго, имеют место следующие соотношения:

$$system \in \{det, stoch\}; \quad control \in \{stat, prog, adapt\}.$$

**2. Инициализация и окончание поиска.** Для операторов инициализации (*INITialization*) популяции используем общие обозначения вида

$$INIT(Q, P) \Big|_{A_{identification}^{system.control}} = \Xi(0) = \Xi^0.$$

Преимущественно, в П-алгоритмах используют равномерное случайное начальное распределение особей в параллелепипеде  $\Omega_X$  (стратегия «дробовика» [3]). Некоторые П-алгоритмы используют также равномерное детерминированное начальное распределение. Так, в алгоритме растущих деревьев (*Saplings Sowing and Growing up, SSG*) на этапе инициализации (посадки) особи (саженцы) случайным образом равномерно располагают в области поиска - создают равномерный сад (*uniform garden*) [5]. Области, в которых осуществляется инициализация других сущностей П-алгоритма, также чаще всего имеют форму параллелепипеда соответствующей размерности.

Выделяем следующие группы операторов инициализации:

$INIT(Q, P) \Big|_{A_{identification}^{det.static}} = S(0) = S^0$  - статический оператор, реализующий детерминированное распределение особей в параллелепипеде  $\Omega_X$ ;  $INIT(Q, P) \Big|_{A_{identification}^{stoch.static}} = S(0) = S^0$  - статический оператор случайного распределения особей в том же параллелепипеде.

Оператор завершения (окончания) поиска (итераций, эволюции популяции) имеет общий вид

$$END(Q, P) \Big|_{A_{identification}^{system.control}} = X(\hat{t}),$$

где  $\hat{t}$  - номер последней итерации. Алгоритм оператора  $END$  реализует следующие действия: проверка выполнимости критерия (или критериев) окончания поиска (условий сходимости итераций); возврат требуемых результатов решения задачи оптимизации (1), если этот критерий (критерии) выполнены; возврат в нужную точку алгоритма и продолжение поиска в противном случае. В качестве критериев окончания поиска в П-алгоритмах чаще всего используют следующие критерии: достижение максимально допустимого числа поколений эволюции популяции  $\hat{t}$  (группа операторов  $END.A_{t...}^{det.static}$ ); стагнация (*stagnation*) итерационного процесса по значениям  $\Phi$ -функции или по значениям компонентов вектора  $X$  (группы операторов  $END.A_{stagn\Phi...}^{det.static}$ ,  $END.A_{stagnX...}^{det.static}$ ; достижение заданного состояния популяции (группа операторов  $END.A_{state...}^{det.static}$ ).

**3. Кодирование (представление) особей.** В терминах генетического алгоритма, в подавляющем большинстве П-алгоритмов используют монохромосомное вещественное кодирование или представление (*coding, representation*) особей. Иногда с целью рандомизации и диверсификации поиска применяют, как вспомогательное, бинарное кодирование на основе обычных двоичных кодов или двоичных кодов Грея [2].

Среди операторов кодирования выделяем группы статических и динамических (программных и адаптивных) операторов, хотя нам известны только примеры использования в П-алгоритмах статических операторов кодирования. Таким образом, используем общее обозначение операторов кодирования вида

$$CODE(Q, P) \Big|_{A_{identification}^{system.control}} = X^{repres},$$

где в случае бинарного кодирования  $repres \in \{binary, Gray\}$ , а в случае вещественного –  $repres = real$  или  $repres = EMPTY$ .

**4. Рандомизация поиска.** Кроме специальных рандомизирующих операторов, делать поиск случайным могут рандомизирующие операторы скрещивания, селекции, локального поиска и другие операторы – опосредованная рандомизация. Идея рандомизации текущего состояния данной особи заключается в замене одного или нескольких компонентов ее вектора состояния  $V$  (генов - в генетическом алгоритме) их случайными значениями. Реализация этой идеи состоит из трех этапов: выбор подлежащих рандомизации особей; выбор подлежащих рандомизации компонентов состояния отобранных особей; случайное изменение этих компонентов по тому или иному правилу. Оператор рандомизации реализует все эти этапы или, по меньшей мере, третий этап.

Общее описание бинарных и вещественных операторов рандомизации имеет, соответственно, вид

$$RND(Q, P) \Big|_{A_{binary...}^{system.control}} = X^{binary}, \quad RND(Q, P) \Big|_{A_{real.determinaton...}^{system.control}} = X'.$$

При этом рандомизации могут подлежать не только компоненты вектора  $X$ , как указано в данных записях, но и любые иные компоненты вектора состояний особи.

**5. Селекция.** В отличие от [2] не различаем операторы отбора и селекции: полагаем, что операторы селекции предназначены как для выбора особей из текущей популяции с целью их последующей рандомизации, скрещивания и для выполнения других операций, так и для выбора особей с целью формирования следующей популяции. Поскольку операторы селекции, в отличие от операторов рандомизации и скрещивания, инвариантны к схеме кодирования особей, имеется возможность не выделять используемую схему кодирования.

Можно предложить несколько классификаций методов селекции и реализующих их операторов. Во-первых, по общим правилам выделяем детерминированные и стохастические операторы селекции (уровень иерархии *system*). Во-вторых, различаем статические и динамические операторы селекции (уровень *control*). Как правило, динамические операторы в процессе поиска изменяют метод отбора так, чтобы на начальных итерациях обеспечить, в первую очередь, диверсификацию поиска, а на завершающих итерациях – его интенсификацию.

На третьем уровне иерархии выделяем следующие группы операторов селекции (*SeLeCtion*) [1]: операторы  $SLC|A_{no-space..}^{system.control}$ , не учитывающие близость особей в том или ином из пространств; операторы  $SLC|A_{\Phi-space..}^{system.control}$ ,  $SLC|A_{X-space..}^{system.control}$ ,  $SLC|A_{T-space..}^{system.control}$ , использующие оценки близости особей в  $\Phi$ -пространстве  $R_{\Phi}$ , пространстве поиска  $R_X$  и в топологическом пространстве  $R_T$  соответственно; комбинированные операторы  $SLC|A_{multi-space..}^{system.control}$  на основе близости особей в нескольких из указанных пространств.

Операторы селекции на основе близости особей в данном пространстве могут быть построены с использованием метрик близости особей в этих пространствах, например, отобранными можно считать все особи, принадлежащие некоторой окрестности  $\Omega^{R_x}(s_i)$  особи  $s_i$  [1]. Наиболее известными методами (статическими) селекции на основе близости приспособленностей особей (то есть в пространстве  $R_{\Phi}$ ) являются: метод рулетки или метод пропорционального отбора (*proportional sampling*) и его различные модификации; метод турнирного (*tournament*) отбора; метод рангового (*rank-based*) отбора, метод отбора на основе элитизма (*elitism-based*) [2]. Приспособленность особей при этом может вычисляться не только на основе исходной  $\Phi$ -функции  $\varphi(X)$ , но и на основе аддитивной свертки одной или нескольких вспомогательных  $\Phi$ -функций. Подобный подход использует, например, алгоритм оптимизации роем светлячков (*Glowworm swarm optimization, GSO*) [6]. К методам селекции на основе близости особей в пространстве соседства  $R_T$  относится метод отбора вытеснением (*crowding*) [2]. Комбинированные операторы селекции используют близость особей в нескольких пространствах одновременно, например, одна из модификаций метода турнирного отбора предполагает, что объединения особей формируются не случайным образом, а в зависимости от их близости в пространстве поиска  $R_X$  [2].

**6. Скрещивание.** В П-алгоритмах оператор скрещивания или репродукции (*crossing, reproduction*) является одним из основных эволюционных операторов. Суть скрещивания заключается в создании особей-потомков из нескольких особей-родителей путем их рекомбинации (*recombination*). Оператор скрещивания реализует процесс, посредством которого особи, имеющие более высокую приспособленность, получают большую возможность для воспроизводства (репродукции) потомков, чем «худшие» особи. В П-

алгоритмах скрещивание может быть использовано для моделирование социального поведения особей (см., например, алгоритм роя части (*Particle Swarm Optimization, PSO* [2])).

Операторы скрещивания не инвариантны к используемому методу кодирования особей. Поэтому по отдельности рассматриваем бинарные и вещественные операторы.

Используем следующее общее обозначение операторов бинарного и вещественного скрещивания (*CRoSsing*) соответственно:

$$CRS(Q, P) \Big|_{A_{binary}^{system.control}} = X'_{j_1}{}^{binary}, X'_{j_2}{}^{binary}, \dots, X'_{j_m}{}^{binary}, \{X_{i_1}{}^{binary}, X_{i_2}{}^{binary}, \dots, X_{i_n}{}^{binary}\} \in Q;$$

$$CRS(Q, P) \Big|_{A_{real}^{system.control}} = X'_{j_1}, X'_{j_2}, \dots, X'_{j_m}, \{X_{i_1}, X_{i_2}, \dots, X_{i_n}\} \in Q.$$

Полагаем, что вектор  $P$  включает в себя величины  $n, m$  – числа скрещиваемых (родительских) особей и особей-потомков соответственно.

Среди вещественных операторов скрещивания можно выделить группу операторов, которые наряду со скрещиванием производят случайные изменения результирующих особей. Строго говоря, такие операторы следовало бы отнести к группе комбинированных операторов. Мы однако относим рандомизирующие операторы скрещивания к одной из групп операторов скрещивания.

**7. Управление популяцией.** Используем общее обозначение операторов управления (*PoPuLation*) популяцией вида

$$PPL(Q, P) \Big|_{A_{identification}^{system.control}} = S'.$$

Выделяем следующие группы операторов: сжатие (*compression*) популяции  $PPL.A_{compr}^{system.control}$ ; расширение (*expansion*) популяции  $PPL.A_{expan}^{system.control}$ ; ассоциация (*association*) популяций  $PPL.A_{assoc}^{system.control}$ ; репликация (*replication*) популяции  $PPL.A_{repl}^{system.control}$ ; декомпозиция (*splitting*) популяции  $PPL.A_{split}^{system.control}$ ; репродукции (*reproduction*) популяции  $PPL.A_{reprod}^{system.control}$ .

Операторы  $PPL.A_{compr}^{system.control}$  предназначены для продуцирования копии части заданной популяции. Общий формат операторов данной группы имеет вид

$$PPL(Q, P) \Big|_{A_{compr}^{system.control}} = S'; \quad |S| > |S'| = n,$$

где алгоритм оператора определяет правило отбора  $n$  особей популяции  $S$  для помещения их в популяцию  $S'$  и является, по сути, оператором селекции. Таким образом, операторы  $PPL.A_{compr}^{system.control}$  можно интерпретировать как макро-операторы. Варианты оператора отличаются используемым алгоритмом селекции.

Операторы  $PPL.A_{expan}^{system.control}$  увеличивают размер популяции за счет отбора, возможно, с повторами, ее особей:

$$PPL(Q, P) \Big|_{A_{expan}^{system.control}} = S'; \quad |S'| = |S| + n.$$

Здесь, аналогично операторам предыдущей группы, алгоритм оператора реализует некоторый оператор селекции, так что операторы  $PPL.A_{expan}^{system.control}$  также можно отнести к макро-операторам.

Операторы  $PPL.A_{assoc}^{system.control}$  предназначены для объединения популяций  $S_1, S_2, \dots, S_n$  в одну популяцию  $S$  заданной мощности  $m$ :

$$PPL(Q, P) \Big|_{A_{assoc...}^{system.control}} = S, \quad \{S_1, S_2, \dots, S_n\} \in Q, \quad |S| = m.$$

Операторы  $PPL.A_{repl...}^{system.control}$  создают копию  $S_1$  исходной популяции  $S$ , возможно, с той или иной перенумерацией ее особей:

$$PPL(Q, P) \Big|_{A_{repl...}^{system.control}} = S_1; \quad |S_1| = |S|.$$

Операторы  $PPL.A_{split...}^{system.control}$  разбивают популяцию на заданное число подпопуляций или объединений особей. В первом случае общий формат операторов данной группы имеет вид

$$PPL(Q, P) \Big|_{A_{split...}^{system.control}} = \{S_1, S_2, \dots, S_n\}; \quad P = (|S_1|, |S_2|, \dots, |S_n|, |S|),$$

где  $|S_1|, |S_2|, \dots, |S_n|$  - требуемые размеры субпопуляций;  $\sum_{i=1}^n |S_i| = |S|$ . Алгоритм оператора реализует то или иное правило отбора особей популяции  $S$  в субпопуляции.

Оператор  $PPL.A_{reprod...}^{system.control}$  вызывает уничтожение наиболее «слабых» особей и клонирование наиболее «сильных», сохраняя размер исходной популяции:

$$PPL(Q, P) \Big|_{A_{reprod...}^{system.control}} = S_1, \quad P = (n, |S|), \quad |S_1| = |S|, \quad n < |S|.$$

Правила отбора «слабых» и «сильных» особей определяет алгоритм оператора. Один из вариантов оператора репродукции использует, например, алгоритм бактериальной оптимизации (*Bacterial Foraging Optimization, BFO*) [2].

**8. Локальный поиск.** Включение алгоритма локального поиска (мема) в П-алгоритм порождает гибридный П-алгоритм. Общие подходы к гибридизации (*hybridization*) П-алгоритмов рассмотрены, например, в [2]. Гибридизацию П-алгоритма с алгоритмом или алгоритмами локального поиска выполняют по схеме гибридизации вложением. В соответствии с общими принципами этого типа гибридизации выделяем низкоуровневое (*low-level*) вложение, высокоуровневое (*high-level*) вложение одного алгоритма (*one-meme*) локального поиска, высокоуровневую мультимемеевую (*multi-memes*) адаптивную гибридизацию.

Низкоуровневая гибридизация вложением предполагает сращивание гибридизируемых алгоритмов так, что, по сути, ее результатом является новый алгоритм, который не может быть декомпозирован на составляющие его алгоритмы. Основная идея такой гибридизации заключается в модификации эволюционных операторов П-алгоритма на основе использования алгоритма или алгоритмов локальной оптимизации [2]. Например, низкоуровневая гибридизация вложением генетического алгоритма может быть осуществлена путем модификации операторов селекции или рандомизации встраиванием в них какого-либо алгоритма локальной оптимизации. Поскольку низкоуровневая гибридизация вложением приводит, по сути, к появлению новых эволюционных операторов, модифицированные операторы следует рассматривать в разделах, посвященных этим операторам.

Используем следующее общее обозначение гибридизируемых операторов локального (*LOCAL*) поиска, связанных с особью  $s_i$ :

$$LOC(Q, P) \Big|_{A_{hybrid...}^{system.control}} = \tilde{s}_i, \quad i \in [1:|S|], \quad s_i \in Q,$$

где  $hybrid \in \{meme, memes\}$ .

В настоящее время известны примеры высокоуровневой гибридизации вложением практически всех известных П-алгоритмов со многими классическими и неклассическими алгоритмами локальной оптимизации.

**Пример.** Стохастический статический (не адаптивный) одно-мемеевый алгоритм, используемый в алгоритме летучих мышей (*Bat-Inspired Algorithm, BIA*) [8].

$$LOC(s_i, P) | A_{meme.BIA}^{stoch.static} = \tilde{s}_i, \quad i \in [1:|S|], \quad P = (\bar{a}, |X|).$$

Здесь  $\bar{a}$  - текущее среднее значение громкостей  $a_i$  всех особей (мышей) популяции. Алгоритм оператора определяет следующая последовательность шагов [2]:

1) случайным образом варьируем текущее положение мыши  $s_i$  в соответствии с формулой

$$X'_i = X_i + \bar{a} U_{|X|}(-1; 1),$$

где  $U_{|X|}(-1; 1)$  -  $|X|$ -мерный вектор некоррелированных случайных вещественных чисел, равномерно распределенных в интервале  $(-1; 1)$ ;

2) вычисляем значение Ф-функции  $\varphi(X'_i) = \varphi'_i$  в новой точке  $X'_i$ ;

3) если  $\varphi'_i > \varphi_i$ , то завершаем процедуру локального поиска, в противном случае фиксированное число раз возвращаемся к шагу 1.

В современных П-алгоритмах широко используют мультимемеевые адаптивные алгоритмы. Для идентификации такого алгоритма необходимо определить: а) используемый набор (рой) мемов (среди которых могут быть как детерминированные, так и стохастические), б) стратегию использования того или иного мема из роя доступных мемов, то есть, гиперэвристику (*hyper heuristic*) алгоритма; характеристики мемов и/или текущего исследуемого фрагмента области решения, на основе которых производится их выбор. Наиболее известны три следующие категории гиперэвристик, используемых в мультимемеевых алгоритмах - случайные (*random*), жадные (*greedy*), гиперэвристики с функцией выбора (*choice-function hyper heuristics*) [2].

Таким образом, используем общее обозначение мультимемеевого оператора локального поиска вида

$$LOC(Q, P) | A_{memes.h-heur...}^{system.adapt} = \tilde{s}_i, \quad i \in [1:|S|], \quad s_i \in Q,$$

где  $h-heur \in \{random, greedy, choice - funct\}$ .

Отметим следующее обстоятельство. Наряду с высокоуровневой мультимемеевой адаптивной гибридизацией известна мультимемеевая само-адаптивная гибридизация. Само-адаптирующиеся мультимемеевые алгоритмы представляют собой меметическую модификацию генетического алгоритма. Каждую особь  $s_i, i \in [1:|S|]$  популяции  $S$  в этом случае кодируют мультихромосомой  $\bar{X}_i = (X_i, m_{i_k}), i_k \in [1:|M|]$ , первая часть которой представляет собой соответствующее решение исходной задачи оптимизации, а вторая часть - номер мема в наборе  $M$ , с помощью которого получено данное решение. По общему правилу эволюции мультихромосом генетические операторы применяют к каждой из указанных частей хромосомы в отдельности, не скрещивая эти части между собой. Таким

образом, данные алгоритмы реализуют метод самоадаптивного управления [2]. Мы относим мультимемеевую само-адаптивную гибридизацию к низкоуровневой гибридизации вложением и по общему правилу считаем необходимым рассматривать в разделах, посвященных соответствующим поисковым операторам.

**Заключение.** В работе предложена многоуровневая классификация эволюционных операторов П-алгоритмов глобальной оптимизации, которая на верхнем уровне иерархии выделяет следующие операторы: инициализация популяции и окончание поиска; кодирование особей; рандомизация; селекция; скрещивание; управление популяцией; локальный поиск. Эти операторы на следующем уровне подразделяются на детерминированные и стохастические. Далее различаем статические, динамические программные и динамические адаптивные операторы. Следующие классификационные уровни являются «операторозависимыми», то есть, вообще говоря, различны для каждого из операторов.

Поскольку мы используем такие наименования операторов, как селекция, скрещивание, может создаться впечатление, что предлагаемая классификация ориентирована только на эволюционные алгоритмы. Однако это не так. Представленная в работе схема описания операторов может быть использована для определения любых П-алгоритмов. Привести соответствующие примеры не позволяет только ограничение на объем статьи.

В развитие работы автор планирует расширить представленный набор операторов и, главное, с помощью этого набора и набора других сущностей популяционных алгоритмов [1] систематизировать наиболее известные алгоритмы этого класса.

Работа поддержана грантом РФФИ (проект № 16-07-00287).

#### СПИСОК ЛИТЕРАТУРЫ

1. Карпенко А.П. Основные сущности популяционных алгоритмов глобальной оптимизации. Опыт систематизации // Интернет-журнал «Науковедение». 2017. Том 9. №6. Режим доступа: <https://naukovedenie.ru/PDF/46TVN617.pdf> (дата обращения 05.01.2018).
2. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы вдохновленные природой. М.: Издательство МГТУ им. Н.Э. Баумана. 2014. 446 с.
3. Скобцов Ю.А., Сперанский Д.В. Эволюционные вычисления: Учебное пособие. М.: Национальный Открытый Университет «ИНТУИТ». 2012. 331с.
4. Bo Xing, Wen-Jing Gao. Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms. Springer International Publishing Switzerland. 2014. 451 p.
5. Karci A. Theory of Saplings Growing Up Algorithm / ICANNGA 2007. Part I. LNCS 4431. Springer-Verlag. Berlin Heidelberg. 2007. Pp. 450–460.
6. Krishnanand K.N., Ghose D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions // Swarm Intelligence. 2009. no. 3. Pp. 87–124.
7. Onwubolu G.C., Babu B.V. New Optimization Techniques in Engineering. Springer-Verlag. Berlin. Heidelberg. 2004. 712 p.
8. Yang X.-S. A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010). Studies in Computational Intelligence. Berlin: Springer. 2010. Vol. 284. Pp. 65–74.



## EVOLUTIONARY OPERATORS OF POPULATION-BASED ALGORITHMS FOR GLOBAL OPTIMIZATION. EXPERIENCE OF SYSTEMATIZATION

Anatoly P. Karpenko

Dr, Professor, head of chair «Computer Aided Design»,  
Bauman Moscow State Technical University, 105005 г. Moscow, 2-nd Baumanskaya str 5,  
e-mail: akarpenko@bmstu.ru

**Abstract.** In connection with the large and constantly increasing number of population-based algorithms (P-algorithms) for global optimization, the problem of systematizing their expressive means is topical. We consider one of the components of this problem - the problem of ordering the evolutionary operators of P-algorithms. We present the formulation of the global optimization problem and the general scheme of the P-algorithms for its solution. We consider the following main evolutionary operators: initialization of the population and the end of the search; coding of individuals; randomization; selection; crossing; control population; local search. We reveal the essence of these operators and give variants of use in various P-algorithms.

**Keywords:** Global optimization, population-based algorithms, evolutionary operators.

### References

1. Karpenko A.P. Osnovnyie suschnosti populyatsionnyih algoritmov globalnoy optimizatsii. Opyit sistemizatsii [The main essentials of population-based algorithms of global optimization. Systematization experience] // Internet-zhurnal «Naukovedenie» = Internet-journal "Naukovedenie". 2017. Tom 9. no 6. Available at: <https://naukovedenie.ru/PDF/46TVN617.pdf> (accessed 05.01.2018). (in Russian)
2. Karpenko A.P. Sovremennyye algoritmyi poiskovoy optimizatsii. Algoritmyi vdoh-novlennyye prirodoy [Modern algorithms of search optimization. Algorithms inspired by nature.]. Moscow. Izdatelstvo MGTU im. N.E. Baumana = Bauman MSTU Publ. 2014. 446 p. (in Russian)
3. Skobtsov Yu.A., Speranskiy D.V. Evolyutsionnyie vyichisleniya: Uchebnoe posobie [Evolutionary Computing: A Tutorial]. Moscow. Natsionalnyiy Otkryityiy Universitet «INTUIT» = National Open University "INTUIT". 2012. 331 p. (in Russian)
4. Bo Xing, Wen-Jing Gao. Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms. Springer International Publishing Switzerland. 2014. 451 p.
5. Karci A. Theory of Saplings Growing Up Algorithm / ICANNGA 2007. Part I. LNCS 4431. Springer-Verlag. Berlin Heidelberg. 2007. Pp. 450–460.
6. Krishnanand K.N., Ghose D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions // Swarm Intelligence. 2009. no. 3. Pp. 87–124.
7. Onwubolu G.C., Babu B.V. New Optimization Techniques in Engineering. Springer-Verlag. Berlin. Heidelberg. 2004. 712 p.
8. Yang X.-S. A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010). Studies in Computational Intelligence. Berlin: Springer. 2010. Vol. 284. Pp. 65–74.