

Университет ИТМО

Факультет информационных технологий и программирования
Кафедра компьютерных технологий

Буланова Нина Сергеевна

**Исследование эффективности применения
вспомогательных оптимизируемых величин при
использовании методов оптимизации на основе
искусственных иммунных систем**

Научный руководитель: н.с. МНЛ КТ, к.ф.-м.н., А. А. Фильченков

Санкт-Петербург
2015

Оглавление

Введение	6
Глава 1. Обзор предметной области	7
1.1. Эволюционные вычисления.....	7
1.1.1. Эволюционные алгоритмы	7
1.1.2. Искусственные иммунные системы	8
1.1.3. Операторы мутации искусственных иммунных систем.....	9
1.2. Теория оптимизации.....	10
1.3. Обучение с подкреплением.....	11
1.4. Выводы по главе 1	13
Глава 2. Постановка задачи и описание методов ее решения.....	14
2.1. Постановка задачи	14
2.2. Выбор вспомогательных оптимизируемых величин в AIS с помощью обучения с подкреплением	15
2.3. Выводы по главе 2	17
Глава 3. Теоретические оценки сложности исследуемых модельных задач для AIS	18
3.1. Модельная задача ONEMAX	18
3.2. Модельная задача LEADINGONES	19
3.3. Модельная задача XDIVK	19

3.4. Модельная задача H-IFF.....	21
3.5. Выводы по главе 3	23
Глава 4. Сравнение эффективности применения AIS совместно и без multi-objectivization	24
4.1. Игнорирование неэффективного критерия	24
4.1.1. Модельная задача ONEMAX с мешающим критерием ZEROMAX	24
4.1.2. Модельная задача MH-IFF.....	26
4.2. Задачи с помогающим критерием	29
4.2.1. Модельная задача XDIVK.....	29
4.2.2. Модельная задача MINMAX	33
4.3. Неэффективный выбор вспомогательного критерия	37
4.4. Результаты статистического теста	40
4.5. Выводы по главе 4	42
Заключение	43
Источники.....	44

Введение

Искусственные иммунные системы (Artificial Immune Systems) – это класс алгоритмов похожих на эволюционные алгоритмы, но имеющий свои особенности. Эти алгоритмы можно использовать для решения задач оптимизации. Показано, что при решении некоторых задач искусственные иммунные системы позволяют получить более качественные решения в начале процесса оптимизации, хотя в долгосрочной перспективе проигрывают эволюционным алгоритмам[1].

Известно, что эффективность работы эволюционных алгоритмов при решении задач однокритериальной оптимизации можно повысить путем ввода вспомогательных оптимизируемых величин – вспомогательных критериев[2]. Этот подход сведения однокритериальных задач оптимизации к многокритериальным может ускорить процесс поиска оптимума целевого критерия и позволяет избежать остановки процесса оптимизации в локальных оптимумах.

В работе предлагается выяснить, может ли данный подход, показавший свою эффективность для эволюционных алгоритмов, быть применен к искусственным иммунным системам (AIS). На нескольких модельных задачах проводится сравнение результатов применения AIS с и без многокритериальной оптимизации.

Глава 1. Обзор предметной области

В этой главе приводится история развития эволюционного вычисления. Дается описание алгоритмов искусственных иммунных систем и эволюционных алгоритмов. Описываются основные подходы многокритериальной оптимизации.

1.1. Эволюционные вычисления

В современно науке существует большое направление в развитии систем искусственного интеллекта, называемое эволюционными вычислениями. В нем присутствуют несколько подходов, основанных на природных принципах эволюции, которые развивались в разных странах для решения задач, в том числе и оптимизационных. В конечном итоге, в результате взаимного проникновения принципов одних методов в другие, различий между ними стало совсем не много. Это создало общее направление эволюционного подхода, в котором поиск оптимального решения происходит путем формирования популяции, воздействия на нее генетических операторов и селекции.

1.1.1. Эволюционные алгоритмы

Решение задач оптимизации с помощью эволюционных алгоритмов сводится к нахождению оптимума функции приспособленности, которая характеризует каждую особь[3]. Во время работы эволюционного алгоритма к особям применяются операторы мутации и скрещивания. Потом производится процесс селекции наиболее приспособленных особей, что определяется значением их функции приспособленности. Далее

эволюционный процесс повторяется до нахождения оптимума задачи, либо по достижении максимума количества итераций алгоритма. В работе будут рассматриваться оба варианта исследований, так как иногда нахождение решения задачи невозможно или требует слишком большого количества вычислений.

1.1.2. Искусственные иммунные системы

Направление эволюционных вычислений, основывающееся на использовании моделей и характеристик человеческой иммунной системы[4]. То есть эволюционный процесс в искусственных иммунных системах проходит по механизмам естественных иммунных систем.

Изучение естественных иммунных систем очень сложная задача, и для описания ее функционирования существуют три теории:

1. негативный отбор
2. иммунная сеть
3. клональная селекция.

Первая теория использует так называемые негативные алгоритмы отбора, которые берут за основу модели процессов позитивной и негативной селекции. В большинстве случаев эти алгоритмы используют при решении задач классификации и распознавания[5].

Вторая теория использует иммунные сетевые алгоритмы при решении задач кластеризации и визуализации данных. Также существуют варианты алгоритмов этого класса, которые достаточно успешно решают задачи оптимизации[6].

Последняя теория представляет наибольший интерес в этой работе. Алгоритмы клонального отбора основаны на теории клоновой селекции

приобретенного иммунитета, и используются в основном для решения задач оптимизации и распознавания образов[7].

1.1.3. Операторы мутации искусственных иммунных систем

Класс операторов мутации, предложенный[8] носит название *Somatic Contiguous Hypermutation* (СНМ). В них случайным образом выбирается позиция первого мутирующего бита и количество бит, идущих за выбранным, которые так же подвергнутся мутации с некоторой вероятностью. При таком подходе вероятность мутации каждого конкретного бита очень зависит от его положения (СНМ₁, СНМ₂)[9, 10]. Однако существует вариант этой мутации (СНМ₃), так называемый *Wrapping Around*, который позволяет последовательности мутирующих бит “оборачиваться” с хвоста особи в ее начало[11]. Таким образом, вероятность мутации становится одинаковой для всех бит особи. Примером алгоритма, использующего такую мутацию, является V-cell Algorithm. Для СНМ более вероятны «большие» мутации, обеспечивающие оптимизационные скачки во время поиска решения. Это объясняет, почему в начале процесса оптимизации AIS алгоритмы более эффективны, чем эволюционные. Но также это замедляет поиск окончательного решения, так как мутации одного бита маловероятны.

Рассмотрим два алгоритма, которые впоследствии будем использовать в исследовании – эволюционный алгоритм Random Local Search (RLS) и алгоритм искусственных иммунных систем V-cell Algorithm (VCA). Будем рассматривать тот вариант алгоритма, который использует в своей работе только оператор мутации СНМ₃, так как этого вполне достаточно для того, чтобы наглядно показать отличия этих двух групп алгоритмов.

Листинг 1 Различия алгоритмов искусственных иммунных систем и эволюционных алгоритмов

<i>RLS</i>	<i>BCA</i>
<p>Initialisation $t := 0$; Select $x_t \in \{0,1\}^n$ u.a.r.</p> <p>Main Loop While $t < b$</p> <p style="padding-left: 40px;">Mutation Set $y := x_t$ Select $i \in \{0, \dots, n-1\}$ u.a.r. Set $y[i] := 1 - y[i]$</p> <p>Selection If $F(y) \geq F(x_t)$ Then set $x_{t+1} := y$ Else set $x_{t+1} := x_t$ $t := t+1$</p>	<p>Initialisation $t := 0$; Select $x_t \in \{0,1\}^n$ u.a.r.</p> <p>Main Loop While $t < b$</p> <p style="padding-left: 40px;">Mutation Set $y := x_t$ Select $p \in \{0, \dots, n-1\}$ u.a.r. Select $l \in \{0, \dots, n\}$ u.a.r. For each $i \in \{0, \dots, l\}$ Set $y[(p+i) \bmod n]$ $:= 1 - y[(p+i) \bmod n]$</p> <p>Selection If $F(y) \geq F(x_t)$ Then set $x_{t+1} := y$ Else set $x_{t+1} := x_t$ $t := t+1$</p>

Из приведенных выше псевдокодов можно заключить, что общего у алгоритмов искусственных иммунных систем и эволюционных гораздо больше, чем отличий. В обоих алгоритмах происходит инициализация особей поколения (в данном случае алгоритмы работают с одной особью в поколении), и последующая мутация и селекция. Заметим, что рознятся эти алгоритмы только операторами мутации.

1.2. Теория оптимизации

Задача однопараметрической (однокритериальной, скалярной) оптимизации заключается в нахождении единственного наилучшего решения, максимизирующего возможное значение целевого критерия при заданных ограничениях. При ее сведении к многокритериальной вводятся дополнительные критерии, помогающие или мешающие процессу поиска

оптимума. Соответствующий подход называется multi-objectivization[12–14]. Основные методы, относящиеся к нему:

1. Multiobjectivization [13]
2. Helper objectives [12]
3. EA+RL [15, 16]

Суть многокритериальной оптимизации заключается в отказе от поиска единственного лучшего решения. Вместо этого производится поиск множества решений, которые лучше других, но сравнить их затруднительно. В этом подходе происходит одновременная оптимизация всех критериев с помощью многокритериального алгоритма. Во многих алгоритмах многокритериальной оптимизации используется концепция доминирования по Парето[17]. То есть в качестве результата предоставляется множество Парето или его приближение.

В случае динамического выбора вспомогательных критериев не все дополнительные критерии оптимизируются одновременно. Происходит случайный выбор вспомогательных критериев из всех доступных и последующая оптимизация выбранных критериев с целевым[12].

1.3. Обучение с подкреплением

В предыдущих подходах подразумевалось, что каждый из дополнительных критериев при использовании оптимизирует и целевой критерий. Разработка критериев, удовлетворяющих этому требованию, достаточно сложная задача, потому интересен подход, который автоматически выбирает наиболее эффективный критерий и игнорирует неэффективные. Таким подходом является обучение с подкреплением (Reinforcement Learning)[15]. В нем задача поиска оптимума

вспомогательных критериев не ставится, они используются только для повышения эффективности используемого алгоритма в поиске максимума целевого критерия.

Являясь одним из способов машинного обучения, обучение с подкреплением осуществляется путем взаимодействия агента и среды. Общая схема этого взаимодействия представлена на Рис. 1.

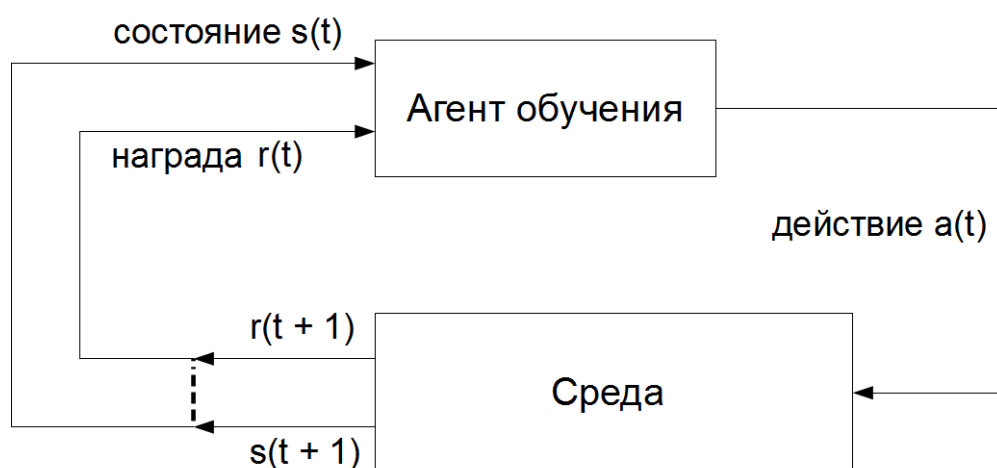


Рис. 1. Схема взаимодействия среды и агента в обучении с подкреплением, t – номер итерации.

Здесь агент действует на среду и получает от нее награду и представление о текущем состоянии среды. После этого агент обучается на полученной информации и происходит повторение цикла взаимодействия. Целью агента является максимизация суммарной награды[16]. В качестве действия будем рассматривать процесс выбора оптимизационного критерия из заранее определенного множества критериев. Наградой является целое или вещественное число, характеризующее успешность применения действия к среде, чем оно больше, тем более эффективно действие.

Существуют различные стратегии исследования среды. В данной работе нас интересуют жадная и ϵ -жадная стратегии[16]. Первая из них

считается самой простой и не использует накопленный опыт, а всегда выбирает действие, дающее наибольшую награду в текущем состоянии. Вторая стратегия заключается в том, что с вероятностью ϵ агент может выбрать случайное из набора действий, вместо того чтобы придерживаться жадной стратегии. Последний вариант помогает процессу оптимизации не застревать в локальных оптимумах.

1.4. Выводы по главе 1

В главе были описаны методы эволюционных вычислений, такие как алгоритмы, основанные на искусственных иммунных системах и эволюционные алгоритмы. Показаны сходства и различия этих двух классов алгоритмов. Так же был сделан обзор методов сведения однокритериальных задач оптимизации к многокритериальным. Описаны особенности многокритериальной оптимизации, динамического выбора критериев и обучения с подкреплением.

Глава 2. Постановка задачи и описание методов ее решения

В данной главе приводится описание исследуемой задачи и описание методов ее решения.

2.1. Постановка задачи

Искусственные иммунные системы (AIS) можно использовать для решения задач оптимизации[1]. При решении некоторых задач искусственные иммунные системы позволяют получить более качественные решения в начале процесса оптимизации, хотя в долгосрочной перспективе проигрывают эволюционным алгоритмам.

Данная работа заключается в исследовании эффективности применения вспомогательных оптимизируемых величин при использовании методов оптимизации на основе искусственных иммунных систем. Известно, что эффективность эволюционных алгоритмов может быть повышена путем ввода дополнительных критериев оптимизации, в частности использованием обучения с подкреплением[15]. Предлагается рассмотреть алгоритмы оптимизации на основе иммунных систем и сравнить результаты их работы с результатами эволюционных алгоритмов. Затем проверить эффективность подхода multi-objectivization, применительно к AIS. Полученные результаты сравнить с результатами решения модельных задач с помощью искусственных иммунных систем без multi-objectivization, а так же с помощью эволюционных алгоритмов.

2.2. Выбор вспомогательных оптимизируемых величин в AIS с помощью обучения с подкреплением

Исследование применения обучения с подкреплением к искусственным иммунным системам будем проводить с помощью метода, который назовем AIS+RL. Схема приведена ниже на Рис. 2.

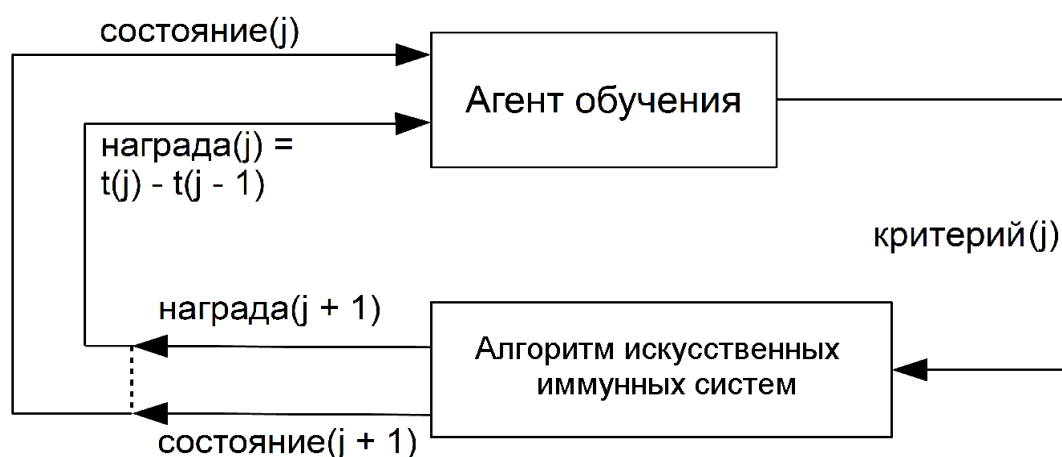


Рис. 2. Схема метода AIS+RL, j – номер итерации, t – целевой критерий.

В течение процесса поиска оптимума целевого критерия выполняется один и тот же цикл действий, который прерывается при срабатывании критерия останова или достижении максимума числа поколений. В качестве среды здесь выступает алгоритм на основе искусственных иммунных систем. Выбор действия агентом заключается в выборе критерия из набора, включающего в себя целевой. Выбранный критерий используется средой для формирования следующего поколения. Затем среда возвращает агенту награду, представляющую собой прирост целевого критерия по сравнению с предыдущим поколением. Далее агент пересчитывает оценку эффективности применения каждого их критериев.

В качестве алгоритма обучения с подкреплением, который использовался для сведения однокритериальных задач к многокритериальным, был взят алгоритм Q-обучения. Далее описано использование искусственных иммунных систем и алгоритма Q-learning.

Алгоритм Q-обучения (Q-learning)[16, 18] относится к алгоритмам итерации по значениям. В ходе работы он не строит модель среды, а аппроксимирует функцию $Q(s, a)$, ожидаемой эффективности применения действия a в состоянии s . Далее приведен псевдокод используемого алгоритма.

Листинг 2 Оптимизация AIS с использованием Q-обучения

Вход: t – целевой критерий оптимизации; F – множество всех критериев оптимизации (содержит t); α – скорость обучения; γ – дисконтный фактор.

```

1: Инициализировать  $Q[f]$  нулями для всех  $f$  из  $F$ 
2: Инициализировать текущую особь  $x$ 
3: while (! критерий останова) do
4:     Запомнить текущую особь  $prev = x$ 
5:     Выбрать критерий:  $f = argmax_f Q[f]$ 
6:     Мутировать особь:  $y = AISmutation(x)$ 
7:     Выбрать текущую особь: если  $f(y) \geq f(x)$ , то  $x = y$ 
8:     Вычислить награду:  $r = t(x) - t(prev)$ 
9:     Обновить  $Q$ :  $Q[f] = Q[f] + \alpha * (r + \gamma * max_{f'} Q[f'] - Q[f])$ 
10: end while

```

В момент выбора критерия возможно использование ϵ -жадной стратегии, но проведенные исследования показывают, что для алгоритмов искусственных иммунных систем оптимальным является использование обычной жадной стратегии.

2.3. Выводы по главе 2

Произведена постановка задачи определения эффективности сведения однокритериальных задач оптимизации, решаемых с помощью искусственных иммунных систем, к многокритериальным. Описаны основные методы, применяемые в исследовании. Предложен метод использования искусственных иммунных систем в обучении с подкреплением, названный AIS+RL.

Глава 3. Теоретические оценки сложности исследуемых модельных задач для AIS

В данной главе рассматривается несколько модельных задач, использованных в этой работе. Дается их описание, и приводятся оценки времени работы выбранного AIS алгоритма на рассматриваемых задачах. Данное исследование будет полезно для дальнейшего эффективного выбора вспомогательных критериев для сведения однокритериальных задач к многокритериальным.

Рассматриваются следующие модельные задачи:

1. ONEMAX
2. LEADINGONES
3. XDIVK
4. H-IFF

3.1. Модельная задача ONEMAX

Первой опишем модельную задачу ONEMAX[19]. Пространством поиска здесь являются битовые строки фиксированной длины. Особь характеризуется количеством единиц в ней, соответственно оптимумом задачи является строка из всех единиц. В источнике приводится теоретическая оценка времени работы AIS алгоритма на этой модельной

задаче – $\Theta(n^2 \log n)$, где n – длина особи[11]. Данная задача имеет один глобальный оптимум – строка из всех единиц.

3.2. Модельная задача LEADINGONES

Следующая модельная задача предлагает в качестве оптимизируемой функции число непрерывно идущих слева направо единиц. Общая формула вычисления этой функции выглядит следующим образом:

$$\text{LEADINGONES}(x) = \sum_{i=1}^n \prod_{j=1}^i x[j].$$

В источнике утверждается, что данная модельная задача решается с помощью рассматриваемого AIS алгоритма за $\Theta(n^2 \log n)$ [19]. Как и предыдущая задача, она имеет один глобальный оптимум, состоящий из всех единиц.

3.3. Модельная задача XDIVK

Эта модельная задача, по сути, является отдалением ландшафта задачи ONEMAX. Здесь в качестве оптимизируемого критерия взята целая часть отношения значения функции ONEMAX к некоторому заранее определенному параметру k , такой, что $n \bmod k = 0$, где n – длина особи[21]. Формула вычисления критерия XDIVK приведена ниже.

$$\text{XDIVK}(x) = \left\lfloor \frac{\text{ONEMAX}(x)}{k} \right\rfloor$$

Для данной модельной задачи еще не проводились теоретические оценки времени работы AIS алгоритмов, потому приведем график (Рис. 3) того, как алгоритмы Random Local Search и B-cell Algorithm справляются с

ней на примере особи длиной 18 и параметром k равным 3. Здесь, и далее в подобных графиках, по горизонтали – шаг алгоритма (номер поколения), по вертикали – значение фитнес функции, вычисленное на данном шаге.

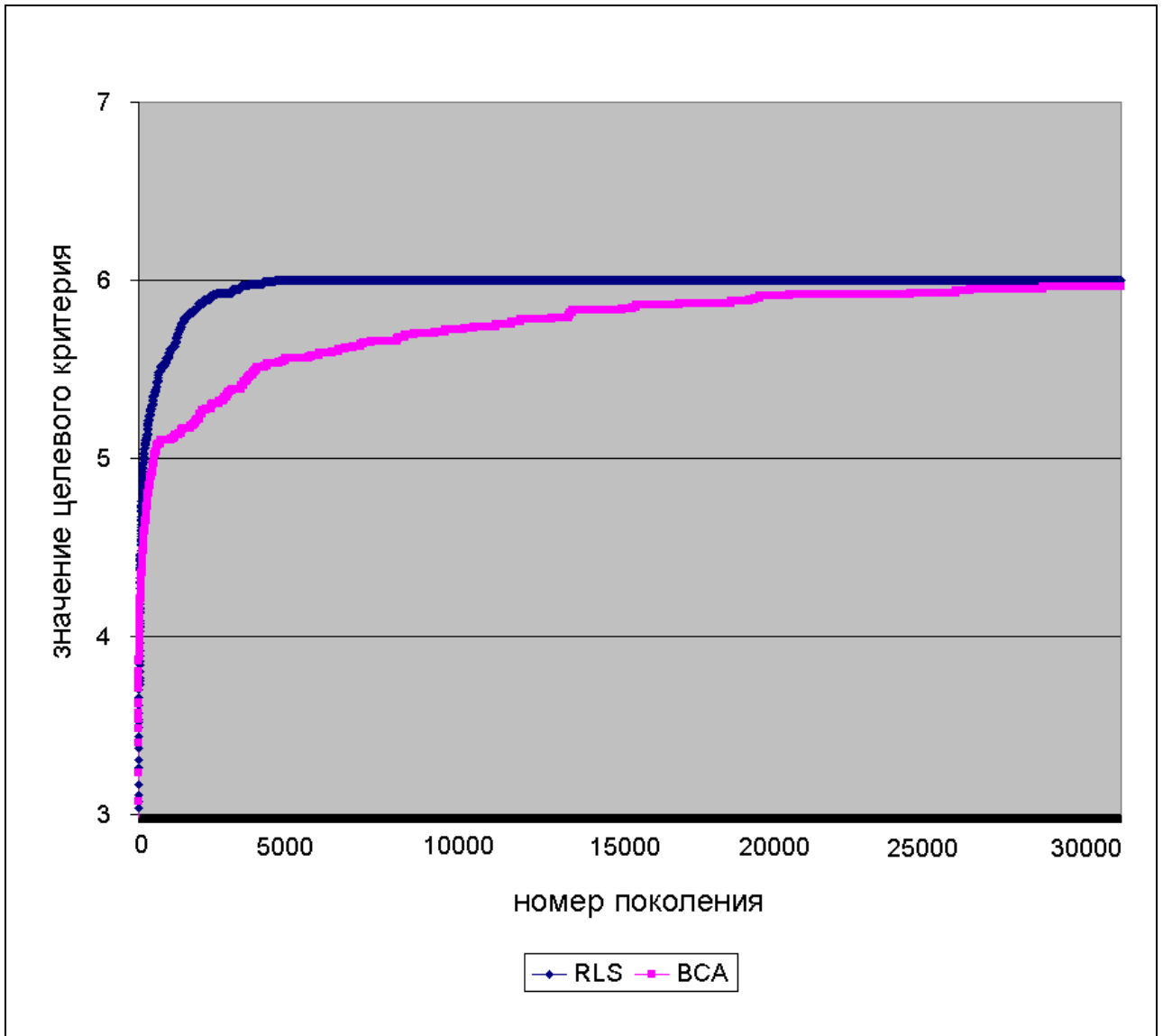


Рис. 3. Решение задачи XdivK алгоритмами RLS и BCA.

Из графика видно, что AIS алгоритм BCA решает задачу, но гораздо медленней, чем эволюционный алгоритм RLS.

3.4. Модельная задача N-IFF

Последняя рассматриваемая модельная задача – Hierarchical-if-and-only-if function, N-IFF [13, 22]. Как и в предыдущих задачах, пространством поиска являются битовые строки фиксированной длины. Они представляются в виде дерева, которое имеет следующую структуру: корнем является исходная строка, ее дети – правая и левая половина этой строки. Далее для каждого из детей аналогично корню формируются дети из правой и левой частей рассматриваемой вершины. Функция вычисляется, как сумма блоков, состоящих из нулей и единиц. Пример вычисления функции для строки 0111 приведен на Рис. 4.

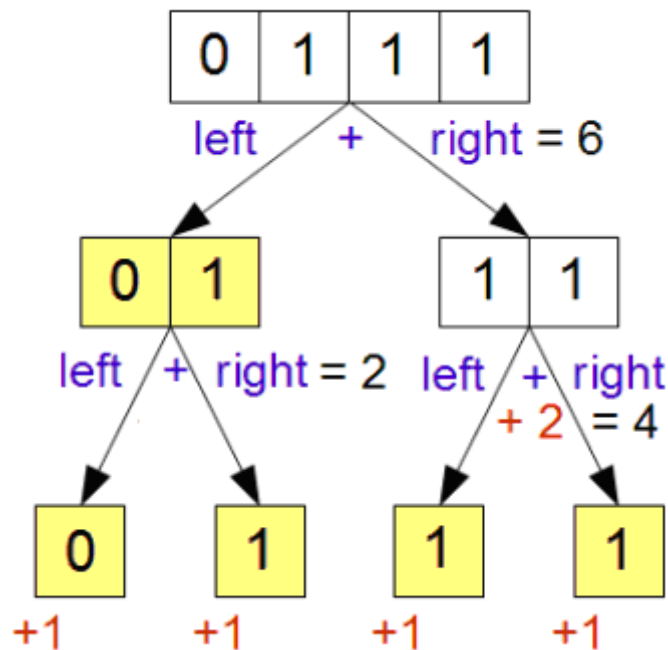


Рис. 4. Вычисление N-IFF строки 0111.

У данной задачи есть два решения: строка из всех нулей и из всех единиц. Рекурсивная формула для вычисления функции приведена ниже:

$$f(B) = \begin{cases} 1 & \text{если } |B| = 1, \text{ иначе} \\ |B| + f(B_L) + f(B_R) & \text{если } \forall i\{b_i = 0\} \text{ или } \forall i\{b_i = 1\} \\ f(B_L) + f(B_R) & \text{в остальных случаях} \end{cases}$$

Теперь пронаблюдаем, как данная модельная задача решается с помощью эволюционных алгоритмов и алгоритмов искусственных иммунных систем. Ниже приведен график (Рис. 5.) вычисления функции приспособленности для особи длиной 1024 тремя алгоритмами: (1+1) эволюционная стратегия, Random Local Search и B-cell Algorithm.

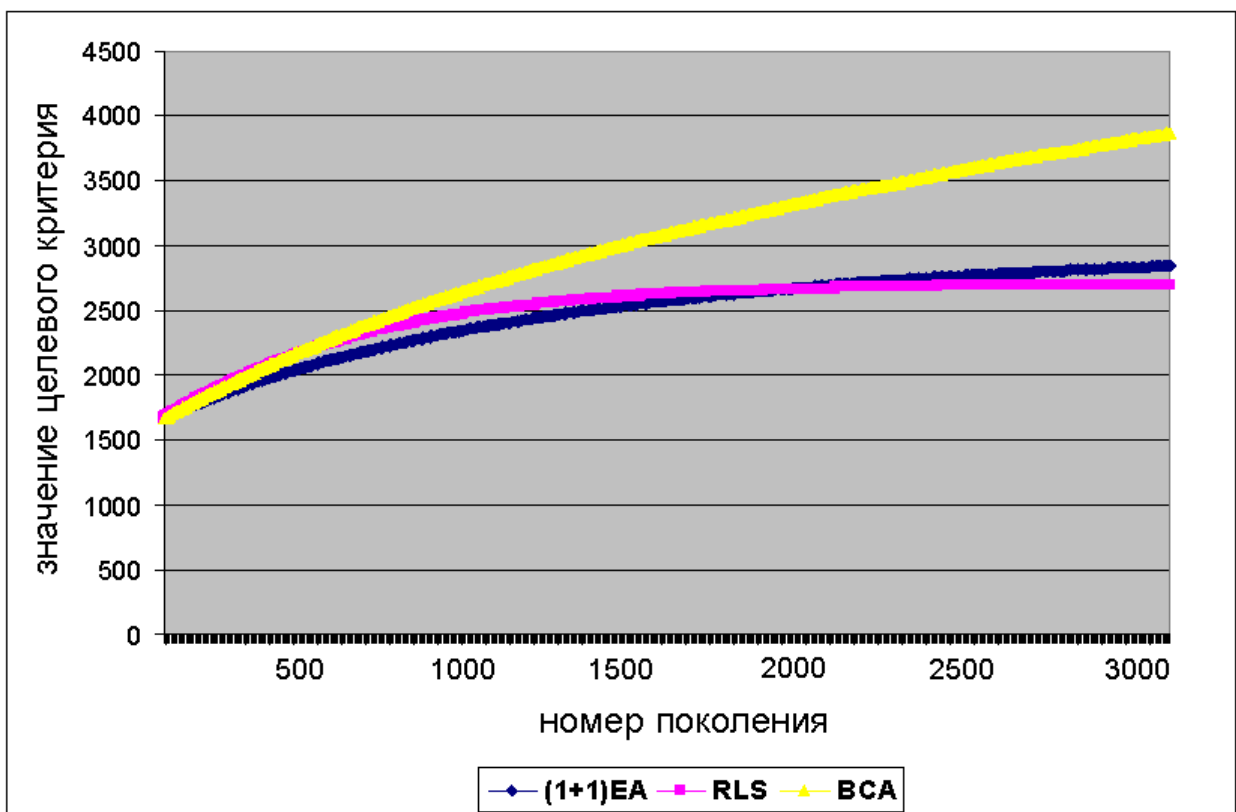


Рис. 5. Вычисление функции H-IFF алгоритмами (1+1)EA, RLS и BSA на протяжении 3000 шагов.

На Рис. 5. видно, что алгоритмы (1+1)EA и RLS застревают в локальном оптимуме, в то время, как алгоритм BSA решает задачу. В

источнике приводится теоретическая оценка времени работы AIS алгоритма на этой задаче – $O(n^3 \log n)$, где n – длина особи[1].

3.5. Выводы по главе 3

В этой главе были продемонстрированы особенности AIS алгоритмов, выявленные при решении ими поставленных модельных задач. Таким образом, показано, что алгоритмы искусственных иммунных систем уступают эволюционным алгоритмам при решении задач с одним оптимумом. Но для тех задач, в которых присутствует множество локальных оптимумов, как, например, в N-IFF, использование AIS позволяет избежать застревания в них и эффективно решать задачу.

Глава 4. Сравнение эффективности применения AIS совместно и без multi-objectivization

В данной главе приводятся результаты экспериментального сравнения эффективности сведения однокритериальных задач оптимизации к многокритериальным. Выбор вспомогательных критериев основывался на исследованиях из предыдущей главы.

Так же приведены результаты проведенного анализа статистической различимости методов с помощью непарного и парного тестов Уилкоксона[23]. Тесты проводились по числу поколений алгоритма искусственных иммунных систем, потребовавшихся для достижения оптимума задачи и по усредненным максимальным значениям целевого критерия, достигнутому за ограниченное количество шагов.

4.1. Игнорирование неэффективного критерия

К первому типу задач отнесем те, в которых при решении участвовали неэффективные или даже мешающие критерии.

4.1.1. Модельная задача ONEMAX с мешающим критерием ZERO MAX

Рассмотрим многокритериальный вариант задачи ONEMAX, в котором в качестве дополнительного критерия выступает ZERO MAX. Данный критерий является мешающим, так как его оптимум не совпадает с оптимумом

ONEMAX и состоит из всех нулей. Известно, что данная задача успешно решается эволюционными алгоритмами, которые решают ее так же быстро, как и просто ONEMAX.

Для данной модельной задачи в экспериментах использовались следующие параметры:

- Длина особи 20-90 бит.
- Целевой критерий: число единиц в особи.
- Мешающий критерий: число нулей в особи.
- Скорость обучения: EA – 0,5, AIS – 0,7.
- Дисконтный фактор: EA – 0,5, AIS – 0,7.
- Вероятность исследования среды: EA – 0, AIS – 0.

n	EA	EA+RL	AIS	AIS+RL
20	60,6(25,1)	59,0(23,3)	931,1(552,2)	874,3(493,2)
30	103,8(39,4)	97,0(35,9)	2089,7(896,8)	2162,3(1069,6)
40	138,3(38,3)	152,1(55,7)	4235,6(1730,2)	4490,4(2052,7)
50	181,9(53,4)	187,5(53,3)	7520,6(3323,1)	7348,5(3826,3)
60	231,6(74,5)	236,1(69,7)	11602,1(4104,2)	11376,8(5443,4)
70	286,4(91,9)	293,5(88,8)	14802,1(4539,8)	15473,6(5679,2)
80	320,8(89,1)	362,1(109,7)	20810,1(8079,7)	21302,1(8377,3)
90	400,6(106,6)	399,1(100,7)	25990,4(9361,7)	28543,3(10554,5)

Таблица 1. Результаты AIS и EA алгоритмов на задачах ONEMAX и ONEMAX с мешающим критерием ZEROMAX.

В таблице 1, и далее в подобных таблицах, приведены усредненные по 100 запускам значения числа шагов, потребовавшихся алгоритму для достижения оптимума. Показаны результаты применения эволюционного (Random Local Search) и AIS (B-cell Algorithm) алгоритмов без и с использованием обучения с подкреплением для различных длин особи. В скобках указано стандартное отклонение результата. Из этого можно заключить, что AIS+RL способен в процессе оптимизации успешно игнорировать мешающий критерий. Данный вывод подтвержден статистическим тестом, результаты которого приведены в таблице 6.

4.1.2. Модельная задача MH-IFF

Эта модельная задача является многокритериальным вариантом описываемой в пункте 3.4. задачи H-IFF. Значения критериев оптимизации вычисляются по формуле:

$$f_n(B) = \begin{cases} 0 & \text{если } |B| = 1 \text{ и } b_1 \neq n, \text{ иначе} \\ 1 & \text{если } |B| = 1 \text{ и } b_1 = n, \text{ иначе} \\ |B| + f_n(B_L) + f_n(B_R) & \text{if } \forall i \{b_i = n\} \\ f_n(B_L) + f_n(B_R) & \text{в остальных случаях} \end{cases}$$

Ввод дополнительных критериев, учитывающих только блоки из единиц или только блоки из нулей, помог в решении этой задачи эволюционными алгоритмами, которые ранее застревали в локальном оптимуме.

Для данной модельной задачи в экспериментах использовались следующие параметры:

- Длина особи 64 бита.
- Целевой критерий: H-IFF.
- Вспомогательные критерии:
 - Длина блоков из нулей.
 - Длина блоков из единиц.
- Мешающий критерий:
 - Совпадение со строкой вида 10101010101...

Так как длина особи 64, максимальным значением функции H-IFF и оптимумом задачи будет значение 448.

В таблице 2 приведены результаты решения поставленной задачи для однокритериального варианта, для варианта с Q-обучением и вспомогательными критериями и варианта с Q-обучением и вспомогательными и мешающим критерием. Указаны максимальное достигнутое значение целевого критерия, среднее достигнутое значение целевого критерия, шаг, на котором был найден оптимум, процент удачных запусков и медиана числа потребовавшихся для достижения оптимума шагов по 30 запускам. Зеленым выделены результаты алгоритма искусственных иммунных систем ВСА. Для сравнения в таблице так же приведены результаты решения этой задачи с помощью эволюционных алгоритмов Random Local Search (RLS) и (1+1) эволюционная стратегия с вероятностью однородной мутации $\frac{1}{n}$ и $\frac{2}{n}$.

	Алгоритм	Лучшее значение целевого критерия	Среднее значение целевого критерия	Среднее число поколений до достижения оптимума	Процент успешных запусков	Медиана числа поколений до достижения оптимума
Н-IFF	RLS	212	171	----	0	----
	BCA	448	448	3177	100	2470
	EA(1/n)	288	246	----	0	----
	EA(2/n)	366	270	----	0	----
МН-IFF	RLS	448	238	277	23	299
	BCA	448	448	3234	100	2333
	EA(1/n)	448	331	656	37	563
	EA(2/n)	448	346	669	43	796
МН-IFF с мешающим критерием	RLS	448	219	244	16	246
	BCA	448	448	2396	100	2210
	EA(1/n)	448	337	624	40	520
	EA(2/n)	449	339	584	40	561

Таблица 2. Результаты AIS и EA алгоритмов на задачах Н-IFF, МН-IFF с вспомогательными и МН-IFF с вспомогательными и мешающим критериями.

Как можно наблюдать BCA сам по себе хорошо решает Н-IFF – всегда находит решение, поэтому положительного эффекта от использования вспомогательных критериев, как это верно для эволюционных алгоритмов, не наблюдается. Дополнительные критерии не особо помогают в решении задачи, но и не портят результат. Так как результаты BCA с хорошими и с плохими критериями не различаются, можно утверждать, что AIS+RL эффективно игнорирует плохой критерий. Последнее утверждение подтверждены статистическим тестом, результаты которого приведены в таблице 6.

4.2. Задачи с помогающим критерием

Здесь рассмотрим задачи, в которых в процессе оптимизации используются помогающие критерии, то есть такие, которые ускоряют процесс поиска оптимума.

4.2.1. Модельная задача XDIVK

От описанной в пункте 3.3. задачи ее отличает наличие вспомогательного критерия ONEMAX. Имея тот же оптимум, что и XDIVK, ONEMAX позволяет в ситуациях, когда две строки для XDIVK не различимы, выбрать лучшую.

Для данной модельной задачи в экспериментах использовались следующие параметры:

- Длина особи n – 12-50 бит.
- x – число единиц в особи.
- Целевой критерий: $\left\lfloor \frac{x}{k} \right\rfloor$, $n \bmod k = 0$.
- Вспомогательный критерий: число единиц в особи.
- Скорость обучения: EA – 0,5, AIS – 0,7.
- Дисконтный фактор: EA – 0,5, AIS – 0,7.
- Вероятность исследования среды: EA – 0, AIS – 0.

В таблице 3 приведены результаты решения задачи XDIVK с вспомогательным критерием ONEMAX. Примечательно, что использование обучения с подкреплением улучшает эффективность AIS алгоритма приблизительно в два раза. Перед некоторыми значениями, выделенными

скобками, находятся знаки процента. Это те случаи, когда из 100 запусков не все достигли оптимума целевого критерия. Соответственно указан процент удачных запусков, а в скобках – результат для удачных запусков.

n	k	EA	EA+RL	AIS	AIS+RL
12	2	123,8(95,5)	66,4(71,9)	598,4(514,2)	387,7(386,9)
16	2	192,3(149,3)	111,8(117,5)	1451,9(1376,2)	959,8(1071,3)
20	2	310,7(224,1)	202,9(200,6)	3142,1(2457,0)	1567,2(1289,3)
18	3	1077,3(1151,0)	637,5(983,1)	5372,1(6453,8)	3498,3(5078,0)
24	3	3142,9(2758,9)	1336,1(1960,5)	21639,0(19305,0)	10156,9(12941,3)
30	3	5239,7(4994,4)	3058,1(4466,1)	48915,6(50056,5)	20166,5(34307,2)
24	4	11820,7(10540,0)	5783,6(10885,3)	79280,7(85457,3)	44855,4(73456,1)
32	4	48746,2(49494,9)	25662,7(40975,7)	93%(337810,6(258932,2))	99%(123298,3(176854,7))
40	4	116844,0(103708,6)	46239,4(67101,1)	72%(455946,1(302672,3))	75%(154427,3(274338,8))
30	5	168184,7(170838,8)	82513,2(159791,7)	61%(494223,6(272875,5))	85%(186701,6(297428,9))
40	5	77%(420403,1(295541,1))	87%(127970,8(215401,7))	14%(539217,6(353954,8))	58%(52390,5(162715,2))
50	5	36%(444729,2(329020,4))	61%(144981,4(255626,3))	2%(592953,5(132911,3))	58%(38212,2(164846,2))

Таблица 3. Результаты AIS и EA алгоритмов на задачах XDIVK и XDIVK с вспомогательным критерием ONEMAX.

На Рис. 6 приведен график вычисления фитнес функции алгоритмами EA и AIS для однокритериального и многокритериального варианта задачи. В эксперименте использовались особи длины 18 и параметр k равный 3. Этот график наглядно демонстрирует улучшение эффективности алгоритма на основе искусственных иммунных систем после ввода дополнительных критериев.

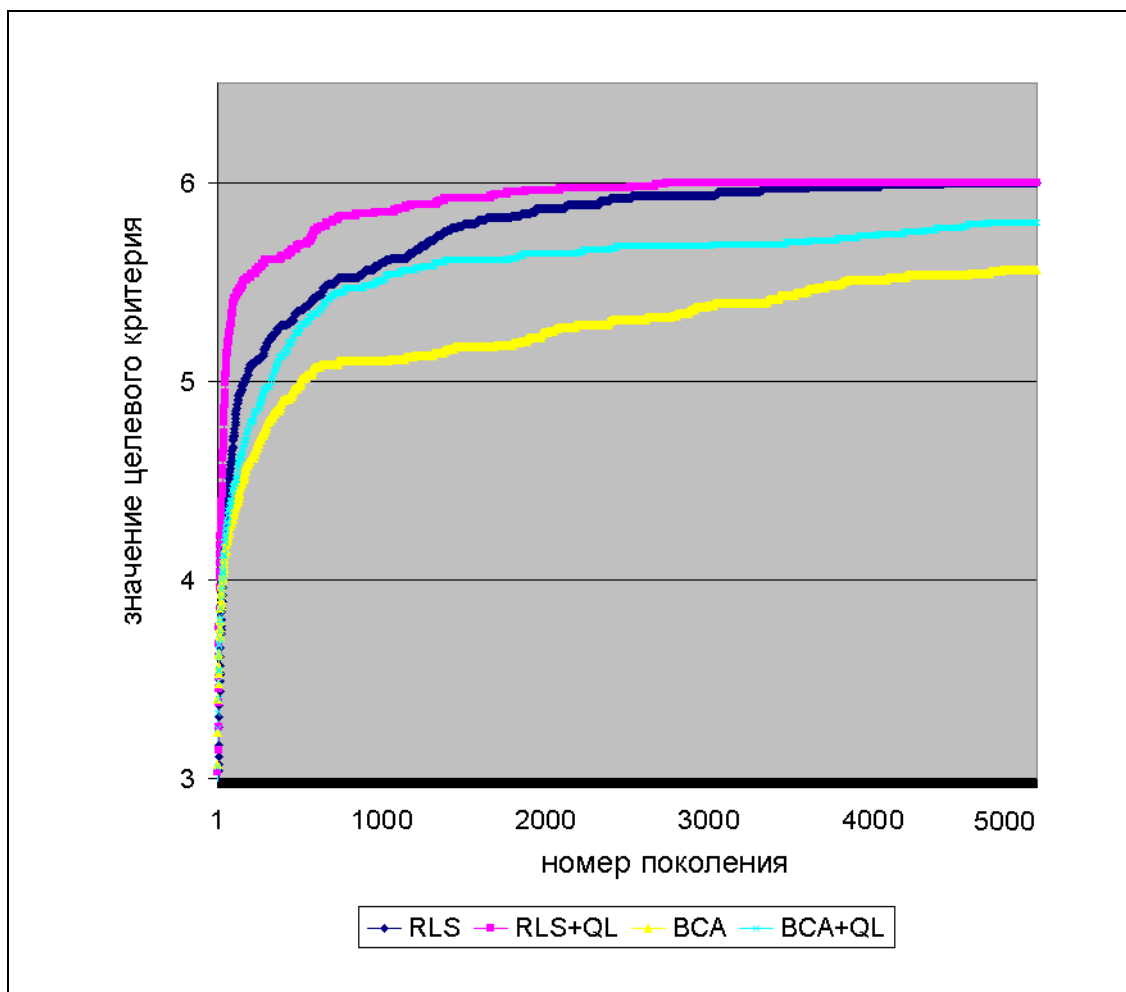


Рис. 6. График вычисления фитнес функции алгоритмами EA и AIS для задач XDIVK и XDIVK с вспомогательным критерием ONEMAX (длина особи – 18, делитель $k - 3$).

Далее приведены гистограммы числа достижений оптимума задачи от значения шага, на котором оптимум был достигнут. То есть, сколько из 50000 запусков алгоритма завершили поиск решения на указанном числе поколений. На рис. 7. показано использование AIS без обучения с подкреплением, на рис. 8. – с Reinforcement Learning. Заметно изменение характера гистограммы, а именно наличие ярко выраженного пика на небольшом числе поколений. Таким образом, демонстрируется ускорение поиска оптимума после ввода дополнительных критериев. Это улучшение эффективности работы AIS алгоритма подтверждается статистическим тестом, результат которого можно посмотреть в таблице 6.

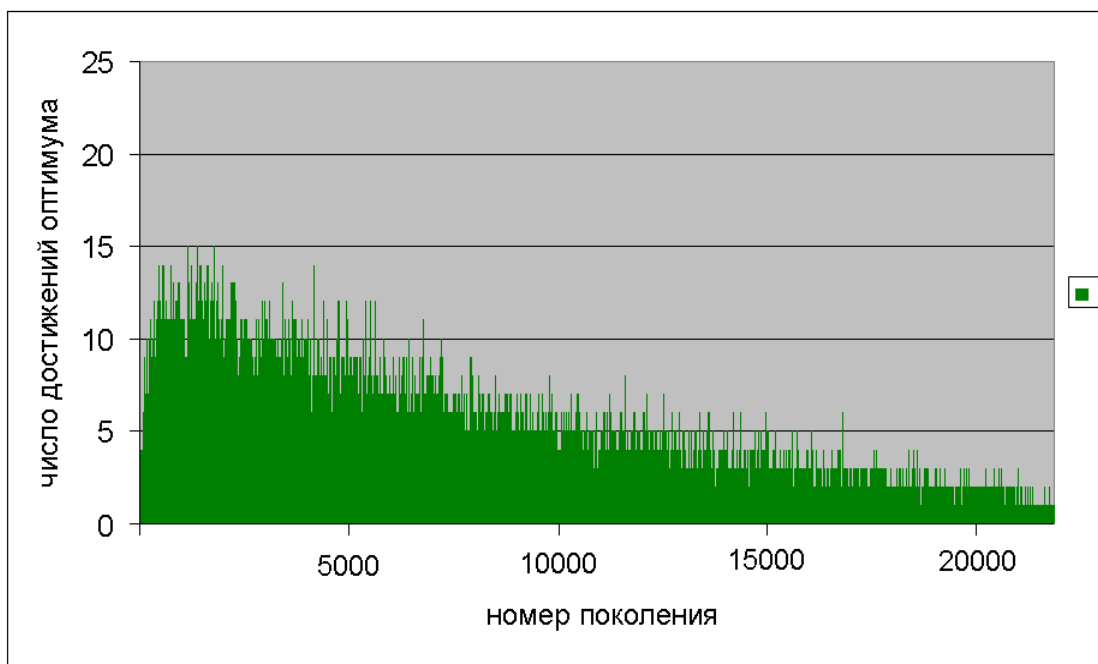


Рис. 7. Гистограмма числа достижений оптимума задачи от числа поколений алгоритма AIS без обучения с подкреплением.

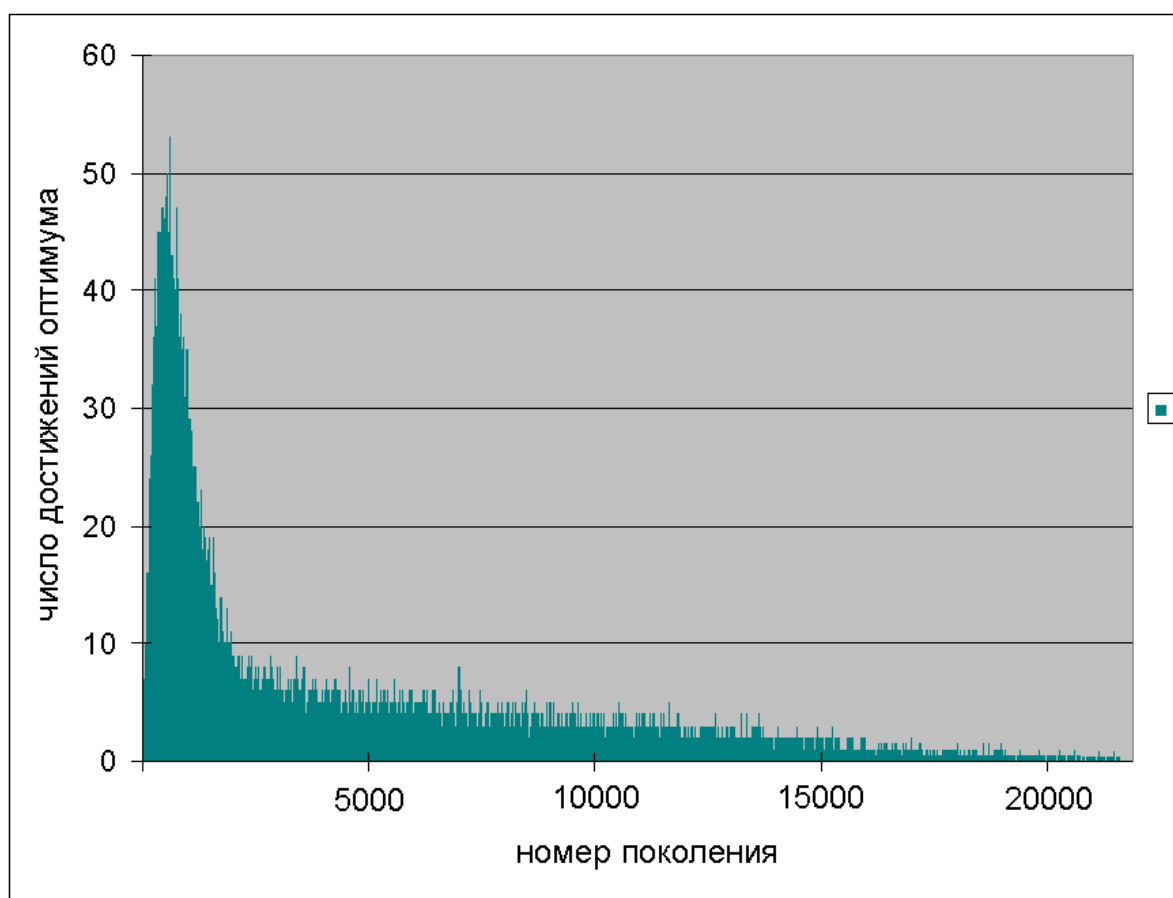


Рис. 8. Гистограмма числа достижений оптимума задачи от числа поколений алгоритма AIS с обучением с подкреплением.

4.2.2. Модельная задача MINMAX

Еще один многокритериальный вариант задачи XDIVK, описанной в пункте 3.3. – задача MINMAX. Она отличается от рассматриваемой ранее многокритериальной XDIVK тем, что в качестве вспомогательных критериев здесь используется не уточняющая ландшафт ONEMAX, а вспомогательные функции $\min(x, d)$ и $\max(x, d)$, которые выбирают соответственно минимум и максимум от числа единиц в особи и точки переключения d . Эта задача интересна тем, что покажет, способен ли AIS+RL динамически выбирать наиболее выгодный критерий на разных этапах оптимизации.

Для данной модельной задачи в экспериментах использовались следующие параметры:

- Длина особи 200-600 бит.
- x – число единиц в особи.
- d – точка переключения.
- Целевой критерий: $\left\lfloor \frac{x}{k} \right\rfloor, n \bmod k = 0$.
- Вспомогательные критерии:
 - $H1 = \min(x, d)$
 - $H2 = \max(x, d)$
- Скорость обучения: EA – 0,5, AIS – 0,7.
- Дисконтный фактор: EA – 0,5, AIS – 0,7.
- Вероятность исследования среды: EA – 0, AIS – 0.

Так как на особях небольшой длины затруднительно показать эффективное переключение критериев, будем рассматривать особи большой длины. Для них достижение оптимума за 10^6 итераций не всегда возможно, поэтому будем усреднять по 100 запускам максимальное достигнутое значение оптимизационного критерия. В качестве точки переключения бралось значение 60-80% от длины особи. В таблице 4 приведены результаты поставленного эксперимента. Данные результаты были проверены на статистическую различимость с помощью парного теста Уилкоксона. Результаты тестов приведены в разделе 4.4.

n	k	d	EA	EA+RL	AIS	AIS+RL
200	5	165	37,6(0,6)	37,8(0,6)	35,3(0,6)	35,4(0,8)
300	10	245	23,6(0,6)	24,3(1,5)	22,5(0,6)	23,2(0,8)
400	10	266	31,2(0,6)	32,1(2,6)	29,6(0,6)	29,8(0,6)
400	10	333	31,4(0,6)	32,1(1,3)	29,7(0,7)	31,2(1,7)
500	10	415	39,0(0,7)	40,8(2,7)	36,7(0,7)	38,7(2,4)
600	15	460	27,8(0,6)	29,8(3,5)	27,0(0,5)	28,6(1,7)
600	10	495	46,6(0,8)	48,0(2,6)	43,8(0,8)	45,8(2,5)

Таблица 4. Результаты AIS и EA алгоритмов на однокритериальной и многокритериальной задаче MINMAX для больших значений длины особи.

Заметим, что стандартное отклонение для MINMAX теперь в процентном отношении гораздо меньше, чем для XDIVK. Получается, что значение целевого критерия за фиксированное число поколений менее зашумлено, чем за число поколений, необходимое для достижения оптимума. Это косвенно подтверждает тот факт, что самый сложный этап – в конце, но при фиксированном числе поколений алгоритм не доходит до него.

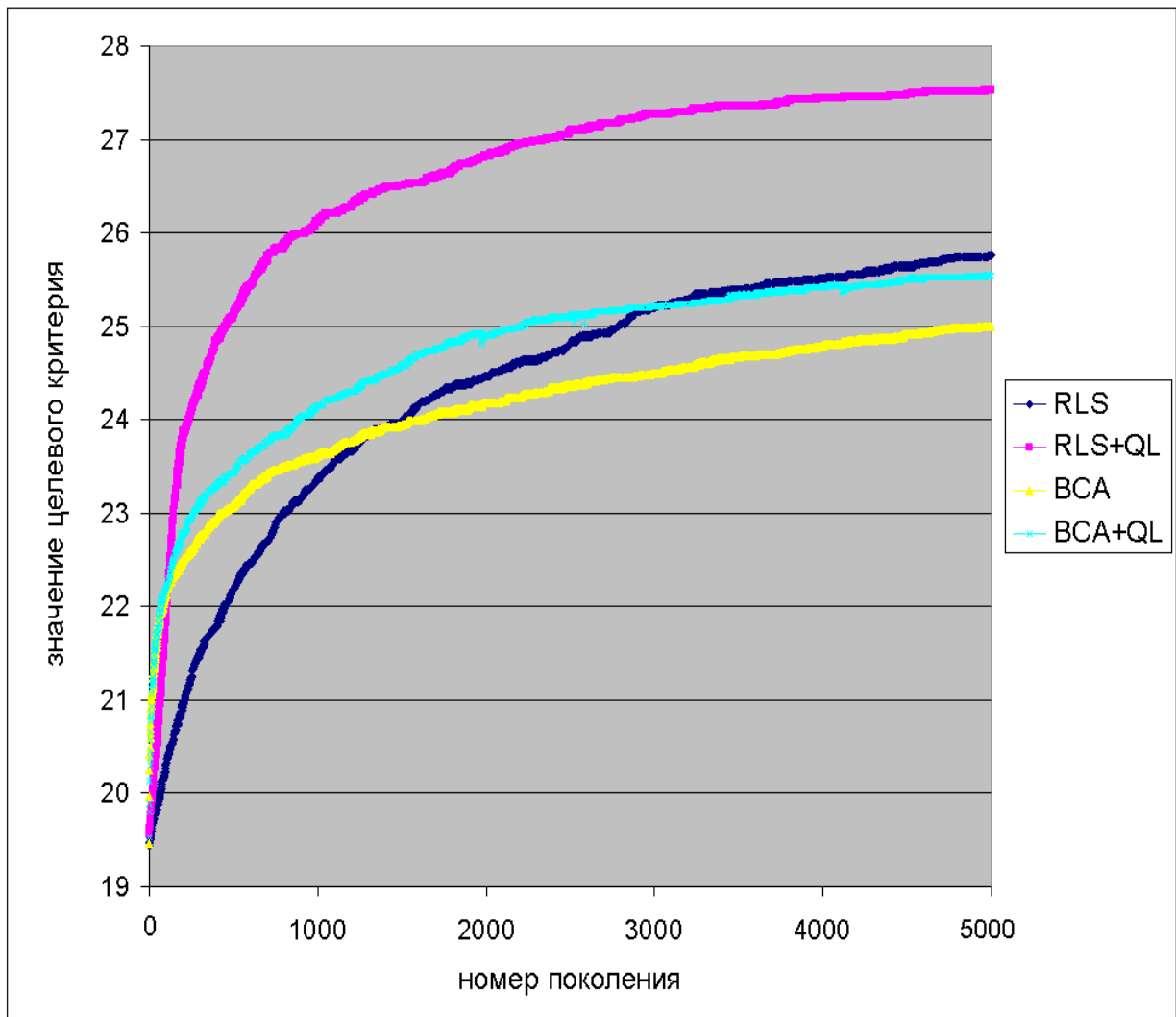


Рис 9. График вычисления фитнес функции алгоритмами EA и AIS для однокритериальной и многокритериальной задачи MINMAX (длина особи – 400, параметр $k = 10$, точка переключения $d = 266$).

Так же примечательно то, что достигаемые AIS+RL значения очень похожи на те, что получаются в результате работы эволюционного алгоритма. Таким образом, применение динамического выбора критериев позволяет алгоритмам искусственных иммунных систем догнать эволюционные алгоритмы, и AIS+RL ведет себя как алгоритм, совмещающий достоинства AIS и EA.

На Рис. 9. приведен график вычисления фитнес функции алгоритмами EA и AIS для однокритериального и многокритериального варианта задачи. В эксперименте использовались особи длины 400, параметр k равный 10 и точки переключения $d - 226$. Данный график наглядно демонстрирует, что в начале процесса оптимизации алгоритмы искусственных иммунных систем более эффективны, чем эволюционные алгоритмы, и ввод дополнительных критериев позволяет заметно увеличить этот промежуток.

Рассмотрим теперь, способен ли AIS+RL динамически выбирать наиболее выгодный критерий на разных этапах оптимизации. Проведем исследование для длины особи 500, параметра $k - 10$ и точки переключения – 415. На Рис. 10 представлен график процентного соотношения выбора различных критериев алгоритмом. То есть процент запусков, когда определенный критерий был применен к особи с определенным количеством единиц в ней. Заметим, что до значения точки переключения агентом выбираются критерии $min(x, d)$ и целевой, но по достижении точки переключения, практически все решения агента сводятся к выбору критерия $max(x, d)$. Таким образом, наглядно показано, что AIS+RL не только улучшает работу алгоритма искусственных иммунных систем для этой задачи, но и позволяет динамически выбирать критерии выгодные на разных этапах оптимизации.

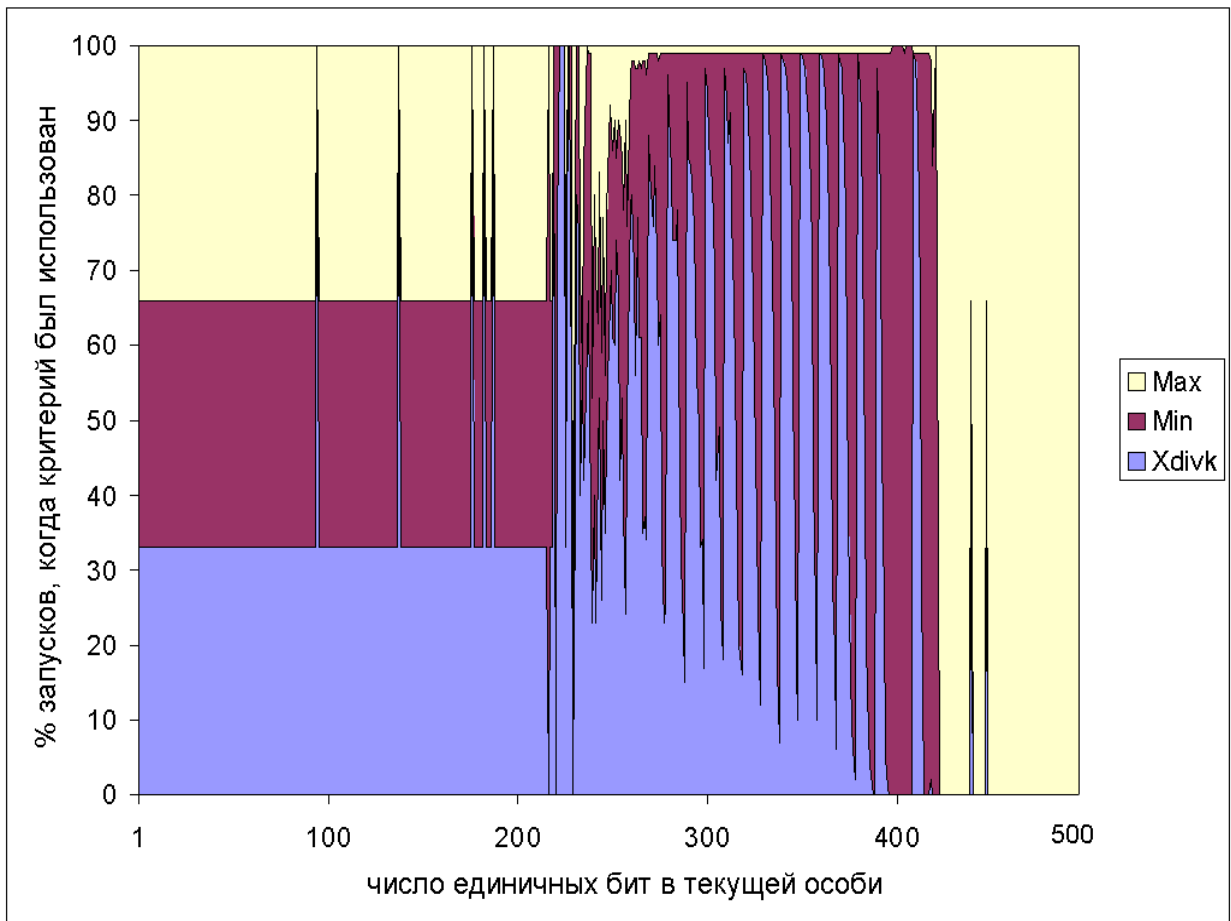


Рис. 10. Соотношение выбора критериев алгоритмом на каждом из числа единиц в текущей особи для задачи MINMAX.

4.3. Неэффективный выбор вспомогательного критерия

Последней рассматриваемой задачей будет LEADINGONES с вспомогательным критерием ONEMAX. Эволюционными алгоритмами задача ONEMAX решается асимптотически быстрее, чем LEADINGONES. Это и делает ее эффективным помогающим критерием.

В экспериментах использовались следующие параметры:

- Длина особи n – 20-90 бит.
- Целевой критерий: число ведущих единиц в особи.
- Вспомогательный критерий: число единиц в особи.
- Скорость обучения: EA – 0,5, AIS – 0,7.
- Дисконтный фактор: EA – 0,5, AIS – 0,7.
- Вероятность исследования среды: EA – 0, AIS – 0.

n	EA	EA+RL	AIS	AIS+RL
20	199,8(78,7)	134,2(81,5)	729,8(473,5)	712,8(388,1)
30	433,5(137,1)	322,5(203,7)	1800,9(831,9)	1685,1(1008,7)
40	796,3(194,6)	497,3(339,4)	3621,0(2159,6)	3626,5(1923,9)
50	1299,6(297,8)	695,2(551,4)	5774,8(2611,8)	6139,6(2979,4)
60	1821,0(416,5)	924,2(733,9)	9111,8(4159,6)	8500,7(4141,5)
70	2500,9(513,9)	1432,3(1064,3)	12015,0(5402,6)	13056,5(5294,6)
80	3152,8(652,3)	1526,8(1348,0)	16959,8(9366,8)	17238,3(7896,9)
90	4112,6(694,9)	2274,9(1891,0)	21235,8(8741,9)	19589,5(7241,8)

Таблица 5. Результаты AIS и EA алгоритмов на задачах LEADINGONES и LEADINGONES с неэффективным критерием ONEMAX.

В таблице 5 приведены усредненные по 100 запускам результаты решения модельной задачи. С вводом дополнительного критерия оптимизации, улучшения эффективности работы алгоритма искусственных иммунных систем не наблюдается. Это происходит по причине того, что

теоретические оценки задач LEADINGONES и ONEMAX для алгоритма искусственных иммунных систем совпадают[11], то есть критерий ONEMAX для исследуемой задачи не является улучшающим. Отсюда следует, что для данной модельной задачи использование вспомогательного критерия ONEMAX бесполезно.

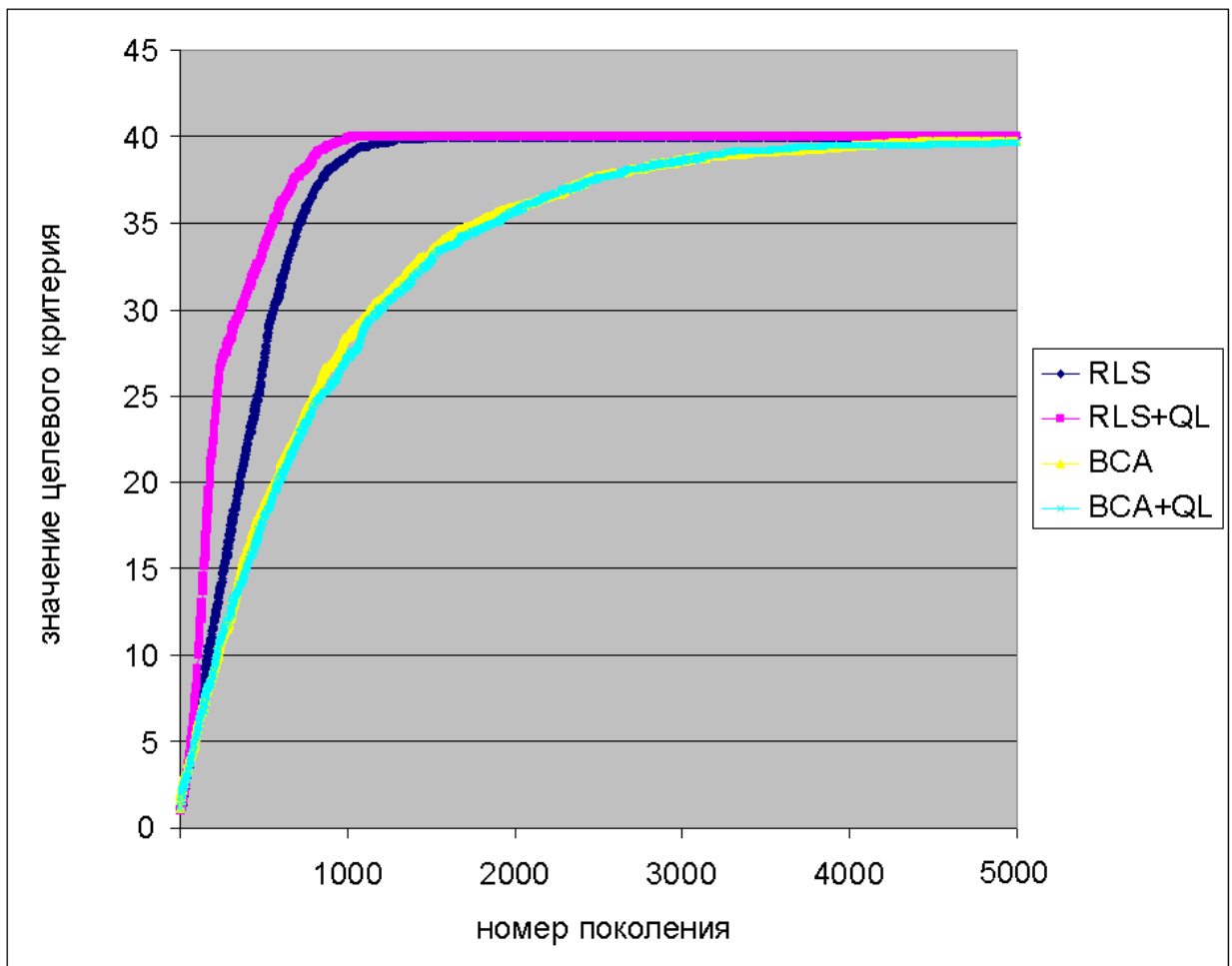


Рис. 11. График вычисления фитнес функции алгоритмами EA и AIS для однокритериальной и многокритериальной задачи LEADINGONES.

Однако можно утверждать, что даже выбор неэффективного критерия не мешает алгоритму решить задачу. Это подтверждается графиком (Рис. 11), на котором видно практически полное совпадение значений целевого критерия для AIS и AIS+RL на каждом из шагов алгоритма.

Эти результаты показывают, что, так как эффективность вспомогательных критериев для AIS не совпадает с их оценками для EA, то выбор вспомогательного критерия для задачи требует дополнительного исследования. В таблице 6 приведены результаты статистического теста для этой задачи.

4.4. Результаты статистического теста

В работе были проведены статистические тесты на основе парных и непарных тестов Уилкоксона[23]. Непарный тест проводился по двум независимым выборкам по 100 значений для каждого из алгоритмов. Исследуемые значения – число вычислений функции приспособленности, потребовавшихся для нахождения оптимума. Результаты непарного теста с сопутствующими выводами приведены в таблице 6. Значения p-value меньше 0.05 подтверждают гипотезу о статистической различимости рассматриваемого метода и однокритериального варианта алгоритма. Такая ситуация наблюдается для задач XDIVK и MINMAX с особями небольшой длины. Для них ввод дополнительных критериев заметно улучшил работу AIS алгоритма. Для остальных задач полученные значения p-value не дают возможности сделать вывод о статистической различимости выборок. Что так же является положительным результатом, ведь таким образом подтверждается предположение об эффективном игнорировании мешающего критерия в задачах ONEMAX и MH-IFF. Так же подтверждается, что использование неподходящих или лишних критериев, как например, в задачах LEADINGONES и MH-IFF без мешающего критерия, не мешает процессу поиска оптимума

Задача	Целевой критерий	Вспомогательный критерий	Мешающий критерий	p-value	Комментарий
MH-IFF	H-IFF	H-IFF(0), H-IFF(1)	-	0.3173	Улучшать уже нечего
	H-IFF	H-IFF(0), H-IFF(1)	101010101....	0.2682	Эффективное игнорирование плохого критерия
OneMax	OneMax	-	ZeroMax	0.5732	
Xdivk	Xdivk	OneMax	-	<2.2e-16	Улучшение эффективности однокритериальной задачи при сведении к многокритериальной
MinMax	Xdivk	min(x, d), max(x, d)	-	<2.2e-16	
LeadingOnes	LeadingOnes	OneMax	-	0.2595	Бесполезный вспомогательный критерий

Таблица 6. Результаты непарного теста Уилкоксона.

Парный тест проводился для ситуации, когда достижение оптимума целевого критерия было невозможно менее чем за 10^6 вычислений, как, например, в задаче MINMAX с большими значениями длины особи. Для этой задачи случайным образом 20 и 40 раз генерировались пары делителя (k : 5 – 15) и длины особи ($n = m*k$, множитель m : 30–60). Затем, используя параметр точки переключения d взятый около 80% от длины особи, было проведено усреднение результатов запусков алгоритмов AIS и AIS+RL на этих данных. На полученных значениях был проведен парный тест Уилкоксона, давший результаты $1.907e-06$ и $1.273e-11$ соответственно, что подтверждает статистическую различимость выборок. Отсюда можно сделать вывод, что использование метода ввода вспомогательных критериев увеличивает эффективность AIS алгоритмов и на длительных вычислениях.

4.5. Выводы по главе 4

В данной главе наглядно продемонстрированы особенности сведения однокритериальных задач оптимизации, решаемых с помощью искусственных иммунных систем к многокритериальным. Показано, что эффективность данных алгоритмов может быть повышена при грамотном подборе вспомогательных оптимизируемых величин. Проведены статистические тесты, подтверждающие выводы, сделанные в процессе исследования.

Заключение

Исследования показали, что эффективность алгоритмов оптимизации на основе искусственных иммунных систем может быть повышена с помощью введения вспомогательных критериев. Однако критерии, которые являются вспомогательными для эволюционных алгоритмов, не всегда являются таковыми для искусственных иммунных систем. Таким образом, при работе с искусственными иммунными системами может возникать необходимость подбора специфических для них вспомогательных критериев.

Известно, что алгоритмы оптимизации на основе искусственных иммунных систем отличаются большей производительностью на начальных этапах оптимизации, чем эволюционные алгоритмы. В работе показано, что на наблюдаемых примерах, использование вспомогательных критериев позволяет увеличить промежуток, на котором AIS алгоритмы эффективнее эволюционных алгоритмов.

ИСТОЧНИКИ

1. *Jansen, T., Zarges, C.:* Understanding randomised search heuristics. Lessons from the evolution of theory: A case study. In: Proceedings of the 20th International Conference on Soft Computing MENDEL, pp. 293–298 (2014)
2. *Segura, C., Coello, C.A.C., Miranda, G., León, C.:* Using multi-objective evolutionary algorithms for single-objective optimization. 4OR 3(11), 201–228 (2013)
3. *Mitchell, M.* An introduction to Genetic Algorithms. MIT Press. (1996)
4. *Farmer, J.D., Packard N., Perelson, A.:* The immune system, adaptation and machine learning. Physica D, vol. 2, pp. 187–204(1986)
5. *Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.:* Self-nonsel self discrimination in a computer. Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy. Los Alamitos, CA. pp. 202–212 (1994)
6. *Timmis, J., Neal, M., Hunt, J.:* An artificial immune system for data analysis. BioSystems 55 (1): 143–150 (2000)
7. *de Castro, L. N., Von Zuben, F. J.:* Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems (IEEE) 6 (3): 239–251 (2002)
8. *Kelsey, J., Timmis, J.:* Immune inspired somatic contiguous hypermutations for function optimisation, in: E. Cantú-Paz, J.A. Foster, K. Deb, L. Davis, R. Roy, U.-M. O’Reilly, H.-G. Beyer, R.K. Standish, G. Kendall, S.W. Wilson, M. Harman, J. Wegener, D. Dasgupta, M.A. Potter, A.C. Schultz, K.A. Dowsland, N. Jonoska, J.F. Miller (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference, GECCO, Springer, pp. 207–218 (2003)

9. *Clark, E., Hone, A., Timmis, J.:* A Markov chain model of the b-cell algorithm, in: C. Jacob, M. Pilat, P. Bentley, J. Timmis (Eds.), Proceedings of the International Conference on Artificial Immune Systems, ICARIS, Springer, pp. 318–330 (2005)
10. *Jansen, T., Zarges, C.:* A theoretical analysis of immune inspired somatic contiguous hypermutations for function optimization, in: P.S. Andrews, J. Timmis, N.D.L. Owens, U. Aickelin, E. Hart, A. Hone, A.M. Tyrrell (Eds.), Proceedings of the International Conference on Artificial Immune Systems, ICARIS, Springer, pp. 80–94 (2009)
11. *Jansen, T., Zarges, C.:* Analyzing different variants of immune inspired somatic contiguous hypermutations. *Theoretical Computer Science*, 412(6):517–533 (2011)
12. *Jensen, M.T.:* Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation: Evolutionary Computation Combinatorial Optimization. *Journal of Mathematical Modelling and Algorithms* 3(4), 323–347 (2004)
13. *Knowles, J.D., Watson, R.A., Corne, D.:* Reducing Local Optima in Single-Objective Problems by Multiobjectivization. In: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, pp. 269–283. Springer-Verlag (2001)
14. *Neumann, F., Wegener, I.:* Can Single-Objective Optimization Profit from Multiobjective Optimization? In: Multiobjective Problem Solving from Nature, Natural Computing Series, pp. 115–130. Springer Berlin Heidelberg (2008)
15. *Buzdalova A., Buzdalov M.:* Increasing Efficiency of Evolutionary Algorithms by Choosing between Auxiliary Fitness Functions with Reinforcement

Learning / Proceedings of the International Conference on Machine Learning and Applications. Vol. 1. P. 150–155. (2012)

16. *Sutton, R.S., Barto, A.G.*: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, USA (1998)
17. *Corne, D.W., Knowles, J.D., Oates, M. J.*: The pareto envelope-based selection algorithm for multiobjective optimisation. In M. S. et al. (Ed.), Parallel Problem Solving from Nature – PPSN VI, Berlin, pp. 839–848. Springer (2000)
18. *Николенко С. И., Тулупьев А. Л.*: Самообучающиеся системы. (2009)
19. *Oliveto, P.S., He, J., Yao, X.*: Time Complexity of Evolutionary Algorithms for Combinatorial Optimization: A Decade of Results. International Journal of Automation and Computing 4(3), 281–293 (2007)
20. *Droste, S., Jansen, T., Wegener, I.*: On the analysis of the (1+1) evolutionary algorithm. Theoretical Computer Science, 276(1–2):51–81 (2002).
21. *Buzdalov, M., Buzdalova, A.*: Onemax Helps Optimizing XdivK: Theoretical Runtime Analysis for RLS and EA+RL. In: Proceedings of Genetic and Evolutionary Computation Conference Companion, pp. 201–202. ACM (2014)
22. *Watson R. A., Pollack J.B.*: Analysis of recombinative algorithms on a hierarchical building-block problem. In Foundations of Genetic Algorithms (2000)
23. *Molina, D., Herrera, M., Derrac, J., Garcia., S.*: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. (2011)