Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет информационных технологий и программирования
Кафедра компьютерных технологий

Афанасьева Арина Сергеевна

Выбор вспомогательных оптимизируемых величин для ускорения процесса оптимизации с помощью машинного обучения

Научный руководитель: ассистент кафедры КТ М. В. Буздалов

Санкт-Петербург
2012

Содержание

Введение

Существуют различные способы повышения эффективности скалярной оптимизации. Некоторые из них основаны на использовании вспомогательных критериев. Например, задача скалярной оптимизации может быть преобразована в задачу многокритериальной оптимизации путем разработки дополнительных критериев, обладающих определенными заранее заданными свойствами, что позволяет избежать остановки поиска решения в локальном оптимуме [1]. Также в качестве источника вспомогательных критериев может выступать предметная область [2]. В этом случае свойства критериев чаще всего заранее не известны, причем они могут меняться в зависимости от того, на каком этапе находится процесс оптимизации.

В данной работе предлагается метод повышения эффективности скалярной оптимизации с использованием вспомогательных критериев. Предполагается, что набор критериев задан заранее и об их свойствах ничего не известно. Таким образом, возникает задача скалярной оптимизации со вспомогательными критериями. Важно отметить, что задача оптимизации самих вспомогательных критериев не ставится. В то же время, в традиционной теории многокритериальной оптимизации одинаково важны все критерии [3—5]. Существуют также задачи, в которых некоторые критерии представляют меньший интерес, чем остальные, хотя такие критерии по-прежнему оптимизируются [6]. В поставленной задаче должен быть оптимизирован только один *целевой* критерий.

Описанная задача решается с применением эволюционного алгоритма (Evolutionary Algorithm, EA) [7—10], настраиваемого во время выполнения с помощью обучения с подкреплением (Reinforcement Learning, RL) [11—14]. В дальнейшем предлагаемый метод будет называться EA + RL. Таким образом, метод решения описанной задачи основан на настройке эволюционного алгоритма во время выполнения. Вспомогатель-

ные критерии, так же как и целевой критерий, задаются с помощью функций приспособленности. Метод EA + RL позволяет выбирать из заранее подготовленного набора наиболее эффективную ФП для генерации каждого последующего поколения эволюционного алгоритма. В других существующих методах настройки эволюционных алгоритмов обычно настраиваются вещественные параметры фиксированной ФП, причем настройка ФП освещена в литературе в меньшей степени, чем настройка иных параметров ЭА [15—18].

Новизна предлагаемого подхода заключается в использовании обучения с подкреплением для выбора ФП эволюционного алгоритма. Обучение с подкреплением является современной развивающейся технологией, применимость которой в различных областях человеческой деятельности находится в процессе исследования [12, 14]. Насколько известно автору, существует лишь две работы, в которых исследуется возможность использования обучения с подкреплением для настройки эволюционного алгоритма [18, 19]. В обеих работах рассматривается настройка вещественных параметров, таких как, например, вероятность мутации или размер поколения. Данная работа вносит вклад в исследование применимости обучения с подкреплением к настройке ФП.

Предлагаемый метод применен к решению различных модельных задач. В частности, решалась задача оптимизации функции H-IFF [20], применяемая для тестирования генетических алгоритмов, а также разработки методов предотвращения остановки процесса оптимизации в локальном оптимуме, основанных на сведении однокритериальной оптимизации к многокритериальной. При решении этой задачи применение разработанного метода к настройке $1 + 1$ эволюционной стратегии [7, 8, 10] позволило стабильно выращивать лучшие возможные особи за фиксированное число поколений, в то время как стратегия без настройки не справилась с этой задачей. Также в работе представлены результаты решения модельных задач, иллюстрирующие такие особенности метода, как динамическое

переключение ФП, выгодных на различных этапах оптимизации, и устойчивость метода в случае наличия в наборе вспомогательной ФП, использование которой ведет к быстрому ухудшению целевой ФП.

Предлагаемый метод не уступает алгоритму многокритериальной оптимизации PESA [21], показывающему лучшие результаты решения многокритериального варианта задачи H-IFF [1]. Более того, при решении модификации этой задачи, в которой присутствует не коррелирующий с целевой ФП критерий, разработанный метод EA + RL позволяет выращивать лучшие возможные особи в отличие от алгоритма многокритериальной оптимизации PESA-II, входящего в число самых эффективных методов многокритериальной оптимизации, известных на данный момент [22, 23]. Это может быть объяснено тем, что обучение позволяет не учитывать неэффективные критерии, в то время как алгоритмы многокритериальной оптимизации стремятся оптимизировать все критерии. Таким образом, метод EA + RL применим для более широкого множества вспомогательных критериев, чем методы многокритериальной оптимизации, используемые для повышения эффективности скалярной оптимизации.

Глава 1. Обзор предметной области

1.1. Теория оптимизации

Рассмотрим различные способы постановки задачи оптимизации [3—5, 24].

1.1.1. Задача скалярной оптимизации

Задача скалярной (однокритериальной) оптимизации может быть определена как задача максимизации целевой функции $f(x) : X \to \mathbb{R}$, где $X \subseteq W$ — множество допустимых решений, содержащееся в пространстве решений W. Множество X задается либо перечислением допустимых решений (в случае их конечного числа), либо некоторым набором ограничений. Таким образом, задача скалярной оптимизации имеет вид $f(x) \to \max, x \in X$. Решением скалярной задачи оптимизации является $x^* \in X : f(x^*) \geq f(x)$ для всех $x \in X$.

1.1.2. Многокритериальная задача оптимизации

Рассмотрим совокупность критериев $\varphi = (\varphi_1, \varphi_2, .., \varphi_m)$, заданных на пространстве W. Отдельный критерий называют частным критерием выбора, а множества его возможных значений — шкалой критерия. Рассматриваемая совокупность задает отображение $\varphi : W \to W'$, где W' — критериальное пространство, включающее в себя прямое произведение шкал критериев. Решение многокритериальной задачи оптимизации можно свести к поиску наиболее предпочтительной точки множества достижимых значений критериев $Y = \varphi(X) \in W'$. Формализация понятия наиболее предпочтительной точки может быть осуществлена, например, путем введения понятия доминирования по Парето. Заметим, что при решении многокритериальной задачи оптимизации в равной степени важны значения всех частных критериев.

1.1.3. Задача оптимизации со вспомогательными критериями

На практике возникает необходимость решать модифицированную скалярную задачу оптимизации, предполагающую наличие вспомогательных критериев. Она отличается от многокритериальной задачи тем, что целью ее решения по-прежнему является максимизация целевой функции, а вспомогательные критерии используются лишь для повышения эффективности процесса максимизации. В таких задачах целевая функция некоторым образом зависит от вспомогательных критериев, поэтому в некоторых случаях вместо максимизации целевой функции оказывается выгодным оптимизировать эти критерии.

Рассмотрим пример, в котором оптимизация по вспомогательным критериям оказывается выгодной. В работе [2] возникает необходимость выбора между различными функциями приспособленности особей, представляющих тесты для проверки олимпиадных задач. В этом случае целевой функцией приспособленности, которую необходимо максимизировать, является время работы тестируемого решения.

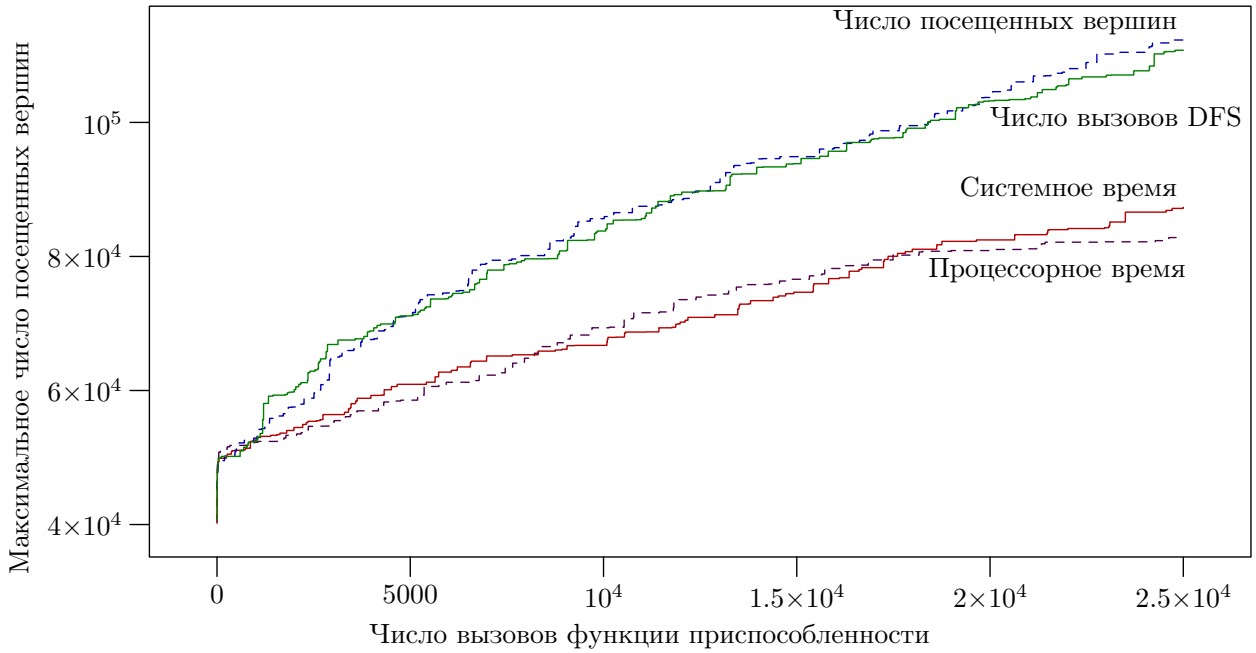

Рис. 1.1: Сравнение функций приспособленности

Оказывается, что время не является оптимальной функцией при-

способленности. Предлагается использовать функции приспособленности, вычисляющие некий показатель сложности теста (например, число итераций алгоритма при прохождении теста). При оптимизации по этим вспомогательным функциям удается получить тесты, время работы на которых на несколько порядков выше, чем значения, получаемые при оптимизации по самой целевой функции приспособленности. Подобное поведение проиллюстрировано на рис. 1.1. Существует несколько вариантов таких вспомогательных функций, из которых также следует выбрать наиболее подходящий. Отметим новизну постановки задачи скалярной оптимизации, в которой фигурируют вспомогательные критерии.

Выбор между функциями приспособленности осуществляется вручную, что неэффективно, так как каждый запуск эволюционного алгоритма требует много ресурсов. Возникает задача автоматизации выбора функции приспособленности в течение одного запуска алгоритма оптимизации.

1.2. Автоматическая настройка эволюционных алгоритмов

В ходе работы предполагается автоматизировать выбор наиболее подходящей функции приспособленности. Рассмотрим, какие еще задачи автоматизации настройки эволюционных алгоритмов решаются. К этим задачам относятся:

- автоматическая настройка параметров (вероятностей мутации и кроссовера, размера поколения);
- разработка адаптивных эволюционных операторов, таких как операторы мутации и кроссовера;
- разработка адаптивных способов представления особей;
- адаптивная настройка функций приспособленности.

Наиболее широко изучена автоматическая настройка параметров. Предлагаемая тема может быть отнесена к адаптивной настройке функций при-

способленности, разработки по которой встречаются в литературе значительно реже. Предлагаемая постановка задачи отличается новизной, так как в других подобных исследованиях акцент делался на настройке некоторой фиксированной функции [17]. В разрабатываемом методе предлагается выбирать между качественно разными функциями.

1.3. Машинное обучение

Методы машинного обучения [25] предоставляют обобщенные подходы к решению задач, что позволяет применять их во многих областях: от разработки оптимальных стратегий игр и управления техническими процессами до решения задач распознавания и классификации. Рассмотрим основные подходы, применяющиеся в машинном обучении, с целью анализа их применимости к выбору вспомогательных функций приспособленности генетического алгоритма.

1.3.1. Деревья принятия решений и обучение концептам

Деревья принятия решений используются для классификации данных. Они позволяют аппроксимировать заданную булевскую функцию. Обучение ведется на конечных наборах дискретных атрибутов, для каждого из которых задано значение аппроксимируемой функции. Также для классификации данных в схожих условиях применяется обучение концептам [25]. Требование наличия готового набора тестовых примеров, их дискретный характер, а также сама цель подобных методов, заключающаяся в классификации данных, затрудняют их применение к решению поставленной задачи.

1.3.2. Нейронные сети

Помимо аппроксимации булевских функций и решения задач классификации, некоторые виды нейронных сетей позволяют аппроксимировать нелинейные функции с вещественными значениями. Большинство мо-

дификаций нейронных сетей предполагает наличие заранее подготовленного обучающего набора, хотя существуют подходы, позволяющие сделать обучение нейронной сети инкрементальным [26]. Была предпринята попытка аппроксимации зависимости целевой функции приспособленности от дополнительных функций с помощью многослойной нейронной сети, составленной из нелинейных перцептронов и использующей алгоритм обратного распространения ошибки [11, 27]. Также применялась реализация нейронной сети из библиотеки алгоритмов машинного обучения *Encog* [28], в которой использовался алгоритм *Quickprop* [29]. Эти нейронные сети показывали хорошие результаты аппроксимации несложных вещественных функций, но для решения поставленной задачи их точности не хватило. Однако дальнейшее исследование возможности использования нейронных сетей, особенно их инкрементальных модификаций, для выбора функций приспособленности кажется целесообразным.

1.3.3. Байесовское обучение

К методом байесовского обучения относятся, например, классификатор Гиббса и наивный байесовский классификатор [25]. Они позволяют решать задачи классификации с большим количеством параметров. Обучаются на заранее подготовленных наборах тестовых примеров. Применение подобных методов для решения поставленной задачи затруднительно, так как ее сведение к задаче классификации неочевидно.

1.3.4. Алгоритмы кластеризации

Задача кластеризации [11] возникает в анализе данных, распознавании образов, а также в биоинформатике, например, при выделении семейств генов, отвечающих за то или иное биологическое свойство. Алгоритмы кластеризации позволяют разделить объекты одной природы на несколько групп таким образом, чтобы объекты каждой группы были близки между собой, а объекты из разных групп существенно различались.

Поставленная задача не относится к задачам кластеризации.

1.4. Обучение с подкреплением

Большинство алгоритмов обучения с подкреплением [13] не требует наличия заранее подготовленного набора тестовых примеров [14]. В алгоритмах этого типа обучение происходит одновременно с применением накопленного опыта, что позволяет выбирать и применять оптимальные ФП в течение одного запуска эволюционного алгоритма. Таким образом, обучение с подкреплением было выбрано в качестве используемого метода. Отметим новизну применения обучения с подкреплением к оптимизации эволюционных алгоритмов.

Обучение с подкреплением решает задачу разработки стратегии поведения в интерактивной среде. Агент применяет действие к среде, после чего получает вещественнозначное вознаграждение и некоторое представление текущего состояния среды. Процесс обучения представляет собой последовательность таких взаимодействий, будем называть их шагами алгоритма обучения. Задачей агента является выработка стратегии поведения, приводящей к максимизации суммарного вознаграждения. Таким образом, для постановки *задачи обучения с подкреплением* необходимо определить следующие компоненты:

- дискретное множество состояний среды, S;
- дискретное множество действий агента, A;
- множество возможных значений вознаграждения (сигналов);
- правило, по которому паре $(s \in S, a \in A)$ сопоставляется сигнал.

Схема обучения с подкреплением представлена на рис. 1.2. Шаги на схеме нумеруются с помощью переменной t. Отметим, что для ряда алгоритмов обучения с подкреплением доказана сходимость к оптимальной стратегии максимизации вознаграждения [27].

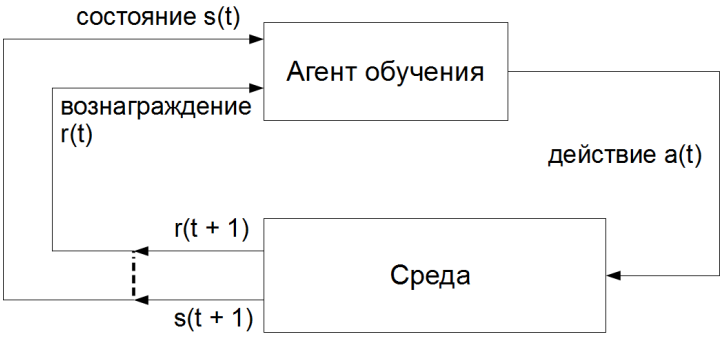

Рис. 1.2: Схема обучения с подкреплением

1.4.1. Оценка оптимальности поведения агента

Для оценки оптимальности стратегии поведения агента применяются различные модели. Будем обозначать вознаграждение, полученное в результате t-ого взаимодействия агента со средой, как r_t.

В *модели конечного горизонта* агент должен оптимизировать ожидаемую награду для следующих h шагов:

$$E(\sum_{t=0}^{h} r_t).$$

Эта модель применима, если число шагов алгоритма заранее известно. В общем случае время жизни агента не определено, поэтому модель конечного горизонта не получила широкого распространения [14].

Модель бесконечного горизонта не ставит ограничений на количество шагов алгоритма. В ней следует максимизировать величину

$$E(\sum_{t=0}^{\infty} \gamma^t r_t).$$

Здесь $0 \leq \gamma < 1$ – дисконтный фактор, позволяющий учитывать, когда было получено вознаграждение. Больший вклад в награду вносят вознаграждения, полученные раньше. Эта модель является одной из самых популярных моделей оценки оптимальности поведения агента в алгоритмах обучения с подкреплением [27].

Также существует *модель среднего вознаграждения*, в которой мак-

симизируется предел ожидания среднего вознаграждения:

$$\lim_{h \to 0} E(\frac{1}{h} \sum_{t=0}^{h} r_t).$$

Эта модель, в отличие от модели бесконечного горизонта, не дает преимущества быстрой прибыли. В некоторых случаях она может быть более эффективна для решения поставленных задач [30]. С другой стороны, существуют работы, в которых утверждается, что результаты, получаемые в модели среднего вознаграждения, могут быть столь же эффективно получены и в модели бесконечного горизонта [31]. В настоящей работе алгоритм обучения R-learning [32], основанный на модели среднего вознаграждения, показал наилучшие результаты в решении модельной задачи H-IFF [1], что описано в разд. 4.4.

1.4.2. Стратегии исследования среды

В обучении с подкреплением исследование среды (exploration) совмещается с применением накопленного опыта, заключающемся в следовании выработанной стратегии (exploitation) [14]. Существуют различные стратегии подобного совмещения. Простейшая *жадная* стратегия состоит в том, чтобы всегда выбирать действие, ожидаемое вознаграждение за которое максимально. Следование этой стратегии может привести к тому, что агент выберет локально оптимальное поведение, недостаточно исследовав среду.

Жадную стратегию можно улучшить, выбирая с некоторой вероятностью ε случайное действие, а с вероятностью $1 - \varepsilon$ — по-прежнему действие, приводящее к максимальному ожидаемому вознаграждению. Такая стратегия носит название *ε-жадной*. Значение ε может уменьшаться со временем, что позволяет постепенно переходить от исследования среды к применению опыта. Подобная тактика наиболее эффективна в том случае, когда свойства среды не меняются.

Также можно отметить *исследование по Больцману* [27], при кото-

ром вероятность выбора действия a на текущем шаге определяется следующей формулой:

$$p(a) = \frac{e^{ER(a)/T}}{\sum_{a' \in A} e^{ER(a')/T}},$$

где ER — ожидаемое вознаграждение, T — температура, убывающий со временем параметр. Эта стратегия также наиболее эффективна в случае постоянной среды.

В некоторых алгоритмах обучения с подкреплением применяются особые эвристики, определяющие выбор действия. Подробнее с различными стратегиями исследования среды можно ознакомиться в [14].

1.4.3. Марковский процесс принятия решений

В общем случае действие агента определяет не только получаемое вознаграждение, но и состояние, в которое переходит среда после воздействия. Таким образом, возникает понятие отложенного вознаграждения. Возможна стратегия, в соответствии с которой агент должен в течение некоторого времени получать незначительное вознаграждение, чтобы в итоге достичь состояния среды, которому соответствует большое вознаграждение. Задачи обучения с подкреплением в случае отложенного вознаграждения могут быть описаны как *Марковский процесс принятия решений* (МППР) [13, 27]. МППР определяют следующие компоненты:

- дискретное множество состояний среды S;
- дискретное множество действий агента A;
- функция вознаграждения $R : S \times A \to \mathbb{R}$
- функция переходов $T : S \times A \times S \to \mathbb{R}$ такая, что вероятность перехода из состояния s в состояние s' в результате воздействия a задается значением $T(s, a, s')$.

1.4.4. Q-обучение

Опишем Q-обучение — алгоритм обучения с подкреплением, не строящий модель среды и относящийся к семейству алгоритмов итерации по

значениям [13, 27, 33]. В ходе работы этого алгоритма аппроксимируется функция $Q(s, a), s \in S, a \in A$ — ожидаемое оптимальное вознаграждение за действие a в состоянии s. В качестве стратегии исследования среды будем использовать ε-жадную стратегию. В листинге 1 приведен псевдокод алгоритма.

Листинг 1 Алгоритм Q-обучения с ε-жадной стратегией исследования среды

Вход: ε — вероятность выбора случайного действия; α — скорость обучения; γ — дисконтный фактор.

Инициализировать $Q(s, a)$ для всех $s \in S$, $a \in A$
while (не достигнуто условие останова) **do**
  Получить состояние среды s
  $p \leftarrow$ случайное вещественное число $\in [0, 1]$
  if $(p \leq \varepsilon)$ **then**
    $a \leftarrow \arg\max_a Q(s, a)$
  else
    $a \leftarrow$ случайное действие $\in A$
  end if
  Применить действие a к среде
  Получить от среды награду r и состояние s'
  $Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$
end while

Отметим, что существует алгоритм отложенного Q-обучения, Delayed Q-learning, для которого доказана более высокая скорость сходимости [34]. Этот алгоритм, как и алгоритм Q-обучения, был реализован в рамках настоящей работы.

Также в данной работе использовался алгоритм Dyna [13], относящийся к классу алгоритмов, строящих модель среды. Такие алгоритмы аппроксимируют и используют для определения стратегии поведения функцию вознаграждения R и функцию переходов T. В применении к рассматривавшимся модельным задачам алгоритм Dyna не показал высоких результатов.

1.5. Выводы по главе 1

Описаны задачи скалярной и многокритериальной оптимизации. Поставлена задача скалярной оптимизации со вспомогательными критериями. Она возникает на практике при использовании эволюционных алгоритмов и наличии набора вспомогательных функций приспособленности.

Отмечена новизна постановки задачи.

Обоснована необходимость разработки метода автоматизации выбора вспомогательных функций приспособленности. Рассмотрены другие подходы к автоматизации эволюционных алгоритмов. Рассмотренные подходы не позволяют решить поставленную задачу.

Обоснован выбор методов машинного обучения в качестве основы разрабатываемого метода. Произведен обзор методов машинного обучения. Обоснована применимость обучения с подкреплением к решению поставленной задачи. Отмечена новизна применения обучения с подкреплением к оптимизации производительности эволюционных алгоритмов.

Кратко описана теория обучения с подкреплением, которое применяется для разработки стратегии поведения агента в интерактивной среде. Формализована задача обучения с подкреплением. Определены способы оценки оптимальности поведения агента. Перечислены наиболее часто использующиеся стратегии совмещения исследования среды с применением накопленного опыта. Дано понятие Марковского процесса принятия решений, которому соответствует типичная задача обучения с подкреплением. Описано Q-обучение — один из базовых алгоритмов обучения с подкреплением, не строящих модель среды.

Глава 2. Постановка задачи и описание метода ее решения

В данной главе обосновывается необходимость улучшения существующих подходов, описывается новый метод . TODO

2.1. Задача

SGM является оптимальным компромиссом качества и скорости по построению карт глубин на текущий момент. Для выполнения всех шагов алгоритма Semi-Global matching требуется $O(W \cdot H \cdot D)$ времени и $O(W * \cdot H * \cdot * D)$ памяти, где W - ширина изображения, H - высота изображения, D - максимальная диспаратность. Ввиду того, что D зависит от размера изображения, мы получаем нелинейное возрастание времени работы алгоритма от размера изображения. Несмотря на огромную разница в производительности относительно глобальных алгоритмов, время работы SGM на больших изображениях слишком велико и не сопоставимо со временем работы локальных алгоритмов. Качество, полученных результатов, близко к качеству глобальных алгоритмов. Поэтому попробуем изменить метод SGM для повышения скорости работы без потери качества.

2.2. Метод с использованием приближенных вычислений.

Качество любых алгоритмов построения карт глубин измеряется с помощью специальных пар картинок, для которых вручную созданы карты. Алгоритму подается такая пара изображений, затем по полученной с помощью него карте делается сравнение с эталонной. Для разных задач требуется разная точность диспаратностей, поэтому при измерении качества диспаратность пикселя считается верной, если она отличается не более, чем на определенную велечину. Алгоритмы могут давать хорошие ре-

зультаты с большой погрешностью и плохие с маленькой.

Предлагаемый метод получает на вход пару изображений, карту глубины с большой погрешностью и возвращает карту глубины с меньшей. Приближенную карту глубины мы можем посчитать с помощью быстрого, например локального, алгоритма, который строит карту с хорошим качеством при большой погрешности.

Предлагаемый метод EA + RL основан на применении эволюционного алгоритма, настраиваемого во время выполнения с помощью обучения с подкреплением. Схема предлагаемого метода представлена на рис. 2.1. Красным цветом выделены особенности данной схемы, отличающие ее от схемы обучения с подкреплением, изображенной на рис. 1.2. Агент взаимодействует со средой, ассоциируемой с ЭА. Он передает ФП, которая должна быть использована в следующем поколении с номером $t + 1$. Затем агент получает вознаграждение, зависящее от разности значений целевой ФП, вычисленной для лучших особей двух последних сгенерированных поколений. Он также получает состояние среды, зависящее от свойств текущего поколения. Генерируется новое поколение ЭА и процесс повторяется.

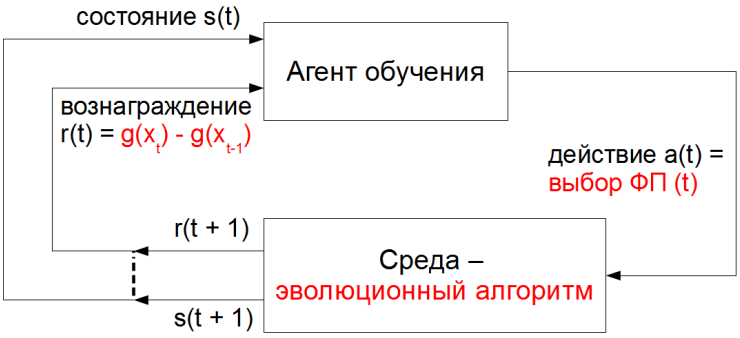

Рис. 2.1: Схема предлагаемого метода

Агент использует один из алгоритмов обучения с подкреплением, максимизирующий суммарное вознаграждение. Чем выше вознаграждение, тем, следовательно, значительнее был рост целевой ФП. Таким образом, оптимальный выбор ФП алгоритмом обучения ведет к лучшей производительности алгоритма оптимизации.

В следующих разделах приводится формальное представление за-

дачи ускорения ЭА как задачи обучения с подкреплением, а также более детально описывается предлагаемый метод.

2.2.1. Задача обучения с подкреплением

Опишем имеющуюся задачу как задачу обучения с подкреплением (1.4). Для этого достаточно задать множество действий агента A, способ определения состояний среды $s \in S$ и функцию вознаграждения $R : S \times A \to X \subseteq \mathbb{R}$.

Будем обозначать особи, выращиваемые ЭА, как x. Пусть G_i — i-ое поколение. Множество действий A соответствует множеству функций приспособленности, состоящему из g — целевой ФП и элементов множества H — вспомогательных ФП:

$$A = H \cup \{g\}. \tag{2.1}$$

Применение действия реализуется как выбор некоторой ФП $f_i \in A$ в качестве функции, используемой для оценки приспособленности особей ЭА, и формирования поколения G_i.

Введем обозначение для *лучшей особи* поколения G_i, обладающей максимальным значением выбранной для этого поколения ФП f_i:

$$z_i = \arg\max_{x \in G_i} f_i(x).$$

Также введем обозначение для нормированной разности значений некоторой ФП, вычисленной на лучших особях двух последовательных поколений:

$$\Delta(f, i) = \frac{f(z_i) - f(z_{i-1})}{f(z_i)}, f \in A.$$

Каждому поколению ЭА поставим в соответствие состояние среды. Состояние s_i, соответствующее поколению G_i, представляет собой вектор функций приспособленности $f \in A$, упорядоченный по убыванию значений нормированных разностей $\Delta(f, i)$:

$$s_i = \langle f_1, f_2, \dots f_{k+1} \rangle : \Delta(f_1, i) \geq \Delta(f_2, i) \geq \dots \Delta(f_{k+1}, i). \tag{2.2}$$

В том случае, если для некоторых f_a, f_b значение $\Delta(f_a, i)$ совпадает со значением $\Delta(f_b, i)$, функции f_a, f_b располагаются в заранее установленном порядке. Например, пусть число вспомогательных ФП $k = 2$ и в некотром поколении G_i выполняется неравенство $\Delta(h_2, i) = \Delta(g, i) > \Delta(h_1, i)$. Тогда соответствующее состояние среды может иметь вид $s_i = \langle h_2, g, h_1 \rangle$ или $s_i = \langle g, h_2, h_1 \rangle$ в зависимости от начальной договоренности.

В заключение определим функцию вознаграждения $R : S \times A \to \{0, \frac{1}{2}, 1\}$, которая вычисляется после выбора действия f_i в состоянии s_{i-1} и генерации поколения G_i:

$$R(s_{i-1}, f_i) = \begin{cases} 1 & \text{если } g(z_i) - g(z_{i-1}) > 0 \\ \frac{1}{2} & \text{если } g(z_i) - g(z_{i-1}) = 0 \\ 0 & \text{если } g(z_i) - g(z_{i-1}) < 0 \end{cases}. \tag{2.3}$$

Согласно (2.3), вознаграждение зависит от разности значений целевой ФП, посчитанной на лучших особях двух последовательных поколений. Значение вознаграждения наиболее высоко, когда целевая ФП растет. Напомним, что в обучении с подкреплением целью агента является максимизация суммарной награды, причем для ряда алгоритмов обучения с подкреплением доказана их сходимость к оптимальной стратегии поведения. Следовательно, задача обучения с подкреплением определена таким образом, что оптимальные действия агента будут приводить к максимизации прироста целевой ФП.

2.2.2. Описание алгоритма EA + RL

Предлагаемый метод позволяет управлять ходом выполнения эволюционного алгоритма путем назначения текущей ФП для каждого вновь сгенерированного поколения. Можно выделить две независимые сущности, составляющие основу метода: модуль обучения и эволюционный алгоритм. Будем называть эволюционный алгоритм средой обучения. Модулю обучения может быть передана награда и состояние среды. Он способен сообщать

действие, которое необходимо применить к среде. В листинге 2 представлен псевдокод разработанного алгоритма. Модуль обучения может быть

Листинг 2 Метод оптимизации работы ЭА с помощью обучения

```
1: Инициализировать модуль обучения
2: Установить номер текущего поколения: i ← 0
3: Сгенерировать начальное поколение G_0
4: while (условие останова ЭА не выполнено) do
5:    Вычислить состояние s_i и передать его модулю обучения
6:    Получить ФП для следующего поколения f_{i+1} из модуля обучения
7:    Сгенерировать следующее поколение G_{i+1}
8:    Вычислить вознаграждение: r ← R(s_i, f_{i+1}) и передать его модулю обучения
9:    Обновить номер текущего поколения: i ← i + 1
10: end while
```

реализован на основе произвольного алгоритма обучения с подкреплением и взаимодействовать с произвольным эволюционным алгоритмом. В ходе выполнения работы было реализовано четыре различных алгоритма обучения: Q-learning [13], Delayed Q-learning [34], Dyna [14, 27] и R-learning [32]. Для обозначения различных реализаций предлагаемого метода EA + RL будем заменять в названии метода "EA"на название конкретного эволюционного алгоритма, "RL— на название алгоритма обучения с подкреплением. Например, если с помощью предлагаемого метода реализуется выбор ФП генетического алгоритма (ГА) с помощью алгоритма обучения Q-learning, то соответствующая реализация метода будет называться *ГА + Q-learning*.

В качестве эволюционного алгоритма применялся генетический алгоритм [9] с различными видами мутации и кроссовера. Также использовались различные виды эволюционных стратегий [7]. Эволюционные алгоритмы были реализованы на языке программирования *Java* с помощью библиотеки эволюционных алгоритмов *Watchmaker* [35].

2.3. Выводы по главе 2

Формализована цель исследований: разработка метода ускорения получения особей эволюционного алгоритма с высокими значениями целевой ФП при наличии конечного набора вспомогательных ФП. Описаны требования, предъявляемые к разрабатываемому методу. Приведено све-

дение поставленной задачи к задаче обучения с подкреплением. Описан предлагаемый метод, основанный на назначении текущей ФП для каждого вновь сгенерированного поколения эволюционного алгоритма.

Глава 3. Демонстрация особенностей метода

В данной главе экспериментальным способом на примере решения модельных задач показаны некоторые особенности разработанного метода. А именно, способность метода выбирать наиболее выгодные ФП на различных этапах оптимизации, тем самым повышая производительность ЭА, а также способность игнорировать такие ФП, применение которых приводит к ухудшению целевой ФП. Таким образом, выполняются требования, предъявляемые к методу в разд. **??**.

3.1. Модельная задача min-max

3.1.1. Постановка задачи

Рассмотрим модельную задачу, на примере которой экспериментально проверена способность метода выбирать наиболее эффективные ФП на различных этапах оптимизации. Особь представлена битовой строкой длиной n, x бит которой равны единице. Определим целевую ФП как

$$g(x) = \left\lfloor \frac{x}{d} \right\rfloor. \tag{3.1}$$

Вспомогательные функции имеют вид $h_1(x) = \min(x, p)$ и $h_2(x) = \max(x, p)$, где p — положительное целое число. Будем называть p *точкой переключения*. Графики описанных функций представлены на рис. 3.1.

Особенностью этой задачи является наличие точки переключения, позволяющее проверить способность метода динамически выбирать наиболее выгодную на данном этапе оптимизации функцию приспособленности. Для особей, число единиц в представлении которых меньше, чем точка переключения p, наиболее эффективно использовать вспомогательную функцию h_1 в качестве текущей. Для остальных особей должна быть использована функция h_2. Ожидается, что в таком случае особи с более высокими

значениями целевой функции g будут выращены быстрее, чем в случае использования самой целевой функции в качестве текущей.

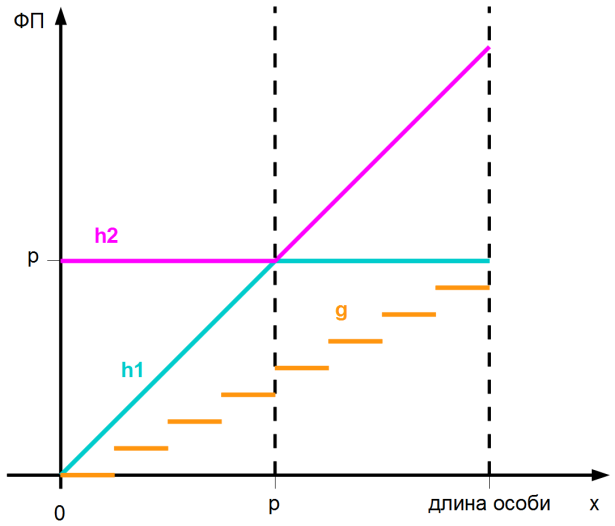

Рис. 3.1: Функции приспособленности в модельной задаче

3.1.2. Описание и результаты эксперимента

Был проведен ряд экспериментов, в ходе которых к решению модельной задачи применялись различные алгоритмы обучения с подкреплением, оптимизирующие генетический алгоритм (ГА). Значения параметров ГА, использованные в экспериментах, представлены в табл. 3.1. Было перебрано более тысячи различных комбинаций значений параметров обучения для каждого использованного алгоритма. Каждый алгоритм запускался 50 раз, и результаты, полученные с его помощью, усреднялись. Следует отметить, что наиболее выгодным значением дисконтного фактора (1.4.1) оказался ноль. Это указывает на то, что в данной задаче информации о текущем состоянии среды достаточно для обучения и учет предыдущего опыта не требуется.

Наилучшие результаты удалось получить, используя алгоритм обучения Q-learning с ε-жадной стратегией (1.4.2). На рис. 3.2 представлены графики зависимости целевой ФП от номера поколения в случае использования метода, основанного на обучении, и обычного ГА. Можно видеть, что применение обучения позволяет гарантированно вырастить идеальную

Таблица 3.1: Значения параметров, использованные в ходе эксперимента

Параметр	Значение
Число запусков алгоритма	50
Число особей в поколении	100
Число поколений	400
Коэффициент элитизма	5
Вероятность кроссовера	0.7
Вероятность мутации	0.003
Длина особи	400
Точка переключения	266
Делитель d	10

особь менее, чем за 250 поколений. Использование обычного ГА, положенного в основу метода с обучением, не приводит к стабильному получению идеальной особи в пределах использованного количества поколений, что проиллюстрировано на рис. 3.3. Также на рис. 3.2 находится диаграмма, отражающая число раз, когда была выбрана та или иная ФП. Отметим, что разработанный алгоритм успешно справился с выбором функций h_1 и h_2 на соответствующих этапах оптимизации.

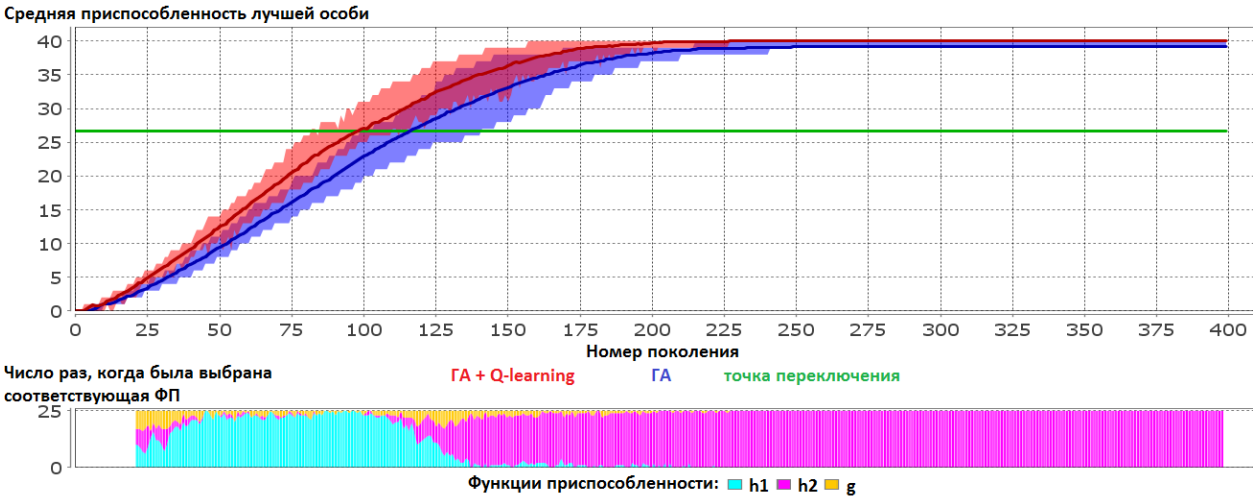

Рис. 3.2: Усредненные результаты решения модельной задачи min-max

3.2. Задача Royal Roads с мешающей ФП

Рассмотрим модельную задачу, в которой оптимизация по единственной вспомогательной ФП приводит к быстрой убыли целевой. Экспериментально покажем, что определенные виды обучения способны справляться с решением такой задачи, выбирая целевую ФП. В качестве целевой

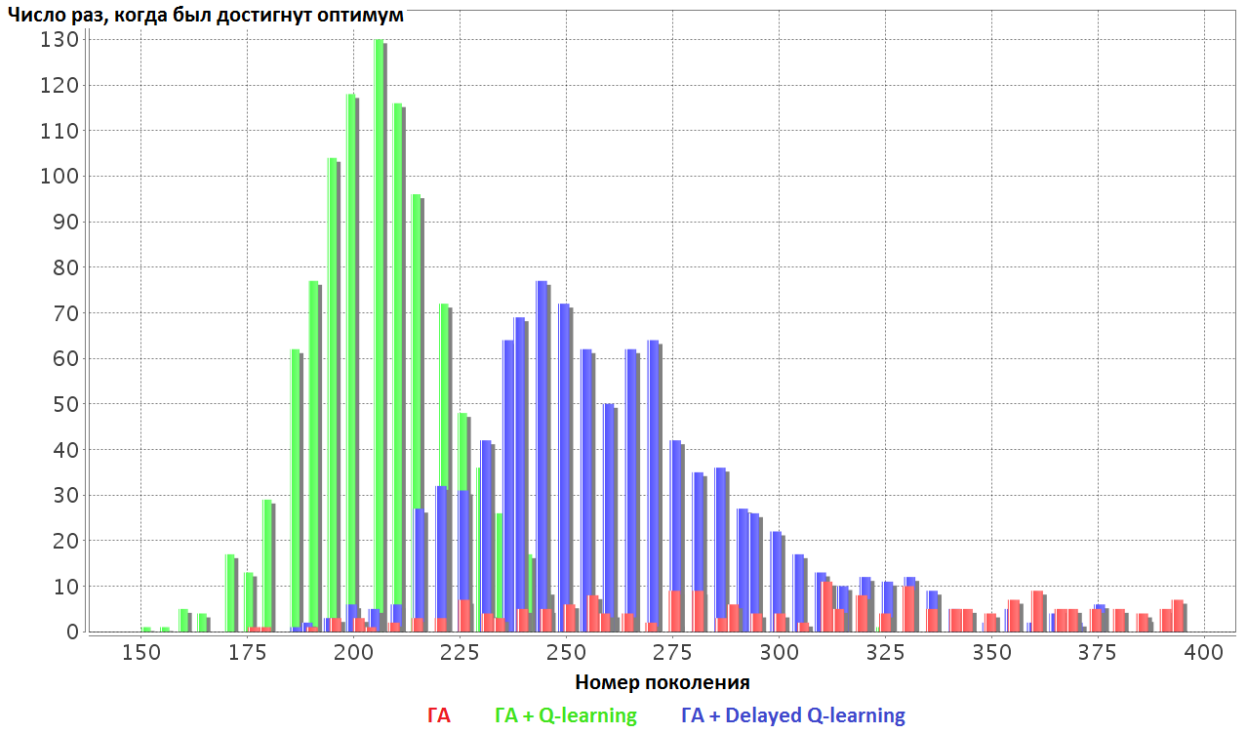

Рис. 3.3: Распределение числа раз, когда было получено оптимальное решение задачи min-max, в зависимости от номера поколения

ФП возьмем функцию Royal Roads [9]. Пусть длина особи равна 64 битам. Выберем длину блока $b = 8$. Наличие очередного блока, заполненного b единицами, увеличивает ФП на b. В качестве вспомогательной ФП будем использовать функцию, значение которой равно числу нулей, содержащихся в особи.

В табл. 3.2 приведены результаты эксперимента по решению поставленной задачи. Усреднение производилось за 50 запусков каждого алгоритма. Выполнение алгоритма останавливалось при нахождении оптимума, либо при превышении максимального числа вычислений ФП, равного 500000. Число шагов в запусках, в которых не удалось получить оптимум, условно обозначено как бесконечность.

Можно видеть, что применение стратегии исследования среды по Больцману (1.4.2) позволяет получить такую же производительность, что и без мешающей ФП. Использование алгоритма Delayed Q-learning, реализующего собственную стратегию исследования среды, также приводит к высоким результатам. Алгоритмы, использующие ε-жадную стратегию

исследования среды значительно проигрывают упомянутым алгоритмам. Это можно объяснить тем, что для ε-жадной стратегии характерно существование постоянной вероятности ε, с которой выбирается случайное действие. Значит, в алгоритмах, использующих эту стратегию, всегда присутствует вероятность выбора мешающей ФП, равная $0,5\varepsilon$, что приводит к ухудшению целевой ФП. Заметим также, что установка нулевой вероятности выбора случайного действия в общем случае неэффективна (что подтверждается экспериментальными данными), так как подобный подход не позволяет агенту исследовать среду.

Таблица 3.2: Результаты решения задачи Royal Roads

Алгоритм	% успешных запусков	Шагов в успешных запусках (среднее)	Max шагов	Min шагов	σ
(1+1) ЭС без мешающей ФП	100	6913,28	16033	2439	2925,07
ЭС + Q-learning [13] (больцмановская стратегия)	100	6813,40	17252	2659	2999,60
(1+1) ЭС + Delayed [34]	88	8365,52	∞	3982	3216,13
ЭС + R-learning [32]	100	69881,74	289936	5254	67990,07
ЭС + Dyna [13] (ε-жадная стратегия)	26	5749,62	∞	2654	2056,78
ЭС + Q-learning (ε-жадная стратегия)	24	6964,17	∞	3012	2256,70

3.3. Выводы по главе 3

Описана модельная задача, позволяющая проверить работоспособность предлагаемого метода в случае, когда на различных этапах оптимизации выгодны различные вспомогательные ФП. Приведены результаты экспериментов, показывающие, что использование разработанного метода для решения описанной модельной задачи позволяет в 100% запусков получать лучшую особь за конечное число поколений, в то время как с помощью ГА без обучения подобной производительности достичь не удается. В ходе эксперимента подтверждено, что разработанный метод динамически выбирает функцию приспособленности, наиболее выгодную на данном этапе.

Также описана задача, в которой оптимизация по вспомогательной ФП сильно ухудшает значение целевой ФП. Показано, что использование стратегии исследования среды по Больцману, а также алгоритма Delayed Q-learning, обладающего собственной стратегией исследования среды, позволяет получать такую же производительность, как в случае отсутствия мешающей ФП. Иначе говоря, обучение способно успешно игнорировать неэффективные вспомогательные ФП.

Глава 4. Сравнение с методами многокритериальной оптимизации

Существуют методы, в которых задача скалярной оптимизации сводится к задаче многокритериальной оптимизации с целью устранения остановки поиска наиболее приспособленных особей эволюционного алгоритма в локальном оптимуме, а также поддержания разнообразия особей в популяции [1, 36, 37]. Сведение к многокритериальной задаче предполагает разработку специальных дополнительных критериев, гарантированно коррелирующих с целевой ФП. В задаче, поставленной в настоящей работе, заранее не известно, коррелируют ли вспомогательные функции с целевой. Методы многокритериальной оптимизации не эффективны в случае, когда среди вспомогательных функций есть функции, не коррелирующие с целевой.

Разработанный метод может применяться вместо методов многокритериальной оптимизации в случае, когда они используются для решения скалярной задачи оптимизации. Для этого в качестве вспомогательных ФП следует использовать введенные на этапе сведения к многокритериальности дополнительные критерии. В данном разделе приведено экспериментальное сравнение разработанного метода с методами многокритериальной оптимизации на примере решения модельной задачи. Показано, что предлагаемый метод позволяет не учитывать невыгодные критерии, что приводит к большей производительности разработанного метода по сравнению с методами многокритериальной оптимизации.

4.1. Модельная задача H-IFF

Рассмотрим функцию, называемую Hierarchical-if-and-only-if function, H-IFF [1, 20]. Максимизация этой функции, применяемой в качестве ФП, является модельной задачей, используемой для тестирования генетических алгоритмов. Пространство поиска решений этой задачи

состоит из битовых строк фиксированной длины l. Функция принимает на вход битовую строку $B = b_1 b_2 \dots b_l$ и интерпретирует ее как двоичное дерево, узлами и листьями которого являются строковые блоки. Вершине дерева соответствует исходная строка B, потомки каждого узла являются левой B_L и правой B_R половинами родительского блока. ФП f вычисляется как сумма длин однородных блоков, состоящих только из нулей или только из единиц. Соответствующая рекурсивная формула приведена в (4.1). Обратите внимание, что существует два возможных оптимальных решения задачи. Один оптимум представляет собой строку из нулевых битов, другой оптимум имеет вид строки, состоящей из единичных битов.

$$f(B) = \begin{cases} 1 & \text{если } |B| = 1, \text{ иначе} \\ |B| + f(B_L) + f(B_R) & \text{если } \forall i\{b_i = 0\} \text{ или } \forall i\{b_i = 1\} \\ f(B_L) + f(B_R) & \text{в остальных случаях} \end{cases} \tag{4.1}$$

Различные методы скалярной оптимизации останавливаются в локальном оптимуме, решая эту задачу. Однако задача может быть решена с использованием методов многокритериальной оптимизации после предварительного введения нескольких критериев [1]. Эти методы используют два критерия f_0 и f_1, которые учитывают только те блоки, которые состоят из нулей, или только те блоки, которые состоят из единиц, соответственно (4.2). Таким образом, получаем модифицированную задачу, называемую MH-IFF.

$$f_n(B) = \begin{cases} 0 & \text{если } |B| = 1 \text{ и } b_1 \neq n, \text{ иначе} \\ 1 & \text{если } |B| = 1 \text{ и } b_1 = n, \text{ иначе} \\ |B| + f_n(B_L) + f_n(B_R) & \text{if } \forall i\{b_i = n\} \\ f_n(B_L) + f_n(B_R) & \text{в остальных случаях} \end{cases} \tag{4.2}$$

4.2. Решение задачи H-IFF с помощью разработанного метода

Рассмотрим применение разработанного метода EA + RL к решению задачи H-IFF. Представим эту задачу как задачу скалярной оптимизации со вспомогательными критериями (H-IFFA).

Будем использовать функцию f (4.1) в качестве целевого критерия:

$$g = f : W \to \mathbb{R}, W = \{B : |B| \leq l\}.$$

Функции f_0 и f_1 (4.2) будем использовать в качестве вспомогательных:

$$H = \{f_0, f_1\}, f_i : W \to \mathbb{R}.$$

Целью решения задачи является максимизации функции f на множестве всех битовых строк фиксированной длины l:

$$f(x) \to \max_{x \in X}, X = \{B : |B| = l\}$$

Каждое решение поставленной задачи H-IFFA является решением задачи H-IFF и наоборот. Задача H-IFFA может быть решена с помощью метода EA + RL. Для этого достаточно определить задачу обучения с подкреплением в соответствии с формулами (2.1), (2.2), (2.3). Таким образом, получаем новый способ решения задачи H-IFF. Результаты эксперимента по использованию этого способа представлены в следующем разделе.

4.3. Описание эксперимента

В ходе эксперимента были решены задачи H-IFF и H-IFFA с помощью соответствующих методов. Параметры эксперимента соответствовали параметрам, использованным в статье [1], в которой приведены результаты решения задач H-IFF и MH-IFF. Длина особи l составляла 64 бита. Заметим, что максимальное значение приспособленности такой особи составляет 448. Было произведено 30 запусков каждого из использованных алгоритмов. В каждом из запусков было выполнено 500000 вычислений

ФП. Для сбора приведенной далее статистики использовались лучшие особи последнего поколения каждого из запусков.

Задача H-IFF решалась с использованием двух различных типов ЭА, а именно с применением генетического алгоритма (ГА) и эволюционной стратегии (ЭС). Опишем сначала параметры ГА. Использовался одноточечный кроссовер [9], применявшийся с вероятностью 70%. Оператор мутации, использовавшийся в ГА, инвертировал каждый бит каждой особи с вероятностью $2/l = 3.125\%$. В качестве механизма селекции был применен турнирный отбор с вероятностью выбора лучшей особи, равной 90%. Пять лучших особей каждого поколения ГА переносились в следующее поколение согласно стратегии элитизма.

В ЭС использовался механизм селекции $(1 + m), m = 1, 5, 10$. Оператор мутации инвертировал один случайно выбранный бит каждой особи. Эта вариация ЭС подвержена остановке в локальном оптимуме. Одной из целей эксперимента было показать, что разработанный метод EA + RL способен эффективно настраивать такой алгоритм, позволяя находить с его помощью глобальный оптимум.

Задача H-IFFA решалась с помощью разработанного метода, контролировавшего описанные выше ЭА. Значения параметров различных использованных алгоритмов обучения приведены в табл. 4.1. Они настраивались вручную таким образом, чтобы результаты применения этих алгоритмов к решению задачи H-IFFA были наилучшими из полученных. В алгоритмах, не специфицирующих стратегию исследования среды, была применена ε-жадная стратегия (1.4.2).

Также для обеспечения более основательного сравнения был реализован алгоритм многокритериальной оптимизации PESA-II [22]. Он применялся как для решения задачи MH-IFF в классической постановке, так и для решения задачи MH-IFF с добавленной *мешающей* ФП, которая подсчитывала число совпадений с битовой маской длины l, состоявшей из чередующихся нулей и единиц: 1010...10. Можно видеть, что оптимизация по

этой функции не приводит к росту значений целевой функции H-IFF. Задача с мешающей ФП была также решена с применением разработанного метода EA + RL. Целью этой части эксперимента было показать, что применение обучения позволяет не учитывать мешающие ФП и по-прежнему достигать глобального оптимума, в то время как результаты, получаемые с использованием многокритериальной оптимизации, при наличии таких ФП ухудшаются.

Таблица 4.1: Значения параметров алгоритмов обучения

Параметр	Описание	Значение
Q-learning [13]		
α	скорость обучения	0.6
γ	дисконтный фактор	0.1
ε	вероятность исследования	0.01
Delayed Q-learning [34]		
m	период обновления	5
γ	дисконтный фактор	0.1
ϵ	бонусное вознаграждение	0.2
R-learning [32]		
α	скорость обновления вознаграждения ρ	0.5
β	скорость обучения функции R	0.35
ε	вероятность исследования	0.25
Dyna [13]		
k	число обновлений	20
γ	дисконтный фактор	0.1
ε	вероятность исследования	0.01

4.4. Результаты эксперимента

В табл. 4.2 представлены результаты оптимизации (M)H-IFF с помощью различных алгоритмов. Результаты отсортированы по среднему значению целевой ФП лучших особей, полученных в результате 30 запусков соответствующих алгоритмов. Выделенные подсветкой алгоритмы реализованы с помощью предлагаемого метода с использованием различных алгоритмов обучения. Остальные результаты получены авторами статьи, причем алгоритмы PESA [21] и PAES [38] являются алгоритмами многокритериальной оптимизации. Можно видеть, что предлагаемый метод в случае использования алгоритма обучения R-learning [8] позволяет преодолеть проблему остановки в локальном оптимуме столь же эффективно, как

и метод PESA и более эффективно, чем метод PAES.

Таблица 4.2: Результаты решения (M)H-IFF(A), отсортированные по убыванию среднего значения приспособленности особей

Алгоритм	Лучшее	Среднее	σ	% одного оптимума	% двух оптимумов
(1+10) ЭС + R-learning	448	448.00	0.00	100	40
ГА + R-learning	448	448.00	0.00	100	10
PESA [21]	448	448.00	0.00	100	100
ГА + Q-learning	448	435.61	32.94	87	3
ГА + Dyna	448	433.07	38.07	80	0
PAES [38]	448	418.13	50.68	74	43
ГА + Delayed Q-learning	448	397.18	49.16	53	0
ГА + Random-FF-chooser	384	354.67	29.24	0	0
DCGA [39]	448	323.93	26.54	3	0
ГА	384	304.53	27.55	0	0
SHC [1]	336	267.47	29.46	0	0
(1+10) ЭС	228	189.87	17.21	0	0

В табл. 4.3 отдельно рассмотрена оптимизация H-IFF с применением ЭС. Применяемая ЭС устроена таким образом, что решает задачу весьма неэффективно. Ни в одном из запусков ни одной из рассмотренных вариаций ЭС не удалось вырастить особь с максимальной приспособленностью. Однако применение предлагаемого метода позволило вырастить идеальные особи в 73% запусков при настройке 1 + 1 ЭС и в 100% запусков в других рассмотренных случаях.

Таблица 4.3: Результаты решения H-IFF(A) с использованием ЭС и R-learning

Алгоритм	Лучшее	Среднее	σ	% одного оптимума	% двух оптимумов
(1+1) ЭС + R-learning	448	403.49	59.48	73	10
(1+1) ЭС	188	167.07	11.98	0	0
(1+5) ЭС + R-learning	448	448.00	0.00	100	37
(1+5) ЭС	216	179.07	16.99	0	0
(1+10) ЭС + R-learning	448	448.00	0.00	100	40
(1+10) ЭС	228	189.87	17.21	0	0

Заметим, что целью предлагаемого метода является ускорение получения особей с высокими значениями целевой ФП. Приведенные результаты свидетельствуют о том, что метод успешно справляется с этой задачей. Невысокий процент нахождения обоих возможных оптимумов объясняется тем, что предлагаемый метод не проводит многокритериальную оптимизацию. В случае присутствия мешающих ФП подобное поведение является преимуществом разработанного метода, так как позволяет не учитывать такие ФП.

Рассмотрим последнюю часть проведенного эксперимента. В ней показано, что когда среди вспомогательных ФП есть не коррелирующие с целевой, предлагаемый метод будет эффективнее методов многокритериальной оптимизации. В табл. 4.4 представлены результаты решения MH-IFF и MH-IFF с мешающей ФП с помощью алгоритма PESA-II [22] и разработанного метода $(1+5)$-ЭС + R-learning. Можно видеть, что исходную задачу MH-IFF оба алгоритма решают идеально. Однако при наличии мешающей ФП с помощью алгоритма PESA-II не удается найти идеальное решение, в то время как предлагаемый метод позволяет выращивать лучшие возможные особи в 92% запусков.

Таблица 4.4: Сравнение разработанного метода с PESA-II

Алгоритм	Лучшее	Среднее	σ	% одного оптимума	% двух оптимумов
Задача MH-IFF					
(1+5) ЭС + R-learning	448	448.00	0.00	100	37
PESA-II	448	448.00	0.00	100	100
Задача MH-IFF с мешающей ФП					
(1+5) ЭС + R-learning	448	439.45	36.32	92	45
PESA-II	312	277.83	20.07	0	0

4.5. Выводы по главе 4

Проведено сравнение предлагаемого метода с методами, основанными на сведении задач однокритериальной оптимизации к задачам многокритериальной оптимизации. Отмечено, что предлагаемый метод больше подходит для решения поставленной задачи скалярной оптимизации со вспомогательными критериями, свойства которых заранее не известны (**??**), чем многокритериальная оптимизация. Это объясняется тем, что предлагаемый метод не ставит своей целью максимизацию всех вспомогательных критериев, что позволяет использовать его в случае, когда среди них есть критерии, отрицательно коррелирующие с целевой функцией.

Описана постановка модельной задачи H-IFF. Приведены результаты экспериментов, в ходе которых эта задача решается с использованием методов одно- и многокритериальной оптимизации, а также с использова-

нием предлагаемого метода, основанного на различных алгоритмах обучения с подкреплением. Показано, что результаты работы предлагаемого метода совпадают с результатами работы наиболее эффективного метода многокритериальной оптимизации. Также на практике подтверждено, что предлагаемый метод не максимизирует все критерии, что не препятствует ускоренному получению оптимальных значений целевой функции. Показано, что в случае наличия мешающих критериев эффективность разработанного метода выше, чем эффективность метода многокритериальной оптимизации PESA-II.

Эксперимент показал устойчивость метода по отношению к характеру эволюционного алгоритма, к оптимизации которого он применяется. Эволюционные стратегии, показавшие худшие результаты по решению модельной задачи, под управлением обучения позволили стабильно выращивать лучшие возможные особи.

Заключение

В работе предложен метод, основанный на обучении с подкреплением, позволяющий решать скалярную задачу оптимизации с наличием вспомогательных критериев, свойства которых заранее не известны. Метод повышает производительность эволюционного алгоритма, выбирая вспомогательную функцию приспособленности, наиболее выгодную на данном этапе оптимизации. Предлагаемый метод автоматической настройки эволюционных алгоритмов, основанный на обучении с подкреплением, а также сама постановка модифицированной задачи скалярной оптимизации, возникшей в ходе генерации тестов к олимпиадным задачам по программированию, отличаются новизной.

Проведен эксперимент по решению модельной задачи, в которой на различных этапах оптимизации выгодны различные вспомогательные функции приспособленности. Предлагаемый метод успешно справился с задачей динамического выбора наиболее эффективных ФП. Применение метода позволило стабильно выращивать особи с максимально возможным значением ФП за фиксированное число поколений использованного генетического алгоритма. Также в ходе другого эксперимента подтверждено, что предлагаемый метод игнорирует мешающие ФП, оптимизация по которым может приводить к замедлению роста целевой ФП.

Проведено сравнение предлагаемого метода с методами многокритериальной оптимизации. На примере модельной задачи H-IFF экспериментально показано, что разработанный метод позволяет получить такой же результат, какой может быть получен с использованием наиболее эффективного метода многокритериальной оптимизации. В ходе этого эксперимента также установлено, что применение предлагаемого метода к настройке эволюционной стратегии позволяет выращивать за фиксированное число поколений особи с максимальной приспособленностью в 100% запусков большинства рассмотренных вариантов эволюционных стратегий, в то

время как с помощью обычной эволюционной стратегии без настройки не удается вырастить идеальную особь ни в одном из запусков. Показано, что алгоритмы многокритериальной оптимизации, в отличие от предлагаемого метода, не всегда применимы для решения поставленной задачи, так как их целью является оптимизация всех критериев, что затрудняет их использование в случае, когда некоторые вспомогательные функции приспособленности не коррелируют с целевой. В этом случае разрабатываемый метод позволяет получать лучшие возможные особи в 92% запусков, в то время как с помощью алгоритма многокритериальной оптимизации PESA-II ни в одном запуске не удалось вырастить особь с максимальной приспособленностью.

Таким образом, экспериментально подтверждено, что разработанный метод соответствует предъявляемым ему требованиям. При наличии помимо целевой ФП хотя бы одной вспомогательной ФП (в том числе мешающей) эффективность работы эволюционного алгоритма под управлением данного метода не ниже, чем эффективность работы того же эволюционного алгоритма при использовании только целевой ФП. При наличии помимо целевой ФП хотя бы одной вспомогательной ФП, оптимизация по которой приводит к более быстрому росту целевой ФП, чем оптимизация по собственно целевой функции, эффективность работы эволюционного алгоритма под управлением данного метода превышает эффективность работы того же эволюционного алгоритма при работе только с целевой ФП. Метод обладает способностью переключаться с одной функции приспособленности на другую, если в процессе оптимизации первая из них перестает быть эффективной.

По тематике данной работы была опубликована статья в Научно-техническом вестнике информационных технологий, механики и оптики [40]. Материал, посвященный предлагаемому методу, опубликован в тезисах к конференции International Conference on Machine Learning and Applicaltions "ICMLA-2011"[41]. Доклад, сделанный по результатам данной

работы, был признан лучшим научно-исследовательским докладом студента на I Всероссийском Конгрессе Молодых Ученых. Также результаты данной работы были отражены в докладе на Всероссийской научной конференции по проблемам информатики "СПИСОК-2012".

Список литературы

1. *Knowles J. D.*, *Watson R. A.*, *Corne D.* Reducing Local Optima in Single-Objective Problems by Multi-objectivization / Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization. EMO '01. London, UK: Springer-Verlag, 2001. Pp. 269–283.
2. *Буздалов М. В.* Генерация тестов для олимпиадных задач по теории графов с использованием эволюционных алгоритмов. Маг. дисс. СПбГУ ИТМО, 2011.
3. *Лотов А. В.*, *Поспелова И. И.* Многокритериальные задачи принятия решений: Учебное пособие. М.: МАКС Пресс, 2008.
4. *Ehrgott M.* Multicriteria Optimization. Springer, 2005.
5. *Deb K.* Multi-objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, 2001.
6. *Azuma E.*, *Shimada T.*, *Takadama K.*, *Sato H.*, *Hattori K.* The Biased Multi-Objective Optimization using the Reference Point: Toward the industrial logistics network / Proceedings of the Tenth International Conference on Machine Learning and Applications, ICMLA 2011. Vol. 2. Honolulu, HI, USA: IEEE Computer Society, 2011. Pp. 27–30.
7. *Yang X.-S.* Nature-Inspired Metaheuristic Algorithms. Luniver Press, 2010.
8. *Luke S.* Essentials of Metaheuristics. http://cs.gmu.edu/~sean/book/metaheuristics/.
9. *Mitchell M.* An Introduction to Genetic Algorithms. MIT Press. Cambridge. MA, 1996.
10. *Скобцов Ю. А.* Основы эволюционных вычислений. Донецк: ДонНТУ, 2008.
11. *Alpaydin E.* Introduction to Machine Learning (Adaptive Computation and Machine Learning). MIT Press, 2004.
12. *Gosavi A.* Reinforcement Learning: A Tutorial Survey and Recent Advances // INFORMS Journal on Computing. 2009. No.2. Pp. 178–192.
13. *Sutton R. S.*, *Barto A. G.* Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 1998.
14. *Kaelbling L. P.*, *Littman M. L.*, *Moore A. W.* Reinforcement Learning: A Survey // Journal of Artificial Intelligence Research. 1996. Pp. 237–285.
15. *Eiben A. E.*, *Michalewicz Z.*, *Schoenauer M.*, *Smith J. E.* Parameter Control in Evolutionary Algorithms. In: Parameter Setting in Evolutionary Algorithms. 2007. Pp. 19–46.
16. *Herrera F.*, *Lozano M.* Adaptation of Genetic Algorithm Parameters Based on Fuzzy Logic Controllers // Genetic Algorithms and Soft Computing. 1996. Pp. 95–125.
17. *Sakanashi H.*, *Suzuki K.*, *Kakazu Y.* Controlling dynamics of GA through filtered evaluation function / Y. Davidor, H. P. Schwefel, R. Maenner, Eds., Parallel Problem Solving from Nature-PPSN II, International Conf. on Evolutionary Computation. 1994. Pp. 239–248.
18. *Müller S.*, *Schraudolph N. N.*, *Koumoutsakos P. D.* Step Size Adaptation in Evolution Strategies using Reinforcement Learning / Proceedings of the Congress on Evolutionary Computation. IEEE, 2002. Pp. 151–156.
19. *Eiben A. E.*, *Horvath M.*, *Kowalczyk W.*, *Schut M. C.* Reinforcement learning for online control of evolutionary algorithms / Proceedings of the 4th international conference on Engineering self-organising systems ESOA'06. Springer-Verlag, Berlin, Heidelberg, 2006. Pp. 151–160.
20. *Watson R. A.*, *Pollack J. B.* Analysis of recombinative algorithms on a hierarchical building-block problem. In Foundations of Genetic Algorithms (FOGA). 2000.
21. *Corne D. W.*, *Knowles J. D.* The Pareto-envelope based selection algorithm for multiobjective optimization / the Sixth International Conference on Parallel Problem Solving from Nature (PPSN VI). Berlin: Springer-Verlag, 2000. Pp. 839–848.
22. *Corne D. W.*, *Jerram N. R.*, *Knowles J. D.*, *Oates M. J.*, *J M.* PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization / Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001). Morgan Kaufmann Publishers, 2001. Pp. 283–290.

23. *Coello C.*, *Lamont G.*, *Veldhuisen D. V.* Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation Series. Springer, 2007.
24. *Гилл Ф.*, *Мюррей У.*, *Райт М.* Практическая оптимизация. М.: Мир, 1985.
25. *Mitchell T. M.* Machine Learning. McGraw Hill, 1997.
26. *Polikar R.*, *Udpa L.*, *Udpa S. S.*, *Honavar V.* Learn++: An Incremental Learning Algorithm for Supervised Neural Networks //. 2001.
27. *Николенко С. И.*, *Тулупьев А. Л.* Самообучающиеся системы. М., 2009.
28. Encog Machine Learning Framework. http://www.heatonresearch.com/encog.
29. *Fahlman S. E.* An empirical study of learning speed in back-propagation networks. Tech. rep. 1988.
30. *Mahadevan S.* Average Reward Reinforcement Learning: Foundations, Algorithms, and Empirical Results. 1996.
31. *Tsitsiklis J. N.*, *Roy B. V.* On Average Versus Discounted Reward Temporal-Difference Learning / Machine Learning. MIT, 1999. Pp. 179–191.
32. *Schwartz A.* A reinforcement learning method for maximizing undiscounted rewards / Proceedings of the Tenth International Conference on Machine Learning. 1993. Pp. 298–305.
33. *Редько В. Г.*, ред. От моделей поведения к искусственному интеллекту. М.: КомКнига, 2010.
34. *Strehl A. L.*, *Li L.*, *Wiewiora E.*, *Langford J.*, *Littman M. L.* PAC Model-free Reinforcement Learning / Proceedings of the 23rd International Conference on Machine Learning (ICML 2006). 2006. Pp. 881–888.
35. Watchmaker Framework for Evolutionary Computation. http://watchmaker.uncommons.org.
36. *Neumann F.*, *Wegener I.* Can Single-Objective Optimization Profit from Multiobjective Optimization? In: Multiobjective Problem Solving from Nature. Ed. by Joshua Knowles, David Corne, Kalyanmoy Deb, and Deva Raj Chair. Natural Computing Series. Springer Berlin Heidelberg, 2008. Pp. 115–130.
37. *Jensen M. T.* Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation: Evolutionary Computation Combinatorial Optimization // Journal of Mathematical Modelling and Algorithms. 2004. No.4. Pp. 323–347.
38. *Knowles J. D.*, *Corne D. W.* Approximating the nondominated front using the Pareto archived evolution strategy // Evolutionary Computation. 2000. No.8. Pp. 149–172.
39. *Mahfoud S. W.* Niching methods for genetic algorithms. PhD thesis. Urbana, IL, USA: University of Illinois at Urbana-Champaign, 1995.
40. *Афанасьева А. С.*, *Буздалов М. В.* Выбор функции приспособленности особей генетического алгоритма с помощью обучения с подкреплением // Научно-технический вестник информационных технологий, механики и оптики. 2012. №1(77). С. 77—81.
41. *Afanasyeva A.*, *Buzdalov M.* Choosing Best Fitness Function with Reinforcement Learning / Proceedings of the Tenth International Conference on Machine Learning and Applications, ICMLA 2011. Vol. 2. Honolulu, HI, USA: IEEE Computer Society, 2011. Pp. 354–357.