ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”

Факультет Информационных технологий и программирования

Направление подготовки 010400.68.04 Специализация Технологии проектирования и разработки программного обеспечения

Квалификация (степень) Магистр

Специальное звание

Кафедра Компьютерных технологий Группа 6539

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему

Выбор вспомогательных оптимизируемых величин для повышения эффективности эволюционных алгоритмов в нестационарных условиях

Автор магистерской диссертации Петрова И. А. (подпись)

Научный руководитель Шалыто А. А. (подпись)

Руководитель магистерской программы (подпись)

К з а щ и т е д о п у с т и т ь
Зав. кафедрой Васильев В. Н. (подпись)

“____” _____2015 г.

Санкт-Петербург, 2015 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Существуют методы повышения эффективности эволюционных алгоритмов при помощи вспомогательных критериев. Рассматриваются два таких метода — EA+RL и MOEA+RL. В методе EA+RL оптимизируемый критерий — целевой или один из вспомогательных выбирается на каждом шаге однокритериального эволюционного алгоритма. В методе MOEA+RL на каждом шаге многокритериального эволюционного алгоритма одновременно оптимизируются целевой критерий и один из вспомогательных. В обоих методах выбор оптимизируемого критерия осуществляется при помощи обучения с подкреплением. В существующих исследованиях используются алгоритмы обучения с подкреплением в стационарной среде. Однако существуют задачи, в которых свойства вспомогательных критериев меняются в ходе процесса оптимизации.

В данной работе предлагается алгоритм обучения с подкреплением, применимый в методах EA+RL и MOEA+RL для выбора оптимизируемого критерия на каждом шаге эволюционного алгоритма в условиях нестационарности, заключающейся в изменении свойств вспомогательных критериев в зависимости от этапа оптимизации. Приводятся результаты применения предложенного алгоритма для решения модельной задачи. Также приводятся результаты применения разработанного алгоритма для решения задачи коммивояжера. Проводится сравнение с существующими методами решения задачи коммивояжера, использующими эволюционные алгоритмы с применением вспомогательных критериев.

ГЛАВА 1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Эволюционные алгоритмы

Существуют задачи оптимизиции, для которых точный алгоритм решения является неэффективным или его не существует. Примером таких задач служат задача составления расписаний, задача коммивояжера, задача о рюкзаке. Одним из методов решения данных задач является применение эволюционных алгоритмов [1, 2].

Эволюционные алгоритмы (ЭА) основаны на принципах природной эволюции. Кандидаты на оптимальное решение задачи представляются в виде особей эволюционного алгоритма. На каждой итерации алгоритма существует набор особей, называемый поколением. То, насколько особь близка к оптимальному решению, определяется функцией приспособленности (ФП). Для получения следующего поколения к особям применяются операторы скрещивания, мутации и отбора, использующие значения ФП особей.

Наиболее часто использующимися условиями останова выполнения ЭА являются нахождение оптимального решения и достижение заданного числа итераций [3]. В первом случае мерой эффективности ЭА является число итераций, необходимое для нахождения оптимального решения. Второе условие используется в случаях, когда оптимальное решение неизвестно или время, необходимое для его поиска, слишком велико. В таком случае эффективность эволюционного алгоритма оценивается по лучшему значению ФП, полученному за заданное число итераций алгоритма.

Отметим, что также существуют многокритериальные ЭА (Multi-Objective Evolutionary Algorithms, MOEA), предназначенные для решения задач многокритериальной оптимизиции. Одним из наиболее эффективных многокритериальных ЭА является NSGA-II [4].

1.2. Использование вспомогательных критериев

Эффективность решения задачи оптимизации при помощи эволюционного алгоритма можно повысить при помощи вспомогательных критериев [5—7]. Использование вспомогательных критериев может помочь решить такие проблемы ЭА, как остановка в локальном оптимуме и недостаток разнообразия особей. Во всех методах, использующих вспомогательные критерии, будем называть оптимизируемый критерий целевым критерием или целевой ФП, а остальные критерии вспомогательными критериями или вспомогательными

ФП. Рассмотрим различные подходы к созданию и использованию вспомогательных критериев.

В одном из подходов вспомогательные критерии получаются путем декомпозиции целевого [8, 9]. Полученные вспомогательные критерии одновременно оптимизируются вместо целевого при помощи многокритериального ЭА. Результатом работы многокритериального ЭА является набор особей оптимальных по Парето. Поэтому в данном подходе важно, чтобы оптимальное решение задачи являлось Парето-оптимальным при оптимизации вспомогательных критериев. Недостатком такого подхода является то, что полученные вспомогательные критерии должны быть независимы, что не всегда легко обеспечить.

В другом подходе вводятся дополнительные критерии, коррелирующие с целевым [10]. На каждом шаге алгоритма некоторые из них оптимизируются вместе с целевым при помощи многокритериального ЭА. Отметим, что задача оптимизации самих вспомогательных критериев не ставится, они используются лишь для повышения эффективности оптимизации целевого критерия.

Определим эффективность использования вспомогательного критерия как эффективность ЭА при решении задачи оптимизации с помощью данного критерия. Однако нельзя определить, какой критерий наиболее эффективен на текущем этапе оптимизации, так как свойства вспомогательных критериев заранее не известны. Оптимизация неэффективного вспомогательного критерия может привести к ухудшению текущего решения [10]. Существуют различные способы выбора вспомогательных критериев, оптимизируемых на каждом шаге алгоритма. Одним из них является случайный выбор [10]. Другим является использование некоторой эвристики [11]. Первый подход является общим, но он не учитывает специфику задачи. Второй подход был разработан специально для решения задачи составления расписаний и его применение для решения других задач затруднительно. Поэтому были предложены методы выбора вспомогательных критериев при помощи обучения с подкреплением.

1.3. Выбор вспомогательных критериев при помощи обучения с подкреплением

Опишем общую схему работы алгоритмов обучения с подкреплением [12, 13]. На каждом шаге агент обучения выбирает действие a и применяет его к среде, находящейся в состоянии s. Среда возвращает агенту награду и

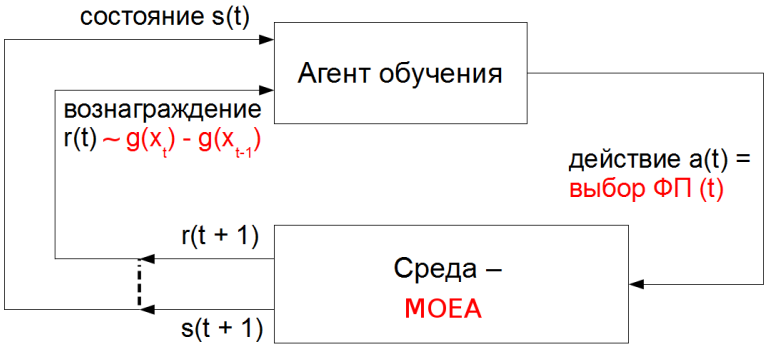


Рисунок 1 – Схема метода MOEA+RL

некоторое представление состояния, в котором она оказалась. Агент использует полученную информацию для обучения и процесс повторяется. Целью агента является максимизация суммарной награды.

В ранее предложенном автором методе выбора вспомогательных критериев MOEA+RL (MOEA + Reinforcement learning) [14] в качестве среды выступает многокритериальный ЭА, а действием является выбор вспомогательного критерия, оптимизируемого на текущей итерации алгоритма одновременно с целевым. Схема метода MOEA+RL представлена на рис. 1, где t — номер итерации алгоритма. Таким образом, в методе MOEA+RL агент учится выбирать наиболее эффективный вспомогательный критерий на каждом шаге алгоритма. В методе MOEA+RL суммарная награда оценивается с использованием модели бесконечного горизонта [15] по формуле $E[\sum_{t=0}^{\infty} \gamma^t r_t]$, где $0 < \gamma < 1$ — дисконтный фактор и r_t — награда, полученная на итерации t. Награда r_t зависит от разности значений целевой ФП в поколениях $t+1$ и t. Причем значение награды r_t тем выше, чем ближе значение целевой ФП в поколении $t+1$ к оптимальному. Поэтому агент, максимизируя суммарную награду, максимизирует прирост целевой ФП за все время работы алгоритма, таким образом максимизируя целевую ФП. Метод MOEA+RL был успешно применен для решения задачи составления расписаний [14] и задачи о генерации тестов [16].

Также существует метод использования вспомогательных критериев EA+RL [17, 18]. В отличие от метода MOEA+RL в нем оптимизируется один выбранный критерий — целевой или один из вспомогательных. Критерии так же, как и в методе MOEA+RL, выбираются при помощи обучения с подкреплением. Метод EA+RL основывается на том факте, что агент максимизируя суммарную награду, неявно максимизирует целевой критерий. Это позволяет не оптимизировать целевой критерий явно и использовать однокритериальный

ЭА, который обычно требует меньше вычислительных затрат, чем многокритериальный. Эффективность метода EA+RL была продемонстрирована и теоретически доказана на ряде модельных задач [17, 19, 20]. Однако, как будет показано в данной работе, иногда неявной оптимизации целевого критерия бывает недостаточно, и метод EA+RL в этом случае является неэффективным.

Ранее в исследованиях предполагалось, что среда является стационарной. Поэтому использовались алгоритмы обучения с подкреплением в стационарной среде. Среда является стационарной, если полученная награда зависит только от состояния среды и примененного действия [12]. Существующие алгоритмы обучения с подкреплением можно разделить на два типа: строящие модель среды и не строящие модель среды. В алгоритмах, не строящих модель среды, например в алгоритме Q-обучения [15], значение ожидаемой награды обновляется в соответствии со значением награды, полученной агентом на текущей итерации. В алгоритмах, строящих модель среды, например в алгоритме Dyna [15, 21], строятся модель переходов между состояниями T и модель награды R. Данные модели обновляются в соответствии со значением полученной агентом награды, а затем значение ожидаемой награды обновляется в соответствии с T и R.

Однако в случае когда свойства вспомогательных критериев меняются в процессе оптимизации, награда, полученная в результате применения одного и того же действия в одном и том же состоянии, может быть разной. Есть два способа бороться с возникающей в этом случае нестационарностью. Одним из них является использование такого определения состояния среды, при котором не будет возникать нестационарности. Однако разработка такого состояния сложна и не всегда возможна. Второй способ заключается в использовании алгоритмов обучения с подкреплением в нестационарной среде. Таким образом, необходима разработка алгоритма обучения с подкреплением в нестационарной среде, применимого для выбора вспомогательных критериев.

1.4. Выводы по главе 1

Описаны основные принципы работы эволюционных алгоритмов, применяющихся для решения задач однокритериальной оптимизации в данной работе. Приведен обзор методов повышения эффективности эволюционных алгоритмов при помощи вспомогательных критериев. Приведены методы выбора критериев с помощью обучения с подкреплением в ходе процесса оптимиза-

ции. Описаны основные принципы работы алгоритмов обучения с подкреплением в стационарной среде. Отмечен случай, когда в задаче выбора критериев оптимизации при помощи обучения с подкреплением необходимо использование алгоритмов обучения с подкреплением в нестационарной среде.

ГЛАВА 2. ЗАДАЧА ВЫБОРА КРИТЕРИЕВ В НЕСТАЦИОНАРНОЙ СРЕДЕ И ПРЕДЛАГАЕМЫЕ МЕТОДЫ ЕЕ РЕШЕНИЯ

В данной главе описывается решаемая задача и требования, предъявляемые к методу ее решения. Рассматриваются существующие алгоритмы обучения с подкреплением в нестационарной среде. Описываются первая и вторая (улучшенная) версии предлагаемого алгоритма обучения с подкреплением для динамического выбора вспомогательного критерия.

2.1. Постановка задачи

Дана задача однокритериальной оптимизации, в которой пространство поиска дискретно. Решение данной задачи производится с помощью эволюционного алгоритма. Также дан набор вспомогательных критериев. Оптимизация некоторых из них может привести к уменьшению числа итераций эволюционного алгоритма, необходимых для нахождения глобального оптимума. Рассматриваются два варианта решения задачи: при помощи метода EA+RL и при помощи метода MOEA+RL. Ранее в данных методах использовались алгоритмы обучения в стационарной среде. В данной работе предлагается рассмотреть случай, когда свойства вспомогательных критериев меняются в ходе процесса оптимизации, вследствие чего возникает нестационарность.

2.2. Требования к алгоритму

Требуется разработать алгоритм обучения с подкреплением для выбора оптимизируемого критерия на каждом шаге ЭА в условиях нестационарности, заключающейся в изменении свойств вспомогательных критериев в зависимости от этапа оптимизации, обладающий следующими характеристиками:

- возможность использования в методах EA+RL и MOEA+RL;
- результаты применения для решения задачи, в которой свойства вспомогательных критериев меняются в ходе процесса оптимизации, должны превосходить результаты использования ранее применявшихся алгоритмов обучения с подкреплением.

Чтобы проверить, удовлетворяет ли предложенный метод указанным требованиям, необходимо применить его для решения модельной и практической задачи совместно с методами EA+RL и MOEA+RL. Также требуется провести сравнение полученных результатов с результатами применения существующих алгоритмов обучения с подкреплением для стационарной среды.

2.3. Применение существующих методов обучения с подкреплением в нестационарной среде к решению поставленной задачи

Рассмотрим различные методы обучения с подкреплением в нестационарной среде. Некоторые из них применимы в случае непрерывных задач [22, 23]. Однако задача выбора вспомогательных критериев в ЭА является дискретной, поэтому использовать такие алгоритмы в данной работе затруднительно. Существуют алгоритмы обучения с подкреплением, применимые для решения дискретных задач, в которых нестационарность заключается в изменяющейся функции награды [24, 25]. Такие алгоритмы нецелесообразно применять в данной работе, так как функция награды в методах EA+RL и MOEA+RL не меняется.

Наиболее подходящим для выбора вспомогательных критериев в ЭА является алгоритм RLCD (Reinforcement Learning Context Detection) [26]. Данный алгоритм применим в случае, когда процесс оптимизации, в ходе которого свойства среды меняются, может быть разбит на интервалы, в которых свойства среды постоянны. Такие интервалы называются контекстами. Для каждого контекста строится модель среды. Эффективность модели определяется ее способностью предсказывать поведение среды. Идея алгоритма RLCD заключается в том, что алгоритм строит несколько моделей среды. На каждом шаге алгоритма выбирается и используется наиболее эффективная в данный момент модель, называемая активной. В случае когда у всех моделей эффективность ниже заданного порога, создается новая модель. Для построения модели используются алгоритмы обучения с подкреплением, строящие модель среды – Dyna и Prioritized sweeping [15, 21].

Таким образом, при использовании алгоритма RLCD в методах EA+RL (MOEA+RL), на каждом шаге алгоритма, действие – выбор вспомогательного критерия – определяется агентом активной модели. Активная модель обновляется, затем для каждой модели вычисляется функция эффективности и процесс повторяется. Однако как будет показано в разделе 2.7.3 результаты применения алгоритма RLCD для решения описанной ниже задачи оказались хуже, чем результаты, полученные с помощью ранее применявшихся алгоритмов обучения с подкреплением. Поэтому был предложен новый алгоритм обучения с подкреплением в нестационарной среде.

2.4. Первая версия предлагаемого алгоритма

В ходе предварительных экспериментов лучшие результаты были получены при применении алгоритма Q-обучения. Поэтому данный алгоритм обучения с подкреплением был использован в новом подходе. Так же как в алгоритме классического Q-обучения, на каждой итерации агент применяет действие a к среде, находящейся в состоянии s. Затем значение ожидаемой награды $Q(s, a)$ обновляется в соответствии с полученной наградой. В основе первой версии алгоритма используется перезапуск алгоритма обучения при условии, что $|Q(s, a) - Q(s', a')| < \delta$ для какой-то пары (s, a) и (s', a'). Перезапуск обучения связан с тем, что в описанном случае ожидаемая награда примерно одинакова для хотя бы одной пары действий, и агент не может определить, какое из них более эффективно. Поэтому обучение стало неэффективным и требуется перезапуск. Предполагается, что такая ситуация возникает при изменении свойств вспомогательных критериев. Псевдокод первой версии алгоритма приведен в листинге 1.

2.5. Экспериментальные исследования первой версии предложенного алгоритма

В данном разделе описаны экспериментальные исследования и результаты применения первой версии алгоритма совместно с методом EA+RL для решения модельной задачи.

2.5.1. Модельная задача

Рассмотрим постановку модельной задачи с двумя вспомогательными критериями, которые могут быть как эффективными, так и неэффективными на разных этапах оптимизации. В этой задаче особи представляются битовыми строками длины n. Пусть x — число бит, равных единице. Целевая ФП задается формулой: $\lfloor \frac{x}{k} \rfloor$, где k — константа, n нацело делится на k. Необходимо максимизировать значение целевой ФП. Вспомогательные ФП имеют вид:

$$h_1(x) = \begin{cases} x, x \leqslant p_1 \\ p_1, p_1 < x \leqslant p_2 \\ x, p_2 < x \leqslant p_3 \\ \dots \\ x, p_s < x \leqslant n \end{cases} \quad h_2(x) = \begin{cases} p_1, x \leqslant p_1 \\ x, p_1 < x \leqslant p_2 \\ p_3, p_2 < x \leqslant p_3 \\ \dots \\ n, p_s < x \leqslant n \end{cases}$$

Листинг 1 – Первая версия предлагаемого алгоритма : Q-обучение с перезапусками

1: Сформировать начальное поколение G_0
2: Инициализировать $Q(s,a) \leftarrow 0$ для каждого состояния s и действия a
3: Инициализировать счетчик числа итераций: $k \leftarrow 0$
4: **while** (не достигнуто заданное число итераций или максимальное значение целевой ФП) **do**
5: Вычислить текущее состояние s_k и передать его агенту
6: Выбрать действие a, где $Q(s,a) \leftarrow \max\limits_{a'} Q(s,a')$
7: Сформировать следующее поколение G_{k+1}
8: Вычислить награду *r*
9: $Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max\limits_{a'} Q(s,a') - Q(s,a))$
10: Инициализировать переменную для перезапуска: $reset \leftarrow false$
11: **for** (каждой пары (s,a)) **do**
12: **for** (каждой пары (s',a')) **do**
13: **if** $Q(s,a) \neq 0$ и $Q(s',a') \neq 0$ и $|Q(s,a) - Q(s',a')| \leqslant \delta$ **then**
14: Необходим перезапуск обучения: $reset \leftarrow true$
15: **end if**
16: **end for**
17: **end for**
18: **if** необходим перезапуск обучения: $reset = true$ **then**
19: Обнулить значение *Q(s, a)* для каждого состояния *s* и действия *a* : $Q(s,a) \leftarrow 0$
20: **end if**
21: Обновить счетчик числа итераций: $k \leftarrow k + 1$
22: **end while**

В точках p_i вспомогательные ФП меняют свои свойства, будем называть p_i точками переключения. Графики вспомогательных и целевого критериев представлены на рис. 2

Вспомогательный критерий h_1 эффективен, когда $x \in [0, p_1], (p_2, p_3], \ldots, (p_s, n]$, а h_2 эффективен на всех остальных отрезках. Отметим, что в данной задаче особи с разным числом единиц могут иметь одинаковое значение целевой ФП. Однако особь, в которой содержится большее число единиц с большей вероятностью породит особь с более высоким значением целевой ФП. Использование правильного вспомогательного критерия позволяет различить особи с одинаковым значением целевой ФП, так как эффективный вспомогательный критерий равен числу единиц в особи. В идеальном случае, на каждой итерации метод выбора критериев

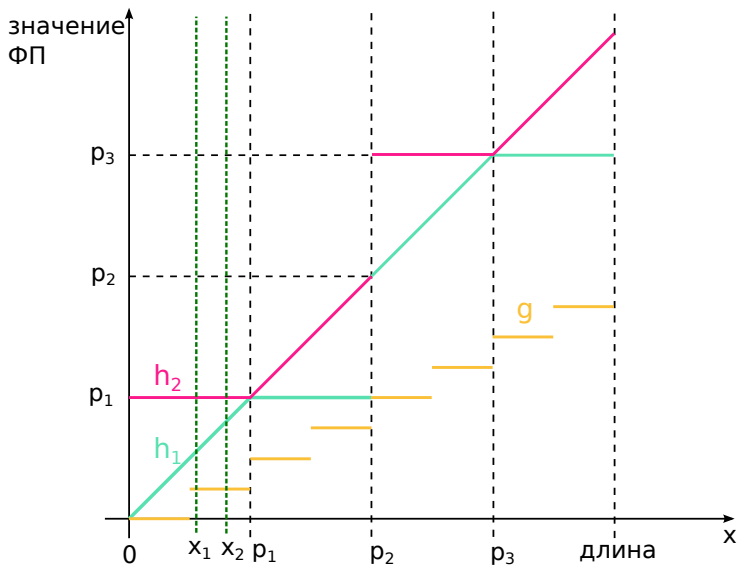


Рисунок 2 – Модельная задача

должен выбрать в качестве оптимизируемого критерия тот вспомогательный критерий, который не является константой на текущем интервале значений x.

2.5.2. Описание экспериментов

В ходе экспериментов сравнивались результаты применения метода EA+RL к различным конфигурациям модельной задачи с использованием различных алгоритмов обучения с подкреплением. Результаты работы каждого алгоритма усреднялись за 100 запусков. Рассматривались конфигурации задачи с пятью и десятью точками переключения. Рассматривались задачи с различными длинами особей, причем в случае десяти точек переключения длины были больше, чем в случае с пятью точками переключения. Точки переключения располагались равномерно по длине особи. В качестве параметра k были выбраны значения 10 и 25. Поколение ЭА состояло из 100 особей. Оператор мутации изменял каждый бит с вероятностью 0.001. Оператор скрещивания [27] применялся с вероятностью 0.7. Выполнение ЭА останавливалось при достижении заданного числа итераций или максимального значения целевой ФП.

В качестве алгоритмов обучения с подкреплением использовались ε-жадное Q-обучение, отложенное Q-обучение [28] и первая версия алгоритма. Первые два алгоритма ранее успешно применялись в методе EA+RL. Параметры для них были взяты из статьи [18]. В качестве параметров для ε-жадного Q-обучения использовались: $\alpha = 0.6$, $\gamma = 0.01$, $\varepsilon = 0.03$. Отложенное Q-

обучение использовалось с параметрами: $\alpha = 0.6$, $\gamma = 0.01$, $\varepsilon = 0.4$, $m = 5$. Параметры для предлагаемого подхода были выбраны в ходе предварительных экспериментов: $\alpha = 0.6$, $\gamma = 0.01$ и $\delta = 0.001$. Так же как и в исследованиях метода EA+RL, описанных в статье [18], состояния представлялись в виде вектора порядковых номеров ФП, упорядоченных по значению $\frac{|f(x_c)-f(x_p)|}{f(x_c)}$, где f — ФП, x_p — число бит, равных единице в лучшей особи предыдущего поколения, x_c — число бит, равных единице в лучшей особи текущего поколения. Лучшей особью в поколении называется особь с наибольшим значением целевой ФП. Для проверки статистической различимости нового подхода и ранее применявшихся методов был проведен тест суммы рангов Уилкоксона. Данный тест является непараметрическим и не предъявляет требований к свойствам анализируемых данных, в частности, к виду распределения. Уровень статистической значимости был равен $\alpha = 0.01$. Тест Уилкоксона проводился следующим образом. Каждый из рассмотренных алгоритмов запускался по 100 раз. При помощи функции $wilcox.test$ в языке R [29] сравнивались наборы результатов решения каждой конфигурации задачи, полученных при помощи нового и каждого из ранее применявшихся подходов.

2.5.3. Результаты экспериментов

Результаты экспериментов представлены в Таблице 1. В первой колонке указано число точек переключения, во второй и третьей – число итераций и длина особи соответственно. В трех последних колонках указаны средние значения целевой ФП, полученные за указанное в таблице число итераций, при использовании предлагаемого алгоритма, ε-жадного Q-обучения и отложенного Q-обучения соответственно. Среднеквадратичное отклонение при использовании первых двух алгоритмов составило около 0.5%, при использовании отложенного Q-обучения около 20%. Для всех рассмотренных конфигураций модельной задачи результаты, полученные при использовании предложенного подхода, превосходят результаты существующих алгоритмов.

Значения p-value, полученные при сравнении нового подхода с ε-жадным Q-обучением и отложенным Q-обучением, приведены в скобках в соответствующих колонках. Можно видеть, что в случаях, когда длина особи превышала 1000, p-value, полученные при сравнении нового подхода с ε-жадным Q-обучением, меньше уровня статистической значимости, что говорит о различимости этих подходов. Однако новый метод не всегда статистически различим с

Таблица 1 – Среднее значение целевой ФП

p_i	Число итераций	Длина	Предложенный ε-жадное Q-обучение	Отложенное Q-обучение	
$k = 10$					
5	3000	750	74.52	74.49 (3.4×10^{-1})	68.39 (4.4×10^{-10})
		1000	99.55	99.47 (1.3×10^{-1})	87.39 (2.2×10^{-16})
		1250	124.44	124.19 (8.3×10^{-4})	113.69 (2.2×10^{-16})
		1500	149.03	148.02 (1.5×10^{-3})	133.50 (8.1×10^{-15})
		1750	173.98	173.63 (8.4×10^{-8})	156.93 (1.9×10^{-13})
	5000	2000	198.93	197.98 (2.2×10^{-16})	186.37 (9.5×10^{-14})
		2250	222.01	220.23 (7.2×10^{-11})	202.80 (1.5×10^{-3})
		2500	245.52	244.55 (3.0×10^{-3})	232.81 (9.9×10^{-1})
10	5000	2000	198.94	198.34 (1.8×10^{-12})	170.59 (2.7×10^{-13})
		2250	223.36	220.79 (2.2×10^{-16})	184.47 (1.1×10^{-12})
		2500	245.28	244.61 (1.2×10^{-4})	204.42 (7.2×10^{-1})
	9000	2750	269.38	269.14 (5.0×10^{-3})	226.14 (5.7×10^{-1})
		3000	294.22	293.73 (2.8×10^{-5})	249.96 (9.6×10^{-1})
		3250	318.92	318.70 (1.4×10^{-2})	268.66 (9.7×10^{-1})
		3500	343.79	343.33 (4.0×10^{-5})	285.76 (9.9×10^{-1})
		3750	368.52	367.90 (1.3×10^{-5})	307.72 (9.9×10^{-1})
$k = 25$					
5	3000	750	29.45	29.40 (2.5×10^{-1})	26.01 (2.2×10^{-16})
		1000	39.18	39.14 (2.2×10^{-1})	36.20 (8.9×10^{-14})
		1250	49.02	49.00 (2.4×10^{-1})	45.86 (2.8×10^{-9})
		1500	59.00	58.92 (6.0×10^{-3})	54.77 (4.3×10^{-8})
		1750	68.96	68.02 (4.0×10^{-15})	61.17 (1.8×10^{-11})
	5000	2000	78.17	77.23 (4.9×10^{-16})	70.54 (1.9×10^{-1})
		2250	87.30	87.12 (8.0×10^{-3})	79.70 (9.8×10^{-1})
		2500	97.01	96.89 (4.0×10^{-3})	84.62 (5.3×10^{-1})

отложенным Q-обучением, несмотря на то, что среднее значение целевой ФП, полученное с помощью нового подхода, значительно выше. Это можно объяснить большим разбросом значений целевой ФП при применении отложенного Q-обучения.

Агент может выбирать на каждой итерации одну из вспомогательных ФП или целевую ФП. Наиболее эффективно выбирать ту ФП, которая в текущем поколении равна x. Будем называть выбор эффективной ФП хорошим. На рис. 3 представлено число выборов ФП в ходе решения модельной задачи с пятью точками переключения, $k = 10$, $n = 750$. По горизонтали указан номер итерации, а ширина полосы по вертикальной оси соответствует числу выборов соответствующей ФП в 100 запусках. В начале оптимизации предлагаемый подход делает хороший выбор (h_1) чаще, чем другие методы, затем предлагаемый подход делает хороший выбор (h_2) чаще, чем другие методы,

и так далее. Таким образом новый подход делает хороший выбор чаще, чем другие методы.

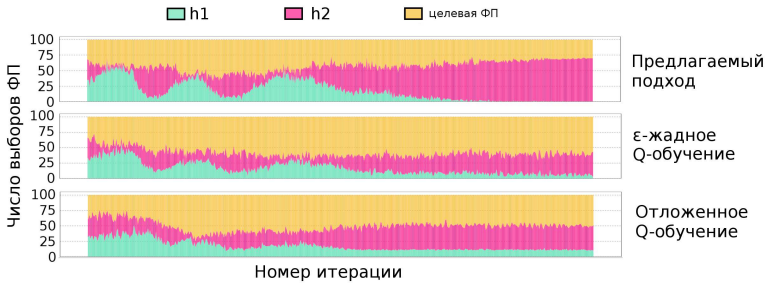


Рисунок 3 – Число выборов ФП

В Таблице 2 представлен усредненный процент числа хороших выборов ФП для конфигураций со значением параметра $k = 10$. В первой колонке указано число точек переключения, во второй – длина особи. В последних трех колонках указаны средние значения числа хороших выборов в процентах от общего числа выборов ФП, полученные при использовании предлагаемого подхода, ε-жадного Q-обучения и отложенного Q-обучения соответственно. Среднеквадратичное отклонение при использовании первых двух алгоритмов составило 8%, при использовании отложенного Q-обучения — 100%. Значения p-value, полученные при сравнении нового подхода с ε-жадным Q-обучением и отложенным Q-обучением, приведены в скобках в соответствующих колонках. Можно видеть, что в случаях, когда длина особи превышала 1000, полученные p-value меньше уровня статистической значимости, что говорит о различимости нового подхода и используемых ранее методов.

Можно видеть, что предлагаемый подход делает хорошие выборы чаще, чем ε-жадное Q-обучение. Однако существуют конфигурации задачи, на которых отложенное Q-обучение делает больше хороших выборов, чем новый подход. В то же время среднее значение целевой ФП, полученное при применении отложенного Q-обучения хуже, чем при применении нового метода. Это можно объяснить тем, что среднеквадратичное отклонение для отложенного Q-обучения гораздо больше, чем для нового подхода.

2.6. Вторая версия предложенного алгоритма

Недостатком первой версии предложенного алгоритма является то, что после изменения свойств вспомогательных критериев, алгоритму требуется большое число итераций до того момента, как значение ожидаемой награды

Таблица 2 – Число хороших выборов, $k = 10$

Число p_i	Длина	Число хороших выборов,%		
		Предложенный	ε-жадное Q-обучение	Отложенное Q-обучение
5	750	62	50 (1.3×10^{-5})	46 (8.0×10^{-3})
	1000	55	51 (1.0×10^{-2})	47 (2.6×10^{-1})
	1250	51	43 (7.0×10^{-4})	36 (1.7×10^{-5})
	1500	50	39 (2.2×10^{-16})	35 (1.0×10^{-10})
	1750	48	39 (2.2×10^{-16})	37 (9.2×10^{-10})
	2000	46	39 (2.2×10^{-16})	29 (2.1×10^{-15})
	2250	37	23 (2.2×10^{-16})	37 (8.5×10^{-7})
	2500	30	17 (2.2×10^{-16})	26 (4.1×10^{-14})
10	2000	47	49 (1.3×10^{-8})	33 (6.9×10^{-13})
	2250	42	29 (2.2×10^{-16})	36 (3.6×10^{-6})
	2500	26	19 (2.2×10^{-16})	38 (1.2×10^{-4})
	2750	29	21 (2.2×10^{-16})	41 (3.9×10^{-3})
	3000	34	26 (2.2×10^{-16})	33 (5.1×10^{-7})
	3250	39	29 (2.2×10^{-16})	36 (1.6×10^{-5})
	3500	41	33 (2.2×10^{-16})	38 (4.6×10^{-5})
	3750	43	35 (2.2×10^{-16})	37 (4.6×10^{-5})

станет примерно одинаковым для хотя бы одной пары действий, и произойдет перезапуск обучения. Поэтому была разработана вторая версии предложенного алгоритма, отличающаяся от первой условием перезапуска Q-обучения. В листинге 2 приведен псевдокод второй версии алгоритма. Во второй версии алгоритма есть два условия для перезапуска обучения. Первое условие (строка 21 в листинге 2) выполнено, если 1) награда меньше или равна нулю в течение нескольких последовательных итераций и 2) в c_1 из этих итераций награда строго меньше нуля, где c_1 – некоторая константа. Второе условие (строка 24 в листинге 2) работает в случае, когда значение целевой ФП может не изменяться на протяжении нескольких итераций. Условие выполнено, если награда меньше или равна нулю в течение нескольких последовательных итераций и в c_2 из этих итераций награда равна нулю, где c_2 – некоторая константа. В этом случае перезапуск происходит с вероятностью $\frac{\text{optimal}-\text{current}}{\text{optimal}}$, где $optimal$ — это максимальное значение целевой ФП, а $current$ — значение целевой ФП на данной итерации. Предполагается, что выполнение первого условия означает изменение свойств вспомогательных критериев. Выполнение второго условия отражает остановку ЭА в локальном оптимуме. Стоит отметить, что в конце процесса оптимизации для нахождения лучшего решения требуется большее число итераций. Поэтому для предотвращения частого перезапуска

Q-обучения в конце процесса оптимизации, вероятность перезапуска уменьшается во время оптимизации.

2.7. Экспериментальные исследования второй версии предложенного алгоритма

В данном разделе описаны экспериментальные исследования и результаты применения второй версии предложенного алгоритма совместно с методом EA+RL для решения модельной задачи.

2.7.1. Модельная задача

Постановка модельной задачи аналогична задаче, описанной в разделе 2.5.1. Отличие данной модельной задачи от рассматриваемой ранее заключается в том, что вспомогательные ФП имеют следующий вид:

$$h_1(x) = \begin{cases} x, x \leqslant p_1 \\ -x + 2p_1, p_1 < x \leqslant p_2 \\ x, p_2 < x \leqslant p_3 \\ \dots \\ x, p_s < x \leqslant n \end{cases} \qquad h_2(x) = \begin{cases} -x, x \leqslant p_1 \\ x, p_1 < x \leqslant p_2 \\ -x + 2p_2, p_2 < x \leqslant p_3 \\ \dots \\ -x + 2p_s, p_s < x \leqslant n \end{cases}$$

Вспомогательная и целевая ФП представлены на рис. 4. Вспомогательный критерий h_1 эффективен, когда $x \in [0, p_1], (p_2, p_3], \dots, (p_s, n]$, а h_2 эффективен во всех остальных случаях. Вспомогательный критерий h_i может быть отрицательным на интервале $[2p_{k-1}, p_k)$, если $p_k < 2p_{k-1}$. В связи с требованиями используемой библиотеки [30], значения ФП должны быть неотрицательными, поэтому было применено следующее преобразование: $h_i(x) \leftarrow h_i(x) + p_k - 2p_{k-1} = -x + 2p_{k-1} + p_k - 2p_{k-1} = -x + p_k$. Данное преобразование используется во всех экспериментах, описываемых далее.

Точки переключения выбирались следующим образом. Первая точка p_1 выбиралась случайно из интервала $(0, n)$. Точка p_i выбиралась случайно из интервала (p_{i-1}, n). В случае, если расстояние между точкой p_i и некоторой из точек $p_1 \dots p_{i-1}$ меньше 100, то точка p_i выбиралась заново.

Так же как и при решении задачи, представленной в разделе 2.5.1, в идеале, на каждой итерации в качестве оптимизируемого критерия должен быть

Листинг 2 – Вторая версия предложенного алгоритма

1: Сформировать начальное поколение G_0
2: Инициализировать $Q(s, a) \leftarrow 0$ для каждого состояния *s* и действия *a*
3: Инициализировать счетчик числа итераций: $k \leftarrow 0$
4: Инициализировать счетчики для перезапуска: $reset_1, reset_2 \leftarrow 0$
5: **while** (не достигнуто заданное число поколений или оптимальное значенмие целевой ФП) **do**
6: Вычислить текущее состояние s_k и передать его агенту
7: Выбрать действие $a : Q(s, a) = \max\limits_{a'} Q(s, a')$
8: Сформировать следующее поколение G_{k+1}
9: Вычислить полученную награду *r* и следующее состояние s'
10: $Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max\limits_{a'} Q(s', a') - Q(s, a))$
11: Инициализировать переменную для перезапуска: $reset \leftarrow false$
12: **if** награда меньше нуля: $r < 0$ **then**
13: Увеличить первый счетчик для перезапуска: $reset_1 \leftarrow reset_1 + 1$
14: **end if**
15: **if** награда равна нулю: $r = 0$ **then**
16: Увеличить второй счетчик для перезапуска: $reset_2 \leftarrow reset_2 + 1$
17: **end if**
18: **if** награда больше нуля: $r > 0$ **then**
19: Обнулить счетчики для перезапуска: $reset_1, reset_2 \leftarrow 0$
20: **end if**
21: **if** первое условие перезапуска выполнено: $reset_1 = c_1$ **then**
22: $reset \leftarrow true$
23: **end if**
24: **if** второе условие перезапуска выполнено: $reset_2 = c_2$ **then**
25: обнулить второй счетчик для перезапуска: $reset_2 \leftarrow 0$
26: сгенерировать случайное число n из отрезка $[0; 1]$
27: **if** $n \leqslant \frac{optimal - current}{optimal}$ **then**
28: $reset \leftarrow true$
29: **end if**
30: **end if**
31: **if** необходим перезапуск обучения: $reset = true$ **then**
32: Обнулить *Q(s, a)* для каждого состояния *s* и действия *a* : $Q(s, a) \leftarrow 0$
33: Обнулить первый счетчик для перезапуска: $reset_1 \leftarrow 0$
34: Обнулить второй счетчик для перезапуска: $reset_2 \leftarrow 0$
35: **end if**
36: Обновить счетчик числа итераций: $k \leftarrow k + 1$
37: **end while**

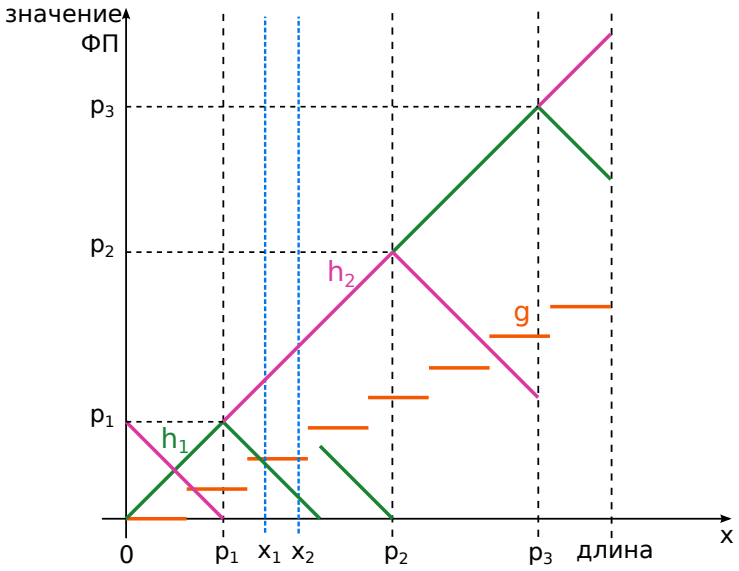


Рисунок 4 – Модельная задача

выбран тот вспомогательный критерий, который не является константой на текущем интервале значений x. Основное отличие данной модельной задачи от предыдущей заключается в том, что в случае выбора неэффективной вспомогательной ФП, значение целевой ФП может уменьшиться. Поэтому эта задача труднее, чем предыдущая.

2.7.2. Описание экспериментов

В ходе экспериментов сравнивались результаты решения модельной задачи при помощи метода EA+RL с использованием различных алгоритмов обучения с подкреплением. Результаты работы каждого алгоритма усреднялись за 100 запусков. Рассматривались конфигурации задачи с пятью точками переключения и значением параметра k равным 10. Число итераций во всех запусках было равно 5000.

Использовались такие же, как в ранее описанных экспериментах, параметры ЭА. В качестве алгоритмов обучения с подкреплением использовались вторая и первая версии предложенного подхода, ε-жадное Q-обучение, отложенное Q-обучение и RLCD. Параметры для второй версии предложенного подхода были выбраны в ходе предварительных экспериментов: $\alpha = 0.6$, $\gamma = 0.01$, $c_1 = 10$, $c_2 = 10$. Параметры для остальных алгоритмов обучения с подкреплением совпадают с параметрами, использовавшимися в экспериментах, описанных в разделе 2.5.2.

Состояние представляется в виде вектора, элементы которого отражают номер поколения, среднее значение целевой ФП в поколении и энтропию целевой ФП в поколении [31]. Номер поколения разделен на четыре интервала по логарифмической шкале. Среднее значение целевой ФП в поколении нормировано по оптимальному значению целевой ФП и разбито на четыре интервала. Значения энтропии разбиты на три интервала равной величины. Таким образом, состояние это вектор из трех чисел – номеров интервалов, в которые попадают номер поколения, среднее значение целевой ФП в поколении и энтропия целевой ФП в поколении соответственно.

Для проверки статистической различимости второй версии предложенного подхода и остальных применявшихся методов был проведен тест суммы рангов Уилкоксона. Уровень статистической значимости был равен $\alpha = 0.01$.

2.7.3. Результаты экспериментов

Средние полученные значения целевой ФП представлены в Таблице 3 в виде $\frac{\text{optimal}-\text{average}}{\text{optimal}}$, где $optimal$ это максимальное значение целевой ФП, $average$ – среднее значение целевой ФП. В первой колонке указана длина особи, в остальных пяти колонках указаны средние значения целевой ФП, полученные при использовании второй и первой версий предложенного алгоритма, ε-жадного Q-обучения, отложенного Q-обучения и RLCD соответственно. Среднеквадратичное отклонение при использовании отложенного Q-обучения составило 10%, а при использовании остальных алгоритмов 0,5%. Для всех рассмотренных конфигураций модельной задачи результаты, полученные при использовании второй версии алгоритма, превосходят результаты остальных алгоритмов. Значения p-value, полученные при сравнении второй версии алгоритма с другими алгоритмами, приведены в скобках в соответствующих колонках. Значения p-value, полученные при сравнении второй версии алгоритма с первой версией, отложенным Q-обучением и RLCD, меньше уровня статистической значимости, что говорит о различимости этих алгоритмов. Можно видеть, что в случаях, когда длина особи превышала 1000, p-value, полученные при сравнении второй версии алгоритма с ε-жадным Q-обучением, меньше уровня статистической значимости, что говорит о различимости этих подходов в этом случае.

Число запусков, в которых анализируемые алгоритмы достигли оптимального значения целевой ФП, приведены в Таблице 4. В первой колонке

Таблица 3 – Среднее значение целевой ФП

Длина	Вторая версия	Первая версия	ε-жадное Q-обучение	Отложенное Q-обучение	RLCD
750	0.56	1.60 (2.2×10^{-16})	0.64 (1.9×10^{-1})	11.43 (2.2×10^{-16})	5.23 (2.2×10^{-16})
1000	0.39	1.16 (2.2×10^{-16})	0.49 (7.7×10^{-2})	4.23 (3.6×10^{-16})	4.33 (2.2×10^{-16})
1250	0.34	1.02 (2.2×10^{-16})	1.44 (2.4×10^{-15})	15.28 (2.2×10^{-16})	4.76 (2.2×10^{-16})
1500	0.17	0.87 (2.2×10^{-16})	1.83 (2.2×10^{-16})	12.01 (2.2×10^{-16})	11.98 (2.2×10^{-16})
1750	0.27	1.14 (2.2×10^{-16})	1.97 (2.2×10^{-16})	16.56 (2.2×10^{-16})	8.37 (2.2×10^{-16})
2000	0.78	1.55 (2.2×10^{-16})	2.00 (2.2×10^{-16})	14.54 (2.2×10^{-16})	6.92 (2.2×10^{-16})
2250	1.07	1.59 (1.2×10^{-13})	1.98 (2.2×10^{-16})	21.91 (2.2×10^{-16})	15.39 (2.2×10^{-16})
2500	1.18	1.73 (2.2×10^{-16})	1.88 (2.2×10^{-16})	8.79 (2.2×10^{-16})	16.05 (2.2×10^{-16})
2750	1.35	1.68 (3.2×10^{-12})	1.96 (2.2×10^{-16})	12.32 (2.2×10^{-16})	8.34 (2.2×10^{-16})
3000	1.40	1.62 (1.2×10^{-11})	3.09 (2.2×10^{-16})	16.08 (2.2×10^{-16})	11.04 (2.2×10^{-16})
3250	1.45	1.52 (9.1×10^{-5})	3.70 (2.2×10^{-16})	5.37 (2.2×10^{-16})	18.06 (2.2×10^{-16})
3500	1.34	1.49 (3.0×10^{-7})	3.97 (2.2×10^{-16})	10.33 (2.2×10^{-16})	12.79 (2.2×10^{-16})
3750	1.32	2.03 (4.0×10^{-16})	4.13 (2.2×10^{-16})	13.31 (2.2×10^{-16})	20.01 (2.2×10^{-16})

указана длина особи, в остальных пяти колонках указано число запусков, в которых было достигнуто оптимальное значение целевой ФП при использовании второй и первой версий предложенного подхода, ε-жадного Q-обучения, отложенного Q-обучения и RLCD соответственно. В случаях, когда длина особи была меньше 2750, результаты, полученные при применении второй версии предложенного подхода, превосходят результаты, полученные при использовании остальных алгоритмов. На других конфигурациях модельной задачи при использовании ни одного из алгоритмов не было достигнуто оптимальное значение целевой ФП.

Таблица 4 – Число запусков, в которых было достигнуто оптимальное значение целевой ФП

Длина	Вторая версия	Первая версия	ε-жадное Q-обучение	Отложенное Q-обучение	RLCD
750	58	4	52	12	0
1000	61	0	51	15	0
1250	65	0	20	5	0
1500	76	0	0	1	0
1750	60	0	0	0	0
2000	60	0	0	0	0
2250	12	0	0	0	0
2500	2	0	0	0	0
2750	0	0	0	0	0

2.8. Выводы по главе 2

Поставлена задача разработки алгоритма обучения с подкреплением для выбора критерия на каждом шаге ЭА в случае изменения свойств вспомогательных критериев в процессе оптимизации. Сформулированы требования, предъявляемые к данному алгоритму. Рассмотрены существующие алгоритмы обучения с подкреплением в нестационарной среде. Один из них, RLCD, может быть применен для выбора вспомогательных критериев в ЭА. Однако в ходе предварительных экспериментов алгоритм RLCD оказался неэффективным. Поэтому был предложен новый алгоритм обучения с подкреплением для совместного применения с ЭА в нестационарных условиях.

Рассмотрены первая и вторая (улучшенная) версии предложенного алгоритма обучения с подкреплением для динамического выбора вспомогательного критерия. Представлены результаты применения первой и второй версий предложенного алгоритма совместно с методом EA+RL для решения модельной задачи. Для проверки статистической различимости первой версии алгоритма с существующими алгоритмами был проведен тест Уилкоксона. Результаты, полученные при использовании первой версии алгоритма, превосходят результаты работы существующих алгоритмов. Для проверки статистической различимости второй версии алгоритма с первой версией и существующими алгоритмами также был проведен тест Уилкоксона. Результаты, полученные при использовании второй версии алгоритма, превосходят результаты работы первой версии алгоритма и существующих алгоритмов. Результаты экспериментов подтверждают, что обе версии предложенного алгоритма соответствует требованиям, сформулированным в разделе 2.2.

ГЛАВА 3. ПРИМЕНЕНИЕ ПРЕДЛОЖЕННОГО АЛГОРИТМА ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЕРА

В данной главе проверяется эффективность применения второй версии предложенного алгоритма, описанной в разделе 2.6, в методах EA+RL и MOEA+RL на примере задачи коммивояжера. Проводится сравнение с существующими методами решения задачи коммивояжера при помощи эволюционных алгоритмов с использованием вспомогательных критериев.

3.1. Задача коммивояжера

Задача коммивояжера является одной из классических задач комбинаторной оптимизации [32]. В задаче коммивояжера рассматривается множество из n городов и матрица M расстояний между ними, размером $n \times n$. Значение $M(c_1, c_2)$ соответствует длине пути из города c_1 в город c_2. Если длина пути из города c_1 в город c_2 равна длине пути из города c_2 в город c_1, то задача коммивояжера является симметричной. В данной работе рассматриваются симметричные задачи коммивояжера. Целью задачи коммивояжера является нахождение пути π минимальной длины, проходящего через каждый город ровно один раз с возвратом в исходный город. Путь $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ представляется как перестановка городов $(1, 2, \ldots, n)$. Длина пути $D(\pi)$ вычисляется как

$$D(\pi) = \sum_{i=1}^{n} M(c_{\pi[i]}, c_{\pi[i \oplus 1]}), \text{где}$$

$$i \oplus 1 = \begin{cases} i + 1, i < n \\ 1, i = n \end{cases}$$

На рис. 5 представлен пример вычисления длины пути $\pi = (1, 3, 4, 2, 5)$ для задачи, в которой $n = 5$.

3.2. Методы решения задачи коммивояжера при помощи вспомогательных критериев

Существует несколько методов решения задачи коммивояжера при помощи ЭА с использованием вспомогательных критериев. В каждом из них особью в ЭА является текущий путь, а в качестве целевой ФП выступает длина пути. Каждому методу решения соответствует определенный способ задания вспомогательных ФП. Рассмотрим три подхода к решению задачи коммивояжера при помощи вспомогательных ФП.

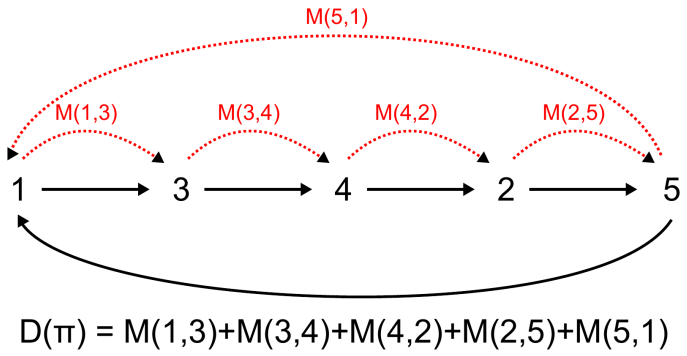


Рисунок 5 – Вычисление длины пути в задаче коммивояжера

3.2.1. Получение вспомогательных критериев из разбиения целевого

Данный метод предложен в статье Knowles et al. [8]. Путь разбивается двумя городами a и b на два подпути. Таким образом целевой критерий разбивается на два вспомогательных, которые соответствуют длинам подпутей. Значения вспомогательных ФП f_1 и f_2 вычисляются следующим образом:

$$f_1(\pi, a, b) = \sum_{i=\pi^{-1}[a]}^{\pi^{-1}[b]-1} M(c_{\pi[i]}, c_{\pi[i\oplus 1]}),$$

$$f_2(\pi, a, b) = \sum_{i=\pi^{-1}[b]}^{n} M(c_{\pi[i]}, c_{\pi[i\oplus 1]}) + \sum_{i=1}^{\pi^{-1}[a]-1} M(c_{\pi[i]}, c_{\pi[i\oplus 1]}),$$

где $\pi^{-1}[x]$ — позиция города x в пути π. Данный метод является методом декомпозиции, то есть на каждом шаге многокритериального ЭА оптимизируются две вспомогательные ФП вместо целевой. Отметим, что $f_1(\pi, a, b) + f_2(\pi, a, b) = D(\pi)$. Таким образом путь, на котором достигается оптимальное значение целевой ФП, является Парето-оптимальным при оптимизации вспомогательных критериев, что требуется в методе декомпозиции. На рис. 6 представлен пример вычисления вспомогательных ФП f_1 и f_2 для пути $\pi = (1, 3, 4, 2, 5)$, где $a = 2$, $b = 5$.

3.2.2. Введение новых вспомогательных критериев

Вспомогательные критерии, предложенные Knowles et al., при решении симметричных задач коммивояжера значения f_1 и f_2 могут быть различны для

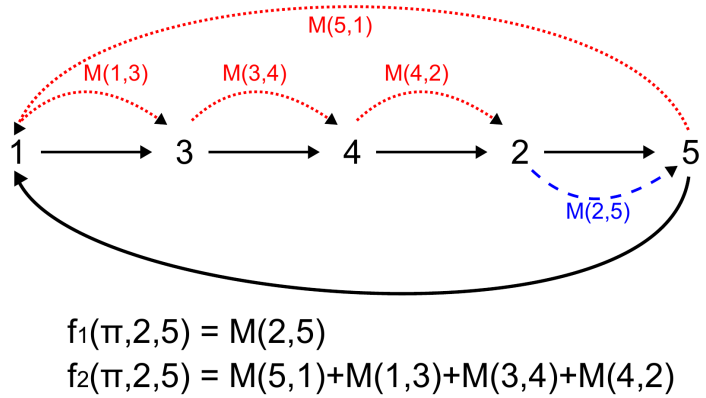


Рисунок 6 – Вспомогательные ФП, предложенные Knowles et al.

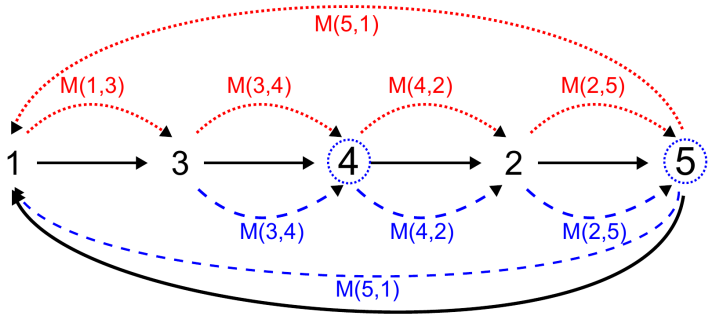
разных представлений одного и того же пути [10]. Поэтому Jensen предложил ввести новые вспомогательные ФП, коррелирующие с целевой ФП [10], которые не имеют данного недостатка. Значение вспомогательной ФП вычисляется следующим образом:

$$h(\pi, p) = \sum_{i=1}^{|p|} M(c_{\pi[\pi^{-1}[p[i]]\ominus 1]}, c_{\pi[i]}) + M(c_{\pi[i]}, c_{\pi[\pi^{-1}[p[i]]\oplus 1]}) \quad (1)$$

где p — подмножество городов $1, 2, \ldots, n$ и

$$i \ominus 1 = \begin{cases} i - 1, i > 0 \\ n, i = 0 \end{cases}$$

Подмножество городов p создается случайным образом, для каждого города вероятность попасть в p равна 50%. Наиболее эффективно на каждом шаге многокритериального ЭА оптимизировать целевую ФП и одну из вспомогательных [10]. Каждая из вспомогательных ФП оптимизируется на протяжении $\frac{T}{k}$ итераций многокритериального ЭА, где T — общее число итераций алгоритма, а k — число вспомогательных критериев. Таким образом, каждый вспомогательный критерий используется в равном числе итераций. Порядок выбора вспомогательных ФП является случайным. На рис. 7 представлен пример вычисления вспомогательной ФП h для пути $\pi = (1, 3, 4, 2, 5)$, где подмножество городов $p = \{4, 5\}$.



D(π) = M(1,3)+M(3,4)+M(4,2)+M(2,5)+M(5,1)

h(π,p) = [M(3,4)+M(4,2)] + [M(2,5)+M(5,1)]

Рисунок 7 – Вспомогательные ФП, предложенные Jensen

3.2.3. Получение вспомогательных критериев путем сегментации

Вспомогательные критерии в методе, предложенном Jensen, выбираются в случайном порядке, поэтому может быть выбран неэффективный вспомогательный критерий. Использование неэффективного вспомогательного критерия может привести к отдалению текущего найденного решения от оптимального значения. Поэтому Jähne et al. предложили преобразовать метод Jensen в метод декомпозиции, в котором нет данного недостатка, так как не нужно выбирать оптимизируемый критерий [9, 33].

Создаются два подмножества городов: подмножество p, формируемое аналогично методу Jensen, и подмножество p^C, являющееся дополняющим к подмножеству p. Затем создаются вспомогательные ФП: $h_1(\pi, p)$ и $h_2(\pi, p^C)$, определяемые в соответствии с формулой 1, как в методе Jensen. На каждом шаге многокритериального ЭА оптимизируются две вспомогательные ФП вместо целевой. Отметим, что $h_1(\pi, p) + h_2(\pi, p^C) = 2D(\pi)$, поэтому требование, что путь, на котором достигается оптимальное значение целевой ФП является Парето-оптимальным при оптимизации вспомогательных критериев, выполнено. На рис. 8 представлен пример вычисления вспомогательных ФП h_1 и h_2 для пути $\pi = (1, 3, 4, 2, 5)$, где подмножество городов $p = \{2, 3, 5\}$, а $p^C = \{1, 4\}$. В работе Jähne et al. показано, что данный подход превосходит методы Jensen и Knowles et al.

3.3. Предлагаемый метод решения задачи коммивояжера

Заметим, что современные исследования в области решения задачи коммивояжера при помощи ЭА с использованием вспомогательных критериев направлены на улучшение метода, предложенного Jähne et al. Однако в работе

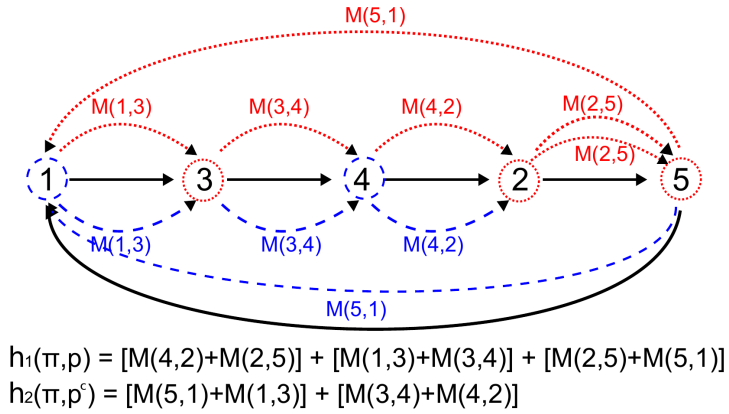


Рисунок 8 – Вспомогательные ФП, предложенные Jähne et al.

Jähne et al. используются вспомогательные критерии, предложенные Jensen, а мотивацией к преобразованию метода Jensen в метод декомпозиции послужило отсутствие метода выбора вспомогательного критерия. В настоящей работе предлагается решение задачи коммивояжера при помощи метода MOEA+RL в котором, также как и в методе Jensen, на каждом шаге ЭА оптимизируются целевой и один из вспомогательных критериев, однако вспомогательный критерий выбирается при помощи обучения с подкреплением. Также исследуется эффективность решения задачи коммивояжера при помощи метода EA+RL.

В качестве алгоритмов обучения с подкреплением рассматриваются алгоритмы обучения в стационарной и нестационарной среде. Поясним, почему в задаче коммивояжера целесообразно использование предложенного в разделе 2.6 алгоритма обучения с подкреплением, применимого в случае изменения свойств вспомогательных критериев в ходе процесса оптимизации. Отметим, что во всех рассмотренных определениях вспомогательных критериев их значения зависят от текущего пути. Путь является особью ЭА. Таким образом, свойства вспомогательных критериев зависят от особи, на которой они вычисляются, поэтому возможна нестационарность. На данный момент не существует состояния, предотвращающего данную нестационарность. Поэтому применялось одно из наиболее эффективных существующих определений состояния [31], которое ранее использовалось для решения модельной задачи, описанной в разделе 2.7.1. Одному и тому же значению состояния соответствует много различных особей, а значит в одном и том же состоянии вспомогательный критерий может иметь разные свойства. Поэтому среда в таком случае не является стационарной.

3.4. Описание экспериментов

В данной работе для решения задачи коммивояжера применялись методы EA+RL и MOEA+RL. Полученные результаты сравнивались с результатами, полученными при использовании методов, предложенных Knowles et al., Jensen и Jähne et al. Исследовалась применимость алгоритмов обучения с подкреплением в стационарной и нестационарной среде. Все алгоритмы запускались на фиксированном числе вычислений ФП. Подход Knowles et al. и метод EA+RL требуют гораздо больше вычислений ФП, чем метод MOEA+RL и подходы Jensen и Jähne et al. Поэтому метод EA+RL не сравнивался с подходами Jensen and Jähne et al. По той же причине метод MOEA+RL не сравнивался с подходом Knowles et al. Метод EA+RL сравнивался с традиционным ЭА, методом имитации отжига [1], и подходом, предложенным Knowles et al. Метод MOEA+RL сравнивался с подходами Jensen и Jähne et al. Рассматривались два алгоритма обучения с подкреплением. Одним из них является стационарный алгоритм Q-обучения с ε-жадной стратегией [12]. Другим является вторая версия предложенного алгоритма, описанная в разделе 2.6.

Детали реализации многокритериального ЭА NSGA-II могут влиять на результаты применяемых подходов [34]. Поэтому были самостоятельно реализованы подходы Jensen и Jähne et al. и все алгоритмы в экспериментах запускались на одной и той же реализации алгоритма NSGA-II.

В ходе предварительных экспериментов было получено, что наилучшие результаты достигаются при использовании награды, используемой в работе [18]. Функция награды выглядит следующим образом:

$$r_t = \begin{cases} 1 & \text{if } g_{t+1} - g_t > 0 \\ 0 & \text{if } g_{t+1} - g_t = 0 \\ -1 & \text{if } g_{t+1} - g_t < 0, \end{cases}$$

где g_t и g_{t+1} — лучшее значение целевой ФП в поколениях t и $t+1$ соответственно Все рассматриваемые алгоритмы запускались по 30 раз на каждой задаче коммивояжера, затем результаты усреднялись. Также как и в работах Knowles et al. и Jähne et al., вспомогательные ФП создавались один раз и использовались во всех запусках.

В задаче коммивояжера в отличие от модельной задачи, описанной в главе 2.7.1, оптимальное решение неизвестно. Поэтому используемое ранее условие, в соответствии с которым перезапуск производится с вероятностью $\frac{\text{optimal}-\text{current}}{\text{optimal}}$, где $optimal$ — это максимальное значение целевой ФП, а $current$ – значение целевой ФП на данной итерации, неприменимо. В ходе предварительных экспериментов исследовалась вероятность перезапуска, вычисленная по формуле $\frac{\text{optimal}-\text{current}}{\text{optimal}}$, где в качестве $optimal$ использовалось лучшее известное решение. Однако такой подход давал худшие результаты, чем использование постоянной вероятности перезапуска. Это можно объяснить тем, что в конце оптимизации разница между значением $optimal$ и $current$ становится очень мала по сравнению с $optimal$, поэтому вероятность перезапуска также становится очень маленькой. В ходе предварительных экспериментов анализировались вероятности перезапуска от 0 до 1 с шагом 0,1. Наилучшие результаты были получены при вероятности перезапуска, равной 0,5.

3.5. Эксперименты с методом EA+RL

В данном разделе описаны экспериментальные исследования и результаты применения метода EA+RL для решения задачи коммивояжера.

3.5.1. Описание экспериментов

Параметры для данного эксперимента взяты из работы Knowles et al. [8]. Также как и в [8] рассматривались пять задач коммивояжера. Метод EA+RL сравнивался с традиционным ЭА, методом имитации отжига, и подходом, предложенным Knowles et al.

Также как и в экспериментах Knowles et al. использовался только оператор мутации с двумя изменениями [8]. Данный оператор выбирает два различных города в пути и инвертирует порядок городов между ними (включая два выбранных города).

Поколение ЭА состояло из 100 особей. Число вычислений ФП взято из статьи [8]. Для задач ran20, ran50, euc50 оно равно $5 \cdot 10^5$, а для задач euc100, kroB100 — $2 \cdot 10^6$.

Алгоритм ε-жадного Q-обучения применялся со следующими параметрами: $\alpha = 0,6$, $\gamma = 0,01$, $\varepsilon = 0,3$. Для второй версии предложенного подхода были выбраны параметры $\alpha = 0,6$, $\gamma = 0,1$, $p_1 = 10$, $p_2 = 500$. Были исследованы различные значения параметра p_2 в диапазоне от 0 до 1000 и наилучшие

Таблица 5 – Среднее (верх ячейки) и лучшее (низ ячейки) значение целевой ФП. Темным (светлым) цветом выделен первый (второй) результат

Задача	Лучшее	ЭА	ИО	Knowles	С EA+RL K	НС EA+RL K	С EA+RL J	НС EA+RL J
ran20	1.91	2.03	2.55	2.66	1.93	1.92	1.96	1.92
		1.91	2.54	2.54	1.91	1.91	1.91	1.91
ran50	2.04	2.63	2.30	2.32	2.49	2.42	2.55	2.29
		2.34	2.13	2.18	2.19	2.14	2.29	2.06
euc50	5.03	5.62	5.72	5.78	5.51	5.49	5.51	5.48
		5.37	5.69	5.69	5.37	5.37	5.37	5.37
euc100	7.12	8.27	7.98	7.97	8.14	8.09	8.28	8.09
		7.96	7.85	7.79	7.91	7.95	8.01	7.90
kroB100	22141	23296.24	22529.20	22546.10	22952.11	22776.89	23161.56	22391.01
		22509	22217	22141	22432	22243	22611	22139

результаты были получены при значении параметра $p_2 = 500$. Результаты, полученные при данном значении параметра значительно лучше, чем при значениях параметра p_2 в диапазонах 0–400 и 600–1000.

3.5.2. Результаты экспериментов

Результаты экспериментального сравнения метода EA+RL с другими методами представлены в таблице 5. Для каждой задачи коммивояжера среднее значение представлено вверху ячейки, а лучшее полученное значение представлено внизу ячейки. В первой колонке содержится название задачи. Во второй колонке содержится наилучшее известное решение. Третья колонка содержит результаты применения традиционного ЭА (ЭА). В следующих двух колонках содержатся результаты применения метода имитации отжига (ИО) и подхода Knowles et al. (Knowles) взятые из статьи [8]. Следующие две колонки содержат результаты применения метода EA+RL с ε-жадным Q-обучением в стационарной среде (С EA+RL K) и второй версией предложенного алгоритма обучения с подкреплением в нестационарной среде (НС EA+RL K). В обоих методах использовались вспомогательные критерии, предложенные Knowles et al. Следующая колонка содержит результаты метода EA+RL с ε-жадным Q-обучением (С EA+RL J) с использованием двух вспомогательных критериев, предложенных Jähne et al. Последняя колонка содержит результаты метода EA+RL со второй версией предложенного алгоритма (НС EA+RL J) с использованием тех же двух вспомогательных критериев.

Первый и второй средние результаты выделены темно-зеленым и светло-зеленым цветами соответственно. Первый и второй лучшие результаты выделены оранжевым и светло-оранжевым цветами соответственно. Среднеквадратичное отклонение среднего значения ФП составило около 0.8%.

Рассмотрим результаты применения метода EA+RL с использованием второй версии предложенного алгоритма и вспомогательных критериев, предложенных Jähne et al. Данный метод превосходит метод EA+RL с использованием второй версии предложенного алгоритма и вспомогательных критериев, предложенных Knowles et al. на четырех задачах из пяти. На задаче euc100 результаты этих подходов одинаковы. Рассматриваемый метод превосходит метод Knowles et al. на четырех задачах из пяти. Для задачи euc100, метод, предложенный Knowles et al., дал лучший результат. Рассматриваемый метод превзошел все остальные исследуемые подходы на всех пяти задачах.

Таким образом, использование вспомогательных критериев, предложенных Jähne et al., в методе EA+RL с использованием второй версии предложенного алгоритма наиболее эффективно. Применение данного подхода дает наилучшие результаты среди всех рассмотренных алгоритмов на большинстве задач.

3.6. Эксперименты с методом MOEA+RL

В данном разделе описаны экспериментальные исследования и результаты применения метода MOEA+RL для решения задачи коммивояжера.

3.6.1. Описание экспериментов

Рассматривались задачи коммивояжера, решаемые в работах Jensen [10] и Jähne et al. [9]. Условия задач были взяты с сайта TSPLIB [1]. В рассматриваемых задачах было от 100 до 1002 городов. Число в названии задачи соответствует числу городов.

Использовались операторы мутации и кроссовера из статей [9, 10]. Была применена эвристика 2-opt, используемая при решении задач коммивояжера, также применяемая в [9, 10]. Поколение многокритериального ЭА состояло из 100 особей. Число вычислений ФП для каждой задачи вычислялось по формуле из статьи [9]: $E(N) = \sqrt{N^3} \times 15$, где N — число городов.

[1] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

В методе MOEA+RL и в методе, предложенном Jensen, на каждом шаге многокритериального ЭА оптимизировался целевой и один из вспомогательных критериев.

В алгоритме ε-жадного Q-обучения использовались следующие параметры: $\alpha = 0,6$, $\gamma = 0,01$, $\varepsilon = 0,3$. Для второй версии предложенного подхода были выбраны параметры $\alpha = 0,6$, $\gamma = 0,1$, $p_1 = 10$, $p_2 = 10$. Были исследованы различные значения параметра p_2 в диапазоне от 0 до 100. Наилучшие результаты были получены при значении параметра $p_2 = 10$. Различие в значениях параметра p_2 в методе EA+RL и MOEA+RL можно объяснить различным числом вычислений ФП, и, как следствие, различным числом итераций алгоритма.

3.6.2. Результаты экспериментов

Результаты применения метода MOEA+RL представлены в таблице 6. В первой колонке содержится название задачи. Во второй колонке содержится наилучшее известное решение. В следующих четырех колонках содержатся результаты применения методов MOEA+RL со второй версией предложенного алгоритма (HC MOEA+RL), MOEA+RL с ε-жадным Q-обучением (C MOEA+RL), Jähne et al. [9] (Jähne) и Jensen [10] (Jensen-Jähne). Во всех этих методах использовались два вспомогательных критерия, предложенных Jähne et al. Последняя колонка содержит результаты метода Jensen, с использованием десяти вспомогательных критериев, предложенных Jensen, как в экспериментах статьи [10].

Для каждой задачи коммивояжера среднее значение представлено вверху ячейки, а лучшее полученное значение представлено внизу ячейки. Первый и второй средние результаты выделены темно-зеленым и светло-зеленым цветами соответственно. Первый и второй лучшие результаты выделены оранжевым и светло-оранжевым цветами соответственно. Среднеквадратичное отклонение среднего значения ФП составило около 0.05%.

Сравним метод MOEA+RL с использованием второй версии предложенного алгоритма с остальными методами. Результаты сравнения представлены в таблице 7. В данной таблице содержится число задач, на которых результаты данного метода лучше, хуже, или совпадают с результатами, полученными при использовании других алгоритмов. Метод MOEA+RL с использованием

Таблица 6 – Среднее (верх ячейки) и лучшее (низ ячейки) значение целевой ФП. Темным (светлым) цветом выделен первый (второй) результат

Задача	Лучшее	НС MOEA+RL	С MOEA+RL	Jähne	Jensen-Jähne	Jensen
kroB100	22141	22144	22145	22150	22158	22155
		22139	22139	22139	22139	22139
kroD100	21294	21342	21353	21344	21349	21347
		21294	21294	21294	21294	21294
kroE100	21294	22093	22095	22169	22095	22100
		22068	22068	22068	22068	22068
eil101	629	641.39	641.84	641.50	641.59	641.95
		640	640	640	640	640
pr124	59030	59030	59030	59030	59032	59052
		59030	59030	59030	59030	59030
bier127	118282	118324	118394	118387	118408	118394
		118293	118293	118293	118293	118293
pr136	96772	96975	97000	96980	97193	97063
		96785	96835	96795	96785	96835
kroA150	26524	26540	26558	26533	26557	26558
		26524	26524	26524	26524	26554
kroB150	26130	26153	26166	26170	26166	26174
		26127	26127	26127	26127	26127
pr152	73682	73693	73702	73904	73820	73821
		73683	73683	73687	73683	73820
pr439	107217	107675	107677	107748	108035	107743
		107241	107248	107301	107258	107248
rat575	6773	6869	6872	6874	6863	6877
		6833	6824	6847	6835	6826
pr1002	259045	263158	263318	263425	263184	263189
		261444	261970	261231	262023	260971

Таблица 7 – Сравнение метода MOEA+RL с использованием второй версии предложенного алгоритма с остальными методами

	С MOEA+RL	Jähne	Jensen-Jähne	Jensen
Лучше	12	11	12	13
Одинаково	1	1	0	0
Хуже	0	1	1	0

второй версии предложенного алгоритма превосходит остальные алгоритмы на большинстве рассмотренных задач.

В соответствии с множественным знаковым тестом [35], метод MOEA+RL с использованием второй версии предложенного алгоритма различим с остальными методами с уровнем статистической значимости $\alpha = 0.05$. Таким образом метод MOEA+RL с применением второй версии предложен-

ного алгоритма с использованием вспомогательных критериев, предложенных Jähne et al. является наиболее эффективным на рассмотренном множестве задач коммивояжера.

3.7. Выводы по главе 3

Рассмотрена задача коммивояжера и существующие методы ее решения при помощи ЭА с использованием вспомогательных критериев. Приведены результаты решения задачи коммивояжера при помощи методов EA+RL и MOEA+RL с использованием алгоритма обучения с подкреплением в стационарной среде и второй версии предложенного алгоритма обучения с подкреплением в нестационарной среде, описанной в разделе 2.6. Исследовано использование различных вспомогательных критериев. Приведено сравнение с существующими методами. Из результатов экспериментов можно видеть, что применение предложенного алгоритма обучения с подкреплением в методах EA+RL и MOEA+RL дает лучшие результаты, чем методы EA+RL и MOEA+RL с использованием алгоритма обучения с подкреплением в стационарной среде и существующие алгоритмы решения задачи коммивояжера при помощи вспомогательных критериев. Также отметим, что результаты экспериментов подтверждают эффективность использования вспомогательных критериев, предложенных Jähne et. al, для решения задачи коммивояжера. Современные исследования направлены на улучшение подхода Jähne et al. Однако, как показано в настоящей работе, метод, предложенный Jensen, в котором на каждом шаге оптимизируется целевая ФП и одна из вспомогательных, при использовании подходящего метода выбора вспомогательной ФП, может быть более эффективным.

ЗАКЛЮЧЕНИЕ

В работе предложен алгоритм обучения с подкреплением, применимый в условиях нестационарности, заключающейся в изменении свойств вспомогательных критериев. Данный алгоритм был применен в методе EA+RL для решения модельной задачи. Результаты применения предложенного алгоритма превосходят результаты использования алгоритмов обучения в стационарной среде: ε-жадного Q-обучения и отложенного Q-обучения, и алгоритма обучения с подкреплением в нестационарной среде RLCD. Статистическая значимость полученных результатов была проверена при помощи теста суммы рангов Уилкоксона.

Предложенный алгоритм был применен в методах EA+RL и MOEA+RL для решения задачи коммивояжера. Было проведено сравнение с тремя существующими методами решения задачи коммивояжера при помощи эволюционного алгоритма с использованием вспомогательных критериев. Отметим, что метод EA+RL требует существенно большего числа вычислений ФП, чем метод MOEA+RL, что подтверждает, что в некоторых случаях неявной оптимизации целевого критерия, осуществляемой в методе EA+RL, недостаточно. Из результатов эксперимента следует, что применение предложенного алгоритма обучения с подкреплением эффективно для решения задачи коммивояжера и позволяет добиться наилучших результатов среди рассмотренных методов. Также в данном исследовании было показано, что при использовании эффективного метода выбора вспомогательного критерия, метод в котором оптимизируется целевой критерий и один из вспомогательных может быть эффективнее метода декомпозиции.

Таким образом была экспериментально проверена эффективность использования предложенного алгоритма обучения с подкреплением в методе EA+RL и MOEA+RL. Разработанный алгоритм удовлетворяет всем поставленным требованиям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 *Скобцов Ю. А.* Основы эволюционных вычислений. — Донецк : ДонНТУ, 2008.

2 *Eiben A. E.*, *Smith J. E.* Introduction to Evolutionary Computing. — Berlin, Heidelberg, New York : Springer-Verlag, 2007.

3 *Mitchell M.* An Introduction to Genetic Algorithms. — Cambridge, MA : MIT Press, 1996. — 221 p.

4 A Fast Elitist Multi-Objective Genetic Algorithm: NSGA-II / K. Deb [et al.] // Transactions on Evolutionary Computation. — 2000. — Vol. 6. — P. 182–197.

5 *Neumann F.*, *Wegener I.* Can Single-Objective Optimization Profit from Multiobjective Optimization? // Multiobjective Problem Solving from Nature. — Springer Berlin Heidelberg, 2008. — P. 115–130. — (Natural Computing Series).

6 *Neumann F.*, *Wegener I.* Minimum Spanning Trees Made Easier via Multi-objective Optimization // Natural Computing. — 2006. — Vol. 5, no. 3. — P. 305–319.

7 Multi-objectivization of reinforcement learning problems by reward shaping / T. Brys [et al.] // 2014 International Joint Conference on Neural Networks. — 2014. — P. 2315–2322.

8 *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. — Springer-Verlag, 2001. — P. 269–283.

9 *Jähne M.*, *Li X.*, *Branke J.* Evolutionary Algorithms and Multi-objectivization for the Travelling Salesman Problem // Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. — New York, NY, USA : ACM, 2009. — P. 595–602. — (GECCO ’09).

10 *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. — Vol. 3, no. 4. — P. 323–347.

11 *Lochtefeld D. F.*, *Ciarallo F. W.* Deterministic Helper-Objective Sequences Applied to Job-Shop Scheduling // Proceedings of Genetic and Evolutionary Computation Conference. — ACM, 2010. — P. 431–438.

12 *Sutton R. S.*, *Barto A. G.* Reinforcement Learning: An Introduction. — Cambridge, MA, USA : MIT Press, 1998.

13 *Gosavi A.* Reinforcement Learning: A Tutorial Survey and Recent Advances // INFORMS Journal on Computing. — 2009. — Vol. 21, no. 2. — P. 178–192.

14 *Petrova I.*, *Buzdalova A.*, *Buzdalov M.* Improved Helper-Objective Optimization Strategy for Job-Shop Scheduling Problem // Proceedings of the International Conference on Machine Learning and Applications. Vol. 2. — IEEE Computer Society, 2013. — P. 374–377.

15 *Николенко С. И.*, *Тулупьев А. Л.* Самообучающиеся системы. — М., 2009.

16 *Buzdalov M.*, *Buzdalova A.*, *Petrova I.* Generation of Tests for Programming Challenge Tasks Using Multi-Objective Optimization // Proceedings of Genetic and Evolutionary Computation Conference Companion. — ACM, 2013. — P. 1655–1658.

17 *Buzdalova A.*, *Buzdalov M.* Adaptive Selection of Helper-Objectives with Reinforcement Learning // Proceedings of the International Conference on Machine Learning and Applications. Vol. 2. — IEEE Computer Society, 2012. — P. 66–67.

18 *Afanasyeva A.*, *Buzdalov M.* Optimization with Auxiliary Criteria using Evolutionary Algorithms and Reinforcement Learning // Proceedings of 18th International Conference on Soft Computing MENDEL 2012. — Brno, Czech Republic, 2012. — P. 58–63.

19 *Buzdalov M.*, *Buzdalova A.*, *Shalyto A.* A First Step towards the Runtime Analysis of Evolutionary Algorithm Adjusted with Reinforcement Learning // Proceedings of the International Conference on Machine Learning and Applications. Vol. 1. — IEEE Computer Society, 2013. — P. 203–208.

20 *Buzdalov M.*, *Buzdalova A.* OneMax Helps Optimizing XdivK: Theoretical Runtime Analysis for RLS and EA+RL // Proceedings of Genetic and Evolutionary Computation Conference (Companion). — 2014. — P. 201–202.

21 *Kaelbling L. P.*, *Littman M. L.*, *Moore A. W.* Reinforcement Learning: A Survey // Journal of Artificial Intelligence Research. — 1996. — Vol. 4. — P. 237–285.

22 *Basso E. W.*, *Engel P. M.* Reiforcement learning in non-stationary continuous time and space scenarios // Proceedings of the VII Brasilian Meeting on Artificial Intelligence ENIA. — Bento Concalves, Brazil : SBC Press, 2009. — P. 687–696.

23 Multiple model-based reinforcement learning / K. Doya [и др.] // Neural Computation. — 2002. — Т. 14. — С. 1347—1369.

24 *Granmo O.-C.*, *Berg S.* Solving Non-Stationary Bandit Problems by Random Sampling from Sibling Kalman Filters // IEA/AIE (3). — 2010. — P. 199–208. — DOI: 10.1007/978-3-642-13033-5_21.

25 *Fu H.*, *Lewis P. R.*, *Yao X.* A Q-learning Based Evolutionary Algorithm for Sequential Decision Making Problems // In Proceedings of the Workshop "In Search of Synergies between Reinforcement Learning and Evolutionary Computation"at the 13th International Conference on Parallel Problem Solving from Nature (PPSN). — VUB Artificial Intelligence Lab, 2014.

26 Dealing with Non-stationary Environments Using Context Detection / B. C. D. Silva [et al.] // Proceedings of the 23rd International Conference on Machine Learning. — ACM Press, 2006. — P. 217–224.

27 *Arkhipov V.*, *Buzdalov M.*, *Shalyto A.* Worst-Case Execution Time Test Generation for Augmenting Path Maximum Flow Algorithms using Genetic Algorithms // Proceedings of the International Conference on Machine Learning and Applications. Vol. 2. — IEEE Computer Society, 2013. — P. 108–111.

28 PAC Model-free Reinforcement Learning / A. L. Strehl [et al.] // Proceedings of the 23rd International Conference on Machine Learning. — 2006. — P. 881–888.

29 *R Core Team.* R: A Language and Environment for Statistical Computing / R Foundation for Statistical Computing. — 2013. — URL: http://www.R-project.org/.

30 Watchmaker Framework for Evolutionary Computation. — URL: http://watchmaker.uncommons.org.

31 A Method to Control Parameters of Evolutionary Algorithms by Using Reinforcement Learning / Y. Sakurai [et al.] // Signal-Image Technology and Internet-Based Systems (SITIS), 2010 Sixth International Conference on. — 2010. — P. 74–79. — DOI: 10.1109/SITIS.2010.22.

32 The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics) / D. L. Applegate [et al.]. — Princeton, NJ, USA : Princeton University Press, 2007.

33 *Lochtefeld D. F.*, *Ciarallo F. W.* An Analysis of Decomposition Approaches in Multi-objectivization via Segmentation // Appl. Soft Comput. — 2014. — Vol. 18. — P. 209–222.

34 *Buzdalov M.*, *Petrova I.*, *Buzdalova A.* NSGA-II Implementation Details May Influence Quality of Solutions for the Job-Shop Scheduling Problem // Proceedings of Genetic and Evolutionary Computation Conference (Companion). — 2014. — P. 1445–1446.

35 A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms / J. Derrac [et al.] // Swarm and Evolutionary Computation. — 2011. — Vol. 1, no. 1. — P. 3–18.