

Родзин С.И., Курейчик В.В.

## СОСТОЯНИЕ, ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ РАЗВИТИЯ БИОЭВРИСТИК

*Исследование выполнено при финансовой поддержке государственного задания (проектная часть) в сфере научной деятельности Министерства образования и науки Российской Федерации (проект № 8.823.2014/К) в Южном федеральном университете.*

**Аннотация:** Предметом обзора является современное состояние, проблемные вопросы и перспективные области исследований биоэвристик для решения оптимизационных задач. Биоэвристики – это математические преобразования, трансформирующие входной поток информации в выходной и основанные на правилах имитации механизмов эволюции, природных аналогий, на статистическом подходе к исследованию ситуаций и итерационном приближении к искомому решению. В настоящее время биоэвристики превратились в важный инструмент поиска близких к оптимальным решений задач, которые до этого считались неразрешимыми. Методологической и теоретической основой обзорного исследования являлись методы оптимизации и поддержки принятия оптимальных решений, искусственный интеллект, теория эволюционных вычислений. В статье анализируются фундаментальные результаты, полученные в области биоэвристических алгоритмов оптимизации: теорема Холланда и NFL-теорема. Устанавливаются закономерности и структура биоэвристик, особенности кодирования решений, базовый цикл биоэвристических алгоритмов. Рассматривается перспективное направление в анализе времени работы когнитивных биоэвристических алгоритмов - анализ дрейфа.

**Ключевые слова:** когнитивный биоинспирированный алгоритм, метаэвристика, оптимизация, эволюционные вычисления, эволюционный оператор, NFL-теорема, анализ дрейфа, функция приспособленности, моделирование, программирование

**Abstract:** The subject of the article is the current state and problematic issues of such promising field of research as bioheuristics. Authors understand the term bioheuristics in context of optimization problems solving. Bioheuristics are mathematical transformations of the input stream to the output data based on simulation mechanisms of evolution, natural analogies, as well on a statistical approach to the study of situations and iterative approximation to the desired solution. Currently, bioheuristics have become an important tool for finding solutions that are close to the optimal, and in problems, which were considered unsolvable earlier. The methodological and theoretical basis of the scoping study are optimization techniques and decision making support methods, artificial intelligence, evolutionary computation theory. The article analyzes the

*fundamental results obtained in the field of bioheuristic optimization algorithms, such as Holland theorem and NFL-theorem. The article establishes patterns and structure of bioheuristics, especially coding solutions, basic cycle of bio heuristics algorithms. The study reviews a promising direction in the analysis time of the biological cognitive heuristics, i.e. drift analysis.*

**Keywords:** *evolution operator, evolutionary computation, optimization, metaheuristics, cognitive bioinspired algorithm, NFL-theorem, drift analysis, fitness function, programming, modeling*

## Введение

Многие процессы в науке и технике, экономике и бизнесе формулируются как задача оптимизации: минимизировать время, стоимость, риск или максимизировать прибыль, производительность, эффективность. Перспективным классом приближенных методов, применимых к разнообразным оптимизационным задачам являются биоэвристические алгоритмы или, просто, биоэвристики. Биоэвристики – это алгоритмические приёмы, которые позволяют ограничить перебор, и основаны на правилах моделирования процессов эволюции, природных аналогиях, на статистике и итерационном поиске решения задачи. Целью работы является анализ фундаментальных результатов, полученных в области разработки биоэвристик, закономерностей, лежащих в их основе, особенностей кодирования решений и базового цикла вычислений, а также оценка времени работы биоэвристик.

## Обзор фундаментальных результатов в области биоэвристик

Наиболее значительный вклад в разработку когнитивных биоинспирированных методов внесли следующие ученые: *Л. Растрингин* предложил методы случайного поиска [1], *Nelder* и *Mead* представили эвристики, которые для некоторых задач сходятся в нестационарных точках [2], *Fogel* и др. разработали алгоритм эволюционного программирования [3], *Kernighan* и *Lin* создали метод поиска в глубину [4], *Holland* предложил генетический алгоритм [5], *Smith* разработал алгоритм генетического программирования [6], *Kirkpatrick* и др. предложили метод имитации отжига [7], *Glover* разработал алгоритм табуированного поиска [8], *Moscato* представил меметический алгоритм [9], *Dorigo* предложил муравьиный алгоритм [10], *Wolpert* и *Macready* доказали NFL-теорему [11]. Большой вклад в развитие теории и практики биоэвристик внесли научные школы *Д. Батищева*, *И. Букатовой*, *Д. Голдберга*, *Ж. Гонкальвеса*, *Б. Доерра*, *В. Емельянова*, *К. Игеля*, *К. де Йонга*, *В.М. Курейчика*, *Р. Мииккулайнена*, *С. Нолфи*, *И. Норенкова*, *В. Редько*, *Л. Уитли*, *К. Феррейры*, *Д. Флореано*, *Н. Хансена*, *Дж. Шапиро*, *З. Яо*.

В информатике и математической оптимизации теоретические вопросы создания биоэвристик весьма разнообразны: сходимости алгоритмов; наличие стратегий, которые направляют процесс поиска оптимума; наличие цели для эффективного исследования пространства поиска оптимальных решений; оценка временной сложности алгоритмов и др.

Существуют самые различные классификации биоэвристик [12]. Например, классифи-

кация по типу стратегии поиска: улучшение простых локальных алгоритмов поиска, таких как моделирование отжига, поиск с запретами, локальный поиск с возвратами, поиск в переменной окрестности и др.; обучение в ходе поиска, например, алгоритмы колонии муравьев, эволюционные алгоритмы. Другой способ классификации алгоритмов зависит от того одно или множество решений улучшается в процессе поиска оптимума, например, траекторные и популяционные алгоритмы. В свою очередь, популяционные алгоритмы разделяются на следующие категории: эволюционные, моделирующие коллективное поведение агентов в популяции или рою (например, рой частиц, колония муравьев, пчелиный рой, рой светлячков, гнездовой паразитизм в поведении кукушки, роение бактерий, обезьяний поиск, алгоритм, инспирированный летучими мышами, поиск косяком рыб, сорняковый алгоритм, алгоритм растущих деревьев); алгоритмы, вдохновленные неживой природой (например, гравитационный, диффузионный, гармонический поиск); алгоритмы, инспирированные человеческим обществом (алгоритм меметики, культурный алгоритм, алгоритм эволюции разума), и др.

Многие термины, используемые при описании элементов биоэвристических алгоритмов, имеют биологическое происхождение, то есть сходны с природными аналогами. Например, в биоэвристиках используется следующая терминология: особь – потенциальное решение; потомок – копия потенциального решения (родителя); популяция – набор потенциальных решений; приспособленность, пригодность (фитнесс) – качество; селекция – отбор особей на основе их приспособленностей; рекомбинация или кроссинговер – размножение, которое использует двух родителей; генотип или геном – структура данных особи, используемая в процессе размножения; хромосома – закодированное решение; ген – определенная позиция в хромосоме; аллель – частное значение гена; фенотип – закодированное решение; поколение – один цикл оценивания, размножения и обновления популяции, либо популяция, создаваемая в каждом таком цикле.

Теоретические исследования стараются подкрепить экспериментами. Эксперименты проводятся на тестовых задачах. Алгоритмы сравниваются путем компьютерного моделирования, варьирования их параметров, а также размерностей задач. Наличие общедоступных библиотек бенчмарков, которые размещаются в сети Интернет в свободном доступе, позволяет исследователю сопоставить полученные результаты с работами других авторов. Примерами подобного рода библиотек являются библиотека генетических алгоритмов *GAlib*, библиотека для выполнения эволюционных вычислений в *Perl*, библиотека для построения популяционных алгоритмов *EAlib*, библиотека тестовых задач Института математики им. С.Л. Соболева, фреймворк для эволюционных вычислений в *Java* и др. Необходимо отметить, что между результатами, полученными в теории и при проведении вычислительных экспериментов, существует определенная разница: теоретические модели, как правило, описывают лишь алгоритмы для задач с простой структурой и с известными оптимальными решениями. В отличие от этого реализация многих биоэвристик пока не подкреплена теоретическим обоснованием. Это не является недостатком этих биоэвристик, а свидетельствует о сложности вопросов, а также подтверждает важность проведения вычислительных экспериментов.

Одним из важных вопросов при анализе биоэвристик считают вопрос о том, какая из биоэвристик является подходящей при решении некоторой задачи, и, наоборот, для каких типов задач эта биоэвристика позволит показать наилучший результат, а также для каких задач ее лучше не применять? Утверждение о существовании биоэвристического алгоритма, который позволяет всегда получать лучшие результаты для разных задач оптимизации несправедливо.

Это следует из классического результата теории биоэвристик – *NFL*-теоремы (*No Free Lunch*, бесплатных завтраков не бывает) [11]. Результат много раз пытались опровергнуть, но безуспешно.

Теорема (*NFL*). Пусть  $P(d_m^y | f, m, A)$  – условная вероятность получения частного решения  $d_m^y$  после  $m$  прогонов алгоритма  $A$  на функции пригодности  $f: X \rightarrow Y$ , где  $x \in X$  – множество входных значений,  $y \in Y$  – множество выходных значений. Тогда для любых эвристических алгоритмов  $A_1$  и  $A_2$  после  $m$  прогонов справедливо:

$$\sum_f P(d_m^y | f, m, A_1) = \sum_f P(d_m^y | f, m, A_2) \quad (1)$$

Формула (1) означает, что сумма условных вероятностей достижения в пространстве решений любой точки  $d_m^y$  одинакова для множества всевозможных функций пригодности независимо от используемого алгоритма. Тогда эффективность алгоритма в среднем для всевозможных функций пригодности  $f$  не зависит от алгоритма  $A$ . Это означает, что не существует лучшего алгоритма для решения всех проблем. Алгоритм  $A$  может выигрывать по своим характеристикам при решении некоторого класса задач. Однако этот выигрыш будет неизбежно компенсирован проигрышем (худшими характеристиками) для остальных задач. Значительная часть усилий специалистов по разработке и поиску лучших значений параметров биоэвристик (в частности, генетических алгоритмов) и созданию многочисленных клонов эволюционных операторов (кроссинговера, мутации и др.) оказалась невостребованной, т.к. из *NFL*-теоремы следует, что полученные результаты справедливы только для использованных бенчмарков, а не для произвольных задач.

В *NFL*-теореме случайный поиск – это просто другой конкурирующий алгоритм. Биоэвристика будет решать поставленную задачу лучше, чем случайный поиск, если в ней будут отражены априорные знания об этой задаче. Однако если будет взята другая задача с иной структурой знаний, то такая биоэвристика может привести к худшим результатам. Прежде всего, должна быть определена структура проблемы, а разрабатываемый алгоритм должен ей соответствовать. Редукция проблемной области без указания соответствия между рассматриваемым множеством проблем и алгоритмом недостаточна для получения преимущества данного алгоритма решения этих проблем по сравнению с другими. *NFL*-теорема просто подтверждает, что разные алгоритмы имеют различную эффективность при решении разных задач. Например, эффективность классических методов оптимизации подтверждена при решении линейных, квадратичных, строго выпуклых, унимодальных, разделяемых и других специальных классов проблем. Биоэвристики способны быстро найти неплохие решения для задач, с которыми классические методы

не справляются: целевые функции не дифференцируемы, имеют разрывы, многоэкстремальны, зашумлены, имеют сложный ландшафт, фазовое пространство переменных не является метрическим, что характерно для решения реальных практических задач.

Современные тренды в области биоинспирированных алгоритмов состоят, в частности, в самоадаптации алгоритмов к решаемой задаче непосредственно в процессе оптимизации.

### Закономерности и структура биоэвристик

Сопоставительный анализ различных моделей и алгоритмов биоэвристик позволяет выявить некоторые *закономерности* [12]:

- вначале строится формализованная модель биоэвристики, определяются правила, по которым они эволюционируют, а также способы представления (кодирования) решений;
- модели биоэвристик применимы к решению трудных задач оптимизации, у которых переменные могут быть лингвистическими и не иметь количественного выражения;
- биоэвристики моделируют процесс поиска оптимальных решений за счет реализации выполнения различных операторов, как правило, поддерживают популяцию альтернативных решений, которая эволюционирует при взаимодействии с внешней средой. Селекция позволяет производить отбор альтернативных решений с более высокими значениями функции пригодности, а эволюционные операторы создают новые альтернативные решения;
- все модели биоэвристик представляют собой итерационные эвристические процедуры, не имеют ограничений на вид целевой функции;
- популяция решений запоминается, не обязательно ограничиваясь лишь последними лучшими решениями;
- применяя механизмы адаптации, можно эффективно управлять сходимостью эволюционного поиска;
- различия в моделях биоэвристик не носят методологический характер и не затрагивают присущие им принципы.

Суть большинства биоэвристик сводится к следующему.

Выбирается популяция альтернативных решений  $X$ , и их параметры. В этой популяции выбираются наилучшие в смысле некоторого критерия пригодности  $F$ . Критерий пригодности учитывает свойств решений и может быть не задан аналитически. Имеется отображение вида  $F: X \rightarrow R$ , где  $R$  - множество действительных чисел и каждому альтернативному решению  $x \in X$  из множества  $X$  соответствует значение критерия пригодности  $F(x)$ .

Фенотип  $X$  может быть произвольным. Для того чтобы создать генотип необходимо закодировать множество  $X$  через векторное пространство  $S$ . Отображение вида  $\varphi: X \rightarrow S$  описывает связь между альтернативными решениями. Существует обратное

отображение вида  $\varphi^{-1}: S \rightarrow X$ , где каждому новому полученному альтернативному решению  $s \in S$  соответствует элемент во множестве  $X$ . Тогда, оптимизационный поиск на основе биоэвристик, заключается в получении множества решений  $X_{opt} \in X$ , при условии:

$$X_{opt} = \operatorname{argmax} F[\varphi^{-1}(s)], s \in S. \quad (2)$$

Таким образом, оптимизационный поиск эволюционирует, изменяя свой состав и параметры входящих в него альтернативных решений. Способ построения множества объектов  $s \in S$  определяется биоинспирированным алгоритмом.

Особенностью моделей эволюционных биоэвристик является применение различных механизмов кодирования в виде хромосом. Строится  $P_0$  - случайная начальная популяция, определенная на множестве  $S$ . Далее, используя следующую вычислительную схему:

$$P_{t+1} = \Theta(P_t), \quad (3)$$

переходим к новой популяции, где  $\Theta$  – композиция различных эволюционных операторов. Значение функции пригодности определяется на каждой итерации в процессе отбора решений по критерию, реализуемому в композиции операторов  $\Theta$ .

Алгоритм заканчивает свою работу, когда текущий набор значений альтернативных решений удовлетворяет хотя бы одному из заданных критериев: количество итераций, время решения, найдено оптимальное решение  $X_{opt} = (x_{1opt}, x_{2opt}, \dots, x_{nopt})$  и др.

### Представление (кодирование) решений

Важным теоретическим вопросом является представление (кодирование) решений, а также правила, по которым модель эволюционирует. В частности, вначале определяется начальная популяция решений (фенотип  $X$ ). Согласно формуле (2), задается критерий пригодности  $F$ .

В настоящее время существует несколько способов кодирования биоэвристик. Основными из них являются: бинарное кодирование; кодирование действительными числами; целочисленное кодирование, включая векторное и кодирование общей структуры данных.

Недостатками бинарного кодирования является хеммингов сдвиг, осуществляемый для пары закодированных значений с большим хемминговым расстоянием. В связи с этим бинарное кодирование не всегда подходит для решения многих оптимизационных задач. Решение многих задач оптимизации функций является более эффективным при использовании кодирования действительными числами. Из анализа экспериментальных исследований видно, что при решении многих комбинаторно-логических задач лучше всего подходит целочисленное кодирование [14].

Рассмотрим правила кодирования решений. Например, *Голдберг* в [13] определил два общих шаблона для кодирования в генетических алгоритмах:

- алфавит и кодирование длин различных генов должны быть, по-возможности, ми-

нимальными;

• различные совместимые по фенотипу решения не должны влиять друг на друга. Авторы предлагают дополнить шаблоны Голдберга следующими правилами:

• кодирование генотипа должно быть сюръективным по отношению ко всем фенотипам, т.е. каждый элемент множества  $S$  является образом хотя бы одного элемента  $X$ :

$$\forall x \in X \Rightarrow \exists s \in S : x = \varphi^{-1}(s); \quad (4)$$

• для пары фенотипов мощность их генотипов должна быть примерно одинакова:

$$\forall x_1, x_2 \in X \Rightarrow |\{s \in S : x_1 = \varphi^{-1}(s)\}| \approx |\{s \in S : x_2 = \varphi^{-1}(s)\}|; \quad (5)$$

• небольшие изменения в генотипе должны приводить к небольшим изменениям в фенотипе:

$$\forall x_1, x_2 \in X, s \in S : x_1 = \varphi^{-1}(s) \& x_2 = \varphi^{-1}(s) \Rightarrow x_1 \approx x_2. \quad (6)$$

Действительно, биоэвристики используют два типа пространств (фенотип и генотип). Операторы биоэвристик выполняются в пространстве генотипа. Селекция и оценка функции пригодности производятся в фенотипическом пространстве. Таким образом, отображение  $\varphi^{-1} : S \rightarrow X$  влияет на поведение биоэвристических алгоритмов. Проблема здесь заключается в том, что работа алгоритма может привести к появлению недопустимых или нереальных решений оптимизационной задачи. Недопустимыми являются решения, которые лежат вне области допустимых решений данной задачи. Нереальными являются решения, которые не описывают решение конкретной задачи. Недопустимость хромосом следует из существования многих оптимизационных задач с ограничениями. Что касается нереальности хромосом, то она вытекает из природы их кодирования.

В связи с этим необходимо произвести оценку выбранному способу кодирования. Одной из таких оценок является легальность кодирования [15]. Свойство легальности выбранного механизма кодирования означает, что при любой перестановке генов в хромосоме, будет получено реальное решение. Другими свойствами являются недостаточность, полнота, причинность. Свойство недостаточности вытекает из дополнительных правил (4) – (6) и означает, что отображение между генотипом и фенотипом должно иметь вид  $(1-k-1)$ - отображение «один к одному». Свойство полноты подчеркивает, что каждое альтернативное решение имеет определенное кодирование, и любая точка в пространстве решений может быть доступна для поиска. Свойство причинности связывает небольшие изменения в хромосомах, которые могут появляться посредством выполнения оператора мутации [16].

### Базовый цикл биоинспирированных алгоритмов

Все когнитивные биоинспирированные алгоритмы используют практически одни и те же процедуры инициализации. Создается популяция из  $n$  случайных особей. При наличии информации о перспективных областях пространства для начала поиска оптимума можно адаптировать случайное создание особей в сторону генерации особей из этих

областей. В действительности, можно даже создать начальную популяцию из отобранных ранее «хороших» решений. Хотя подобного рода подход следует использовать осторожно: всегда существует вероятность, что информация о «хороших» начальных областях пространства поиска является не достоверной. Лучше если инициализация использует существенную долю равномерной случайности.

Для усиления разнообразия начальной популяции также имеет смысл проверить, является ли каждая особь уникальной. Биоэвристики, как правило, создают новые наборы решений (популяции) на основании свойств предыдущих наборов решений. Хотя имеются исключения, например, алгоритм роящихся частиц [17]. Базовый цикл алгоритмов начинается с создания популяции, к которой итерационно применяются следующие процедуры. Вначале вычисляется целевая функция (приспособленность каждой особи в популяции). Затем полученная информация о приспособленностях используется для размножения – получения популяции потомков. На заключительном этапе базового цикла вычислений некоторым образом происходит объединение популяций родителей и потомков для формирования новой популяции решений, после чего поиск продолжается итерационно.

Псевдокод биоинспирированного алгоритма в базовом цикле имеет следующий вид:

```
Input: Функция пригодности  $F$   
Input:  $n$  – размер популяции альтернативных решений  
Data:  $t$  – текущий номер итерации  
Data:  $P(t=0)$  – начальная популяция альтернативных решений  
Data: параметры биоэвристики, включая функцию пригодности  
Output:  $X^*$  – найденное решение (субоптимальное)  
begin  
   $t := 0$   
   $Pop := initPop(n)$  /*функция init выполняет случайную генерацию  
  начальной популяции альтернативных решений*/  
  while (критерий останова) do  
     $v := F(Pop, F)$   
     $P(t) := selection(Pop, v)$   
     $t := t + 1$   
     $Pop := P(t)$  /*репродукция потомков из родителей с использованием  
    комбинации эволюционных операторов  $\Theta$ :  $P(t+1) = \Theta(P(t))$ */  
  return /*восстановление фенотипа  $Pop$ */  
end
```



### Теорема Холланда

Теорема Холланда дает обоснование эффективности генетических алгоритмов [5]. Она определяет оценку снизу для количества примеров схем после очередного цикла эволюции:

$$N(h, t + 1) \geq N(h, t) \frac{f(h, t)}{f(t)} \left[ 1 - \frac{\delta(h)}{L-1} p_c - o(h) p_m \right], \quad (7)$$

где  $N(h, t)$  – количество примеров схемы  $h$  на шаге  $t$ ;  $N(h, t + 1)$  – количество примеров схемы  $h$  на следующем шаге;  $f(h, t)$  – функция приспособленности схемы на шаге  $t$ ;  $f(t)$  – среднее значение функции приспособленности во всей популяции на том же шаге  $t$ ;  $L$  – число позиций в хромосоме;  $\delta(h)$  – определяющая длина схемы;  $o(h)$  – порядок схемы;  $p_c$  – вероятность уничтожения схемы под действием оператора кроссинговера;  $p_m$  – вероятность уничтожения схемы под действием оператора мутации. Согласно (7) шаблоны, обладающие малой определяющей длиной, невысоким порядком и функцией приспособленности выше средней в популяции, увеличивают количество строк, аналогичных шаблону, в последующих поколениях по экспоненциальному закону.

Теорема Холланда весьма упрощенно описывает поведение генетических алгоритмов, а её доказательство, построенное на элементах теории вероятности, не учитывает достаточно большой разброс значений функций пригодности шаблонов в некоторых практических задачах. К тому же, поскольку популяция в генетических алгоритмах имеет конечную величину, то возникает неизбежная ошибка при определении значения функции приспособленности шаблона. Теорема не имеет практического применения, за исключением выполнения простых приложений с пропорциональным селективным отбором. Отметим также, что при решении задач большой размерности действительное значение целевой функции из-за ошибки может сильно отличаться от среднестатистического значения.

Одним из главных теоретических вопросов в области решения оптимизационных задач является условия сходимости к глобальному оптимуму. Сходимость ГА к глобальному оптимуму в теореме Холланда не доказана. Это в настоящее время является малоисследованной проблемой. Доказательство, что простейший ГА не сходится к глобальному оптимуму за конечное время основано на фундаментальной теории цепей Маркова. Следует признать, что методология управления сходимостью даже для простейшего ГА до сих пор не выработана, хотя подтверждением эффективности алгоритма является доказательство его сходимости и оценка вычислительной сложности. Но, как правило, это возможно только в случае упрощенной постановки задачи.

Вероятно, для практики более важным является не вопрос сходимости, а другой вопрос: позволяет ли биоэвристика найти субоптимальное решение за заданное время? Ответ на этот вопрос неоднозначен, теория пока не дает объяснений многим успешным практическим результатам.

Каковы критерии того, насколько подходящим является использование биоэвристик

для решения тех или иных задач? В [13] исследовались ложные проблемы, связанные с нахождением глобального оптимума. Очевидно, что на работу биоэвристики оказывает существенное влияние выполнение эволюционных операторов. Это влияние отражает коэффициент корреляции  $r_{eo}$ . Этот коэффициент устанавливает взаимосвязь между значениями функций пригодности родительских хромосом и хромосом-потомков.

В [12] приводятся бенчмарки с известным значением глобального максимума и определенными значениями  $r_{eo}$ . Далее эти задачи классифицировались на следующие группы:

- легко разрешимые ( $r_{eo} \leq -0,15$ );
- трудно разрешимые ( $-0,15 < r_{eo} < 0,15$ );
- обманчивые ( $r_{eo} \geq 0,15$ )

Если значение глобального оптимума заранее неизвестно, то можно использовать вместо него лучшее из известных решений.

### **Дрейф-анализ**

Перспективным направлением в анализе времени работы когнитивных биоинспирированных алгоритмов на сегодняшний день является анализ дрейфа (*drift*) [18].

Например, пусть задана следующая задача комбинаторной оптимизации. В конечном пространстве состояний  $S$  имеется некоторая функция  $f(x)$ ,  $x \in S$ . Найти

$$\max \{f(x); x \in S\} \quad (8)$$

Пусть  $x^*$  - состояние с максимальным значением функции  $f_{max} = f(x^*)$ . Абстрактный биоинспирированный алгоритм для решения поставленной оптимизационной задачи включает следующие шаги.

1. *Инициализация* популяции (случайным образом или эвристически)  $\xi_0 = (x_1, \dots, x_{2n})$  из  $2n$  особей ( $n$  – целое число). Присвоить  $k = 0$ . Для каждой популяции  $\xi_k$  определить  $\xi_k = \max \{f(x_i) : x_i \in \xi_k\}$ .
2. *Генерация* популяции  $\xi_{k+1/2}$  с помощью эволюционных операторов.
3. *Селекция и репродукция*  $2n$  особей из популяций  $\xi_{k+1/2}$  и  $\xi_k$  и получение новой популяции  $\xi_{k+S}$ .
4. Если  $f(\xi_{k+S}) = f_{max}$ , то *stop* иначе  $\xi_{k+1} = \xi_{k+S}$ ,  $k = k + 1$  и переход к шагу 2.

Приведенная выше структура алгоритма ближе к эволюционной стратегии и эволюционному программированию [12], чем к генетическим алгоритмам, в том смысле, что селекция применяется после выполнения эволюционных операторов. Однако для анализа дрейфа эти детали не имеют значения.

Пусть  $x^*$  - точка оптимума. Обозначим  $d(x, x^*)$  расстояние между точкой  $x$  и  $x^*$ . Если имеется множество оптимумов  $S^*$ , то  $d(x, S^*) = \min\{d(x, x^*) : x^* \in S^*\}$  является расстоянием между точкой  $x$  и множеством  $S^*$ . Это расстояние будем просто обозначать через  $d(x)$ . Тогда  $d(x^*) = 0$ ,  $d(x) > 0$  для любого  $x \notin S^*$ .

Учитывая, что популяции  $X = \{x_1, \dots, x_{2n}\}$ , положим

$$d(X) = \min\{d(x) : x \in X\}. \quad (9)$$

Формула (9) служит для измерения расстояния между популяцией и оптимальным решением.

Последовательность  $\{d(\xi_k); k = 0, 1, 2, \dots\}$ , порожденная биоэвристикой, является случайной последовательностью, которая моделируется однородной цепью Маркова.

Тогда дрейф случайной последовательности в момент времени  $k$  определяется как

$$\Delta(d(\xi_k)) = d(\xi_{k+1}) - d(\xi_k). \quad (10)$$

Время останова алгоритма оценивается как  $\tau = \min \{k : d(\xi_k) = 0\}$ . Задача состоит в изучении взаимосвязи между временем  $\tau$  и размерностью задачи  $n$ . При каких значениях дрейфа  $\Delta(d(\xi_k))$  можно оценить математическое ожидание  $E[\tau]$ ? Найдет ли в среднем алгоритм оптимальное решение за полиномиальное время или ему потребуется экспоненциальное время?

Идея анализа дрейфа довольно проста. Ключевым вопросом здесь является оценка соотношения  $d$  и  $\Delta$ .

Биоинспирированный алгоритм может решить оптимизационную задачу за полиномиальное среднее время при следующих условиях дрейфа:

- если существует полином  $h_0(n) > 0$  ( $n$  – размерность задачи) такой, что  $d(X) \leq h_0(n)$  для любой данной популяции  $X$ , т.е. расстояние от любой популяции до оптимального решения является полиномиальной функцией от размерности задачи;
- в любой момент  $k \geq 0$ , если  $d(\xi_k) > 0$ , то существует полином  $h_1(n) > 0$  такой, что

$$E[d(\xi_k) - d(\xi_{k+1}) | d(\xi_k) > 0] \geq 1/h_1(n), \quad (11)$$

т.е. дрейф случайной последовательности  $\{d(\xi_k); k = 0, 1, 2, \dots\}$  по отношению к оптимальному решению всегда положителен и ограничен обратным полиномом.

В [18] доказано следующее:

- если данные условия соблюдаются для случайной последовательности  $\{d(\xi_k); k = 0, 1, 2, \dots\}$ , то уже от начальной популяции  $X$  с  $d(X) > 0$  выполняется неравенство

$$E[\tau | d(\xi_0) > 0] \leq h(n), \quad (12)$$

где  $h(n)$  – полиномом размерности задачи  $n$ ;

- если имеется множество задач и биоинспирированный алгоритм для их решения, то для любой начальной популяции  $X$  с  $d(X) > 0$ , справедливо

$$E[\tau \mid \xi_0 = X] \leq h(n); \quad (13)$$

- если целевая функция является линейной с положительными коэффициентами ( $c_1 > c_2 > \dots > c_n > 0$ ), то для получения оптимального решения биоинспирированному алгоритму с вероятностью мутации  $p_m = 1/n$  потребуется в среднем  $O(n \ln(n))$  шагов;
- если целевая функция  $f: S \rightarrow \mathbf{R}$  является псевдомодульной  $\forall x, y \in S$  (например,  $f(x) = \sum_{i=1}^n \prod_{j=1}^i s_j$ ), то при вероятности мутации  $p_m = 1/n$  ожидаемое время останова биоинспирированного алгоритма удовлетворяет неравенству

$$E[\tau] \leq n^2(e - 1). \quad (14)$$

Если биоэвристика не в состоянии найти оптимальное решение за полиномиальное время, то тогда есть смысл искать приближенное решение, определив момент останова алгоритма  $\tau$  как  $\tau = \min\{k : d(\xi_k) \leq d_b\}$ , где  $d_b \geq 0$ . Это выполняется при следующих условиях дрейфа [18]:

- для любой популяции  $X$  с  $d_b < d(X) < d_a$ , где  $d_b \geq 0, d_a > 0$  справедливо

$$E[\exp(-(d(\xi_{k+1}) - d_a) \mid \xi_k = X, d(\xi_k) \geq d_a)] \leq D \text{ (константа } D \geq 1). \quad (15)$$

Условие (15) означает, что популяция на интервале  $[d_a, +\infty)$  не будет в среднем дрейфовать к оптимальному решению, потому что  $d(\xi_{k+1}) \geq d_a - \ln(D)$ .

В [18] доказано, что при выполнении условий (15), если  $d(\xi_0) > d_a$ ,  $D \geq 1, r < 1$ , существуют некоторые  $\delta_1 > 0$  и  $\delta_2 > 0$  такие, что справедливо неравенство

$$E[\tau \mid d(\xi_0) > d_a] \geq \delta_1 \exp(\delta_2 (d_a - d_b)). \quad (16)$$

Иными словами, биоэвристика при некоторых условиях может потребовать в среднем экспоненциального времени для поиска оптимального решения.

Следствием результатов дрейф-анализа является то обстоятельство, что оценка значения дрейфа превращается в оценку времени работы алгоритма, а локальное свойство (дрейф за один шаг) преобразуется в глобальное свойство (время работы алгоритма до нахождения оптимума)! Это новый результат в оценке временной сложности биоэвристик, полученный посредством анализа дрейфа. Оценить дрейф проще. С помощью анализа дрейфа определены условия, выполнение которых гарантирует решение некоторых задач в среднем за полиномиальное время, а также условия, при которых алгоритм требует для решения задачи в среднем экспоненциальное время вычисления.

Отметим, что эти результаты были получены в предположении, что число поколений (что эквивалентно числу вычислений функции приспособленности) является наиболее важным фактором при оценке времени вычислений алгоритма. Это, пожалуй, справедливо для большинства приложений биоинспирированных алгоритмов, поскольку в них оценка приспособленности является наиболее трудоёмкой частью алгоритма, в отличие от операций кроссинговера и мутации, трудоёмкость которых оценивалась как  $O(n)$ , и

операции селекции, трудоемкость которой оценивалась как  $O(n \ln(n))$ .

В этом плане заслуживает внимания работа [19], в которой использовалась теорема «аддитивного дрейфа». Было предложено несколько теорем такого вида, включая верхние и нижние оценки в случае мультипликативного и переменного дрейфа [20]. Значительный прогресс наблюдается в области порождения функций расстояния, которые используются для адаптации процесса оптимизации к входным условиям дрейф-теорем. Теоретический аппарат дрейф-теорем, существующий в настоящее время, позволяет анализировать биоэвристики на тестовых задачах и задачах комбинаторной оптимизации.

Несомненно, на временную сложность биоэвристик оказывают влияние их параметры (вероятность кроссинговера, мутации и др.). Было бы интересно исследовать степень этого влияния для различных задач, чтобы получить некоторое представление об эффективности различных операторов и параметров настройки алгоритма.

Требуют дальнейшего изучения более строгие условия дрейфа, чтобы получить более сильные верхние оценки для среднего времени вычислений.

Перспективным представляется проведение дрейф-анализа биоэвристик на известных трансвычислительных задачах, чтобы понимать насколько эффективным является их применение для решения этих задач.

### Заключение

Одна из метафор биоэвристик состоит в том, что задача оптимизации решается так, как если бы ее решала природа: с помощью направленной эволюции потенциальных решений задачи путем внесения случайных небольших изменений и рекомбинации хороших решений с целью получить лучшие.

В настоящее время биоэвристики активно используются в различных областях: в дизайне космических кораблей и зубных щеток, при разработке антенн для микро спутников, усилителей, фильтров, контроллеров, осцилляторов и других электронных схем [21], двигателей самолетов и новых антибиотиков, для конструирования роботов и спам-фильтров, когнитивных музыкальных ди-джеев и др. Среди фирм, использующих когнитивные биоинспирированные технологии, присутствуют такие известные организации и компании, как *NASA, General Motors, Boeing, Honda, Yamaha, Gillette, General Electric, Genetic Programming, Cloudmark, Yandex, Hewlett-Packard, Proctor&Gamble, Coca Cola* и др.

Одна из серьезнейших проблем биоэвристик связана с оценкой времени их работы. Причиной этого на фундаментальном уровне можно назвать их обобщенность вследствие отсутствия специализации под какие-либо задачи. С помощью аппарата дрейф-анализа оценка значения дрейфа превращается в оценку времени работы алгоритма. Теоретический аппарат дрейф-анализа, существующий в настоящее время, позволяет анализировать биоинспирированные алгоритмы на тестовых задачах и задачах комбинаторной оптимизации.

**Библиография :**

1. Rastrigin L.A. The convergence of the random search method in the extremal control of a many parameter system // Automation and remote control. 1963. no. 24(10). pp. 1337–1342.
2. Nelder J.A., Mead R. A simplex method for function minimization // Computer journal. 1965. no. 7. pp. 308–313.
3. Fogel L., Owens A.J., Walsh M.J. Artificial intelligence through simulated evolution. Wiley, 1966. 452 p.
4. Kernighan B.W., Lin S. An efficient heuristic procedure for partitioning graphs // Bell system technical journal. 1970. no. 49. pp. 291–307.
5. Holland J.H. Adaptation in natural and artificial systems. Uni of Michigan press, 1975. P. 45.
6. Smith S.F. A Learning system based on genetic adaptive algorithms. PhD thesis, Uni of Pittsburgh, 1980. P. 21.
7. Kirkpatrick S., Gelatt Jr. C.D., Vecchi M.P. Optimization by simulated annealing // Science. 1983. no. 220. pp. 671–680.
8. Glover F. Future paths for integer programming and links to artificial intelligence // Computers and operations research. 1986. no. 13. pp.533–549.
9. Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms // Caltech concurrent computation program, 1989. Report 826.
10. Dorigo M. Optimization, learning and natural algorithms. PhD thesis, Politecnico di Milano, Italy, 1992. 152 p.
11. Wolpert D.H., Macready W.G. The no free lunch theorems for optimization // IEEE Trans. evol. comp. 1997. no. 1. pp.67–82.
12. Курейчик В.В., Курейчик В.М., Родзин С.И. Теория эволюционных вычислений. М.: Физматлит, 2012. 260 с.
13. Goldberg D.E. Genetic algorithms in search, optimization, and machine learning. USA: Addison–Wesley publishing company, inc., 1989. 432 p.
14. Гладков Л.А., Курейчик В.В., Курейчик В.М., Сорокалетов П.В. Биоинспирированные методы в оптимизации. М.: Физматлит, 2009. 380 с.
15. Курейчик В.М., Родзин С.И. Эволюционные алгоритмы: генетическое программирование (обзор) // Известия РАН. Теория и системы управления. 2002. № 1. С. 127–137.
16. Rodzin S. Smart dispatching and metaheuristic swarm flow algorithm // Jour. of computer and systems sciences international. 2014. Vol. 53. no. 1. pp. 109–115.
17. Eberhart R., Shi Yu., Kennedy J. Swarm intelligence. Morgan Kaufmann, 2010. 512 p.
18. He J., Yao X. Drift analysis and average time complexity of evolutionary algorithms // Artificial intelligence. 2001. Vol. 127. no. 1. pp. 57–85.
19. Jansen T. Fixed budget computations: why, how and what? // Proc. of Dagstuhl seminar on theory of evolutionary algorithms, 2013. pp. 1325–1332.
20. Doerr B., Goldberg L. Adaptive drift analysis // Algorithmica. 2013. Vol. 65, no. 1. pp. 224–250.
21. Rodzin S., Rodzina L. Theory of bionic optimization and its application to evolutionary synthesis of digital devices // Proc. of the 14th IEEE east-west design & test symposium (EWDTS'14), 2014. pp. 147–152.

**References:**

1. Rastrigin L.A. The convergence of the random search method in the extremal control of a many parameter system // Automation and remote control. 1963. no. 24(10). pp. 1337–1342.
2. Nelder J.A., Mead R. A simplex method for function minimization // Computer journal. 1965. no. 7. pp. 308–313.
3. Fogel L., Owens A.J., Walsh M.J. Artificial intelligence through simulated evolution. Wiley, 1966. 452 r.
4. Kernighan B.W., Lin S. An efficient heuristic procedure for partitioning graphs // Bell system technical journal. 1970. no. 49. pp. 291–307.
5. Holland J.H. Adaptation in natural and artificial systems. Uni of Michigan press, 1975. P. 45.
6. Smith S.F. A Learning system based on genetic adaptive algorithms. PhD thesis, Uni of Pittsburgh, 1980. P. 21.
7. Kirkpatrick S., Gelatt Jr. C.D., Vecchi M.P. Optimization by simulated annealing // Science. 1983. no. 220. pp. 671–680.
8. Glover F. Future paths for integer programming and links to artificial intelligence // Computers and operations research. 1986. no. 13. pp.533–549.
9. Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms // Caltech concurrent computation program, 1989. Report 826.
10. Dorigo M. Optimization, learning and natural algorithms. PhD thesis, Politecnico di Milano, Italy, 1992. 152 r.
11. Wolpert D.H., Macready W.G. The no free lunch theorems for optimization // IEEE Trans. evol. comp. 1997. no. 1. pp.67–82.
12. Kureichik V.V., Kureichik V.M., Rodzin S.I. Teoriya evolyutsionnykh vychislenii. M.: Fizmatlit, 2012. 260 s.
13. Goldberg D.E. Genetic algorithms in search, optimization, and machine learning. USA: Addison–Wesley publishing company, inc., 1989. 432 r.
14. Gladkov L.A., Kureichik V.V., Kureichik V.M., Sorokaletov P.V. Bioinspirirovannye metody v optimizatsii. M.: Fizmatlit, 2009. 380 s.
15. Kureichik V.M., Rodzin S.I. Evolyutsionnye algoritmy: geneticheskoe programmirovaniye (obzor) // Izvestiya RAN. Teoriya i sistemy upravleniya. 2002. № 1. S. 127–137.
16. Rodzin S. Smart dispatching and metaheuristic swarm flow algorithm // Jour. of computer and systems sciences international. 2014. Vol. 53. no. 1. pp. 109–115.
17. Eberhart R., Shi Yu., Kennedy J. Swarm intelligence. Morgan Kaufmann, 2010. 512 r.
18. He J., Yao X. Drift analysis and average time complexity of evolutionary algorithms // Artificial intelligence. 2001. Vol. 127. no. 1. pp. 57–85.
19. Jansen T. Fixed budget computations: why, how and what? // Proc. of Dagstuhl seminar on theory of evolutionary algorithms, 2013. pp. 1325–1332.
20. Doerr B., Goldberg L. Adaptive drift analysis // Algorithmica. 2013. Vol. 65, no. 1. pp. 224–250.
21. Rodzin S., Rodzina L. Theory of bionic optimization and its application to evolutionary synthesis of digital devices // Proc. of the 14th IEEE east-west design & test symposium (EWDTS'14), 2014. pp. 147–152