

УДК 529.6

Типовые структуры популяционных алгоритмов глобальной оптимизации

Карпенко Анатолий Павлович apkarpenko@bmstu.ru

Кузьмина Инна Анатольевна kuzminaia@bmstu.ru

МГТУ им. Н.Э. Баумана, Москва, Россия

Рассмотрена структура популяционных алгоритмов глобальной оптимизации на уровне популяции. Представлены следующие структурные классы этих алгоритмов: одноуровневые однопопуляционные; одноуровневые многопопуляционные; двухуровневые; трех- и более уровневые алгоритмы. В каждом из указанных классов на верхнем уровне иерархии выделены стационарные алгоритмы, программно-динамические алгоритмы, параметрически и структурно адаптивные динамические алгоритмы, а также адаптивные динамические алгоритмы, в которых средства адаптации явно не выделены.

Ключевые слова: глобальная оптимизация, популяционные алгоритмы, структура алгоритма,

Введение. Рассматриваем детерминированную стационарную задачу глобальной условной оптимизации

$$\min_{X \in D_X \subset \mathbb{R}^{|X|}} f(X) = f(X^*) = f^*$$

или аналогичную задачу безусловной оптимизации. Здесь приняты обозначения: $X \in \mathbb{R}^{|X|}$ — вектор варьируемых параметров; $f(X) \in \mathbb{R}^1$ — целевая функция; X^*, f^* — искомые оптимальное решение и соответствующее значение целевой функции; D_X — выпуклое множество допустимых значений вектора X .

Полагаем, что для решения поставленной задачи оптимизации применяются *популяционные* алгоритмы глобальной (непрерывной) оптимизации, которые принадлежат классу прямых (не использующих производные целевой функции) стохастических поисковых алгоритмов. В разных публикациях эти алгоритмы называют поведенческими, интеллектуальными, метаэвристическими, вдохновленными (инспирированными) природой, роевыми, много-агентными, популяционными и т. д. [1, 2].

Популяционные алгоритмы (П-алгоритмы) предполагают одновременную обработку нескольких вариантов решения задачи оптимизации и представляют собой альтернативу классическим *траекторным* поисковым алгоритмам, в которых в области поиска эволюционирует только один кандидат в решение этой задачи [3].

Говорить о структуре П-алгоритмов можно на различных иерархических уровнях: на уровне популяции; на уровне структуры субпопуляционных ал-

горитмов; на уровне структуры макрооператоров и операторов; на уровне итерационной формулы и т. д. [1]. На уровне популяции различаем следующие структурные классы П-алгоритмов:

- одноуровневые однопопуляционные алгоритмы;
- одноуровневые многопопуляционные алгоритмы;
- двухуровневые алгоритмы;
- трех- и более уровневые алгоритмы.

Одноуровневые однопопуляционные и многопопуляционные алгоритмы. К одноуровневым П-алгоритмам принадлежат однопопуляционные гибридные алгоритмы, построенные путем объединения какого-либо П-алгоритма и непопуляционного алгоритма локального поиска.

На верхнем уровне иерархии среди одноуровневых алгоритмов выделяем:

- стационарные алгоритмы;
- программно-динамические алгоритмы;
- параметрически и структурно адаптивные динамические П-алгоритмы;
- адаптивные динамические алгоритмы, в которых средства адаптации

явно не выделены.

На следующем уровне иерархии различаем:

- однофазные (один макрооператор) алгоритмы;
- двухфазные (два макрооператора) алгоритмы;
- трех- и более фазные (три и более макрооператоров) алгоритмы.

Примерами *стационарных одноуровневых однопопуляционных однофазных* алгоритмов являются базовый однопопуляционный алгоритм (BOA), алгоритм поиска системой зарядов (CSS), алгоритм оптимизации роем частиц (PSO). К аналогичным *двухфазным* алгоритмам принадлежат гибридный алгоритм непрерывно взаимодействующей муравьиной колонии (НСИАС), гусиный алгоритм оптимизации (GOA), алгоритм опыления цветов (FPA) и другие алгоритмы. Один из самых известных П-алгоритмов — генетический алгоритм, относится к трехфазным алгоритмам [1].

К *программно-динамическим алгоритмам* относим, например, алгоритм оптимизации роем пчел BSO [1].

Примером *динамического адаптивного алгоритма* является оптимизационный алгоритм, основанный на модели атмосферных облаков АСМО [1].

Двухуровневые алгоритмы. В качестве примера алгоритмов данного класса представляем трехпопуляционный алгоритм пчелиной оптимизации OptBees. Алгоритм использует три типа особей: *вербовщики* S^r ; *разведчики* S^s ; *завербованные* $S_h = \{S_i^h\}$. Кроме того, в алгоритме используются особи *не-вербовщики* $\bar{S}^r$, совокупность которых в процессе эволюции популяции расщепляется на некоторое число субпопуляций завербованных особей $\{S_i^h\}$ и субпопуляцию особей-разведчиков S^s (рис. 1). Отличительными особенностями алгоритма OptBees является популяция переменной мощности и сложная схема управления ею.

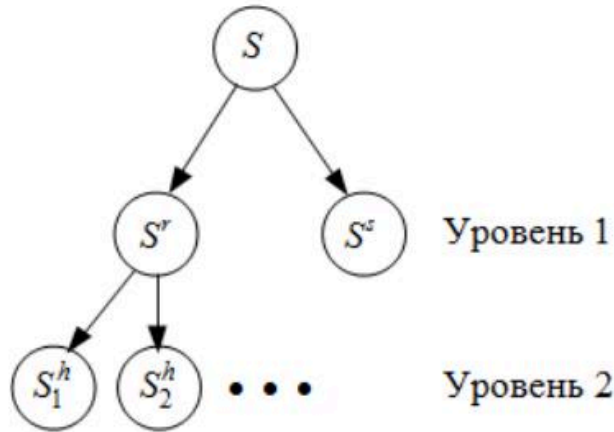


Рис. 1. Структура алгоритма OptBees:

S, S^s — субпопуляции особей-вербовщиков и особей разведчиков соответственно;
 $\{S_i^h\}$; — субпопуляции завербованных особей, так что S_i^h — совокупность особей $(S_{i,j}^h)$,
 завербованных особью-вербовщиком $s_i^r \in S^r$

Схема алгоритма OptBees имеет следующий вид.

1) Расщепление популяции. Расщепляем популяцию S на субпопуляции $S^r, \bar{S}^r$, относя особь $s_i \in S$ к субпопуляции S^r с вычисляемой вероятностью ξ_i и в противном случае – к субпопуляции $\bar{S}^r: S \rightarrow \{S^r, \bar{S}^r\}$.

2) Переупорядочение популяции. Последовательно для каждой из особей $s_i, i \in [1: |S^r|]$, начиная с лучшей, находим в шаровой окрестности заданного радиуса с центром в точке X_i^r всех особей $\{s_j^r\}$ и изменяем их статус на «не-вербовщик», то есть исключаем из субпопуляции S^r и включаем в субпопуляцию $\bar{S}^r: \{S^r, \bar{S}^r\} \rightarrow \{S'^r, \bar{S}\}$.

3) Переформирование популяции 1. Исходя из заданных максимальной мощности популяции $|S|$ тах и желаемого числа не-вербовщиков, для каждого вербовщика вычисляем текущее желаемое число m особей популяции S . Если $m > |S'|$, то по правилам инициализации начальной популяции создаем в пространстве поиска новых особей числом $(m - |S'|)$, а если $m < |S'|$, то удаляем из популяции S' худших особей числом $(|S'| - m): S' \rightarrow S''$.

4) Переформирование популяции 2. Находим текущее число не-вербовщиков $\bar{m}^r = m^h$, которые могут быть завербованы, а также число особей-разведчиков m^s . Связываем с каждым из вербовщиков $s_i^{r'}$ число ближайших (в евклидовой метрике пространства $\mathbb{R}^{|X|}$) не-вербовщиков, образующих субпопуляцию S_i^h особей, завербованных особью $s_i^{r'}: \bar{S}^r$.

5) Глобальный поиск. Перемещаем в пространстве поиска особей каждой из субпопуляций S_i^h в случайно возмущенном направлении соответствующей особи-вербовщика по одношаговым одноточечным итерационным формулам, а особей-разведчиков – в новые точки пространства поиска, созданные по правилам инициализации начальной популяции: $\{S_i^h\} \rightarrow \{S_i'^h\} = \{S_i^h(t+1)\}; S^s \rightarrow S'^s = S^s(t+1)$.

б) Локальный поиск. В окрестности каждой из особей субпопуляций S^r , $\{S_i^h\}$, S^s выполняем фиксированное число итераций оригинального алгоритма локального поиска [2].

Трехуровневые алгоритмы. Примерами алгоритмов этого класса являются следующие динамические адаптивные алгоритмы: алгоритм круговорота воды в природе WCO; алгоритм общества и цивилизации SCA; алгоритм империалистической конкуренции ICA. Рассмотрим последний из указанных алгоритмов.

В алгоритме ICA используются субпопуляции *империй* S^{emp} и *колоний* S^{col} . На этапе инициализации популяции, особей последней субпопуляции «связываем» с особями субпопуляции S^{emp} , так что S_i^{col} – субпопуляция таких особей, «связанных» с особью $s_i \in S^{emp}$; $|S_i^{col}| = n_i$. Связи субпопуляций алгоритма ICA иллюстрирует рис. 2.

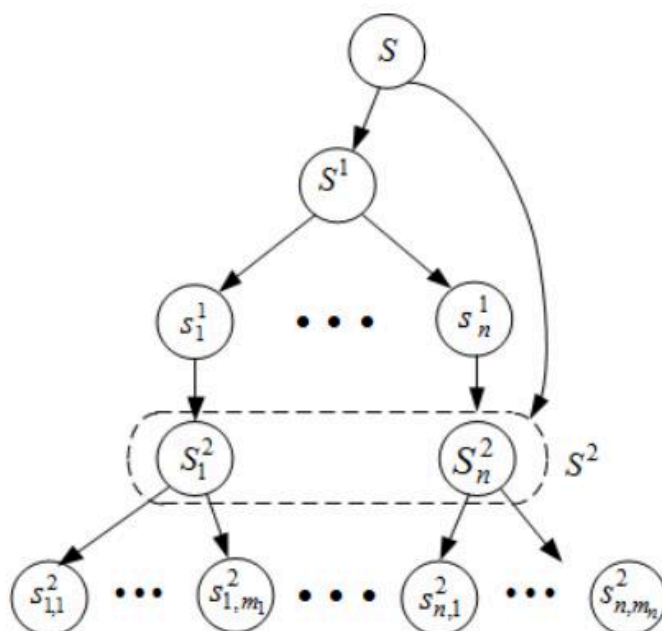


Рис. 2. Схема связей субпопуляций в алгоритме ICA

Схема алгоритма ICA имеет следующий вид.

1) Глобальный поиск. Перемещаем особей каждой из субпопуляций $S_i^{col} \subset S^{col}$ в случайно возмущенном направлении наиболее приспособленной особи субпопуляции S^{emp} : $S^{col} \rightarrow S'^{col}$.

2) Частичная «встряска» популяции. В каждой из субпопуляций $S_i'^{col}$ равномерно случайно удаляем фиксированный процент особей и заменяем их таким же числом новых особей, сгенерированных по правилу инициализации начальной популяции: $S'^{col} \rightarrow S''^{col}$.

3) Замена. В каждой из субпопуляции $S_i''^{col}$ находим лучшую особь s''_{i,j^*} , и если $\varphi''_{i,j^*} > \varphi_i$, то меняем местами указанных особей: $S^{emp} \rightarrow S'^{emp}$; $S''^{col} \rightarrow S'''^{col}$.

4) Формирование новой популяции. Выбираем из субпопуляции S'''^{col} заданное число наиболее «слабых» особей и перераспределяем их между империями S_i^{emp} с вероятностью, пропорциональной нормализованной «силе» империи [1]. Если мощность некоторого полученного набора S_i^{emp} оказывается равной нулю, то удаляем империю S_i^{emp} из субпопуляции S^{emp} : $S^{emp} \rightarrow S^{emp}(t+1)$; $S'''^{col} \rightarrow S^{col}(t+1)$.

Заключение. Исчерпывающий анализ П-алгоритмов вряд ли возможен в силу того, что новые алгоритмы и модификации известных алгоритмов продолжают появляться и общее число их вместе с модификациями чрезвычайно велико [4]. Широкий набор П-алгоритмов позволяет осуществить предварительный выбор (быть может, с помощью методов метаоптимизации) наиболее эффективного алгоритма оптимизации, учитывающего особенности конкретной оптимизационной задачи. Методологической основой этой технологии является известная теорема о бесплатных завтраках (No Free Lunch theorem), из которой следует, что если алгоритм a_1 эффективнее алгоритма a_2 при решении некоторой задачи, то обязательно найдется задача, для которой алгоритм a_2 окажется эффективнее алгоритма a_1 [5].

Список источников

- [1] Карпенко А.П. *Анализ и синтез популяционных алгоритмов глобальной оптимизации*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2024, т. 1, 368 с., т. 2, 387 с.
- [2] Пантелеев А.В., Метлицкая Д.В., Алешина Е.А. *Методы глобальной оптимизации: метаэвристические стратегии и алгоритмы*. Москва, Вузовская книга, 2013, 244 с.
- [3] Гладков Л.А., Курейчик В.В., Курейчик В.М. и др. *Биоинспирированные методы в оптимизации*. Москва, Физматлит, 2009, 384 с.
- [4] Xing B., Gao W.-J. *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*. Springer Cham Heidelberg, New York, Dordrecht, London, 2014, 451 p.
- [5] Wolpert D.H., Macready W.G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1997, no. 1 (1), pp. 67–82.

Typical structures of global optimization population algorithms

Karpenko Anatoly P. apkarpenko@bmstu.ru

Kuzmina Inna A. kuzminaia@bmstu.ru

BMSTU, Moscow, Russia

We consider the structure of population-based global optimization algorithms at the population level. We present the following structural classes of these algorithms: single-level single-population; single-level multi-population; two-level; three- and more-level algorithms. In each of these classes, at the top level of the hierarchy, we distinguish stationary algorithms, software-dynamic algorithms, parametrically and structurally adaptive dynamic algorithms, as well as adaptive dynamic algorithms in which adaptation tools are not explicitly allocated.

Keywords: global optimization, population algorithms, structural synthesis