

Раздел III. Алгоритмы обработки информации

УДК 004.023

DOI 10.18522/2311-3103-2024-2-221-228

М.В. Пикалов, А.М. Письмеров

НАСТРОЙКА ПАРАМЕТРОВ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ПРИ ПОМОЩИ АНАЛИЗА ЛАНДШАФТА ФУНКЦИИ ПРИСПОСОБЛЕННОСТИ И МАШИННОГО ОБУЧЕНИЯ

*Выбор значений параметров в эволюционных алгоритмах сильно влияет на их производительность. Многие популярные методы настройки параметров ограничены максимальным числом вычислений целевой функции для поиска хорошего набора значений параметров. Недавно был предложен подход к выбору алгоритмов для решения оптимизационных задач, использующий анализ ландшафта функции приспособленности и машинное обучение для выбора оптимального алгоритма решения задачи на основе особенностей ее ландшафта. Подобное применение анализа ландшафта функции приспособленности мотивирует на дальнейшие исследования, в частности, применительно к настройке параметров эволюционных алгоритмов. Использование признаков ландшафта функции приспособленности позволяет выявлять похожие задачи и использовать данные о настройке параметров, полученные при тестировании на эталонных задачах, что значительно снижает число необходимых вычислений целевой функции при настройке. В этой работе на примере генетического алгоритма $(1 + (\lambda, \lambda))$ рассматривается подход к автоматическому выбору параметров с использованием анализа ландшафта целевой функции и машинного обучения. В предлагаемом решении оцениваются особенности ландшафта целевой функции поставленной задачи оптимизации и предлагаются оптимальные значения параметров алгоритма с помощью нейронной сети. Данная сеть была обучена на наборе данных об особенностях ландшафта, выраженных в виде числовых признаков и соответствующих им оптимальных наборов параметров алгоритма. В отличие от подходов к автоматическому выбору алгоритмов оптимизации для конкретной задачи, в данной работе рассматривается задача регрессии параметров алгоритма вместо проблемы классификации наиболее подходящего алгоритма из заданного набора. Результаты экспериментов на различных конфигурациях задачи *W-model*, а также на задаче *MAX-3SAT* показывают, что предлагаемый подход к автоматическому выбору параметров с учетом ландшафта целевой функции может помочь определить подходящие значения статических параметров генетического алгоритма $(1 + (\lambda, \lambda))$, так как алгоритм с предложенными значениями параметров превосходит другие рассмотренные варианты $(1 + (\lambda, \lambda))$ GA, в среднем требуя меньше вычислений целевой функции для нахождения оптимума, чем остальные рассмотренные алгоритмы.*

Эволюционные алгоритмы; настройка параметров; анализ ландшафта.

M.V. Pikalov, A.M. Pismirov

GENETIC ALGORITHM PARAMETER TUNING USING EXPLORATORY LANDSCAPE ANALYSIS AND MACHINE LEARNING

The choice of parameter values in evolutionary algorithms greatly affects their performance. Many popular parameter tuning methods are constrained by the maximum number of fitness function evaluations to find a good set of parameter values. Recently, an approach to algorithm selection for optimization problems has been proposed, which uses the analysis of the fitness function landscape and machine learning to select the optimal algorithm based on the characteristics of its landscape. Such application of fitness landscape analysis motivates further research, particularly in the context of parameter tuning in evolutionary algorithms. The use of landscape features allows for the identification of similar problems and the use of parameter tuning data obtained from testing on benchmark problems, significantly reducing the number of required fitness function evaluations during tuning. This work considers an approach to automatic parameter selection using landscape analysis of the objective function and machine learning, using

a (1 + (λ, λ)) genetic algorithm as an example. The proposed solution evaluates the characteristics of the landscape of the optimization problem's objective function and suggests optimal parameter values for the algorithm using a neural network. This network was trained on a dataset of landscape features expressed as numerical features and their corresponding optimal algorithm parameter sets. In contrast to approaches for automatic algorithm selection for a specific problem, this work addresses the problem of regressing algorithm parameters instead of classifying the most suitable algorithm from a given set. The results of experiments on different configurations of the W-model problem, as well as on the MAX-3SAT problem, show that the proposed approach to automatic parameter selection considering the landscape of the objective function can help determine appropriate values for the static parameters of the (1 + (λ, λ)) genetic algorithm. The algorithm with the proposed parameter values outperforms other considered (1 + (λ, λ)) GA options on average, requiring fewer evaluations of the objective function to find the optimum compared to the other algorithms considered.

Evolutionary algorithms; parameter tuning; landscape analysis.

Введение. Развитие методов выбора параметров в эволюционных алгоритмах является необходимым условием для их успешного применения на практике [1]. Существует несколько подходов к решению этой проблемы, которые обычно называются настройкой параметров и управлением параметрами.

Анализ ландшафта приспособленности [2] активно изучается как инструмент, который может помочь извлечь знания из экземпляров задач, позволяя настраивать используемые метаэвристики для рассматриваемой оптимизационной задачи. Исследовательский анализ ландшафта [3] рассматривает характеристики ландшафта приспособленности, которые могут быть извлечены из промежуточных решений и их значений приспособленности, что позволяет получить некоторое понимание проблемы, даже в условиях "черного ящика" [4, 5].

Исследовательский анализ ландшафта имеет многочисленные приложения [6]. Одно из перспективных направлений исследований – автоматический выбор алгоритма с помощью анализа ландшафта и машинного обучения [7], а также динамический выбор алгоритма [8].

В этой статье мы представляем наши первые шаги в разработке подхода к автоматическому выбору параметров, используя анализ ландшафта и нейронные сети. В предложенном подходе мы оцениваем особенности ландшафта задачи и предлагаем оптимальные значения параметров, используя нейронную сеть, обученную на наборе данных особенностей ландшафта и соответствующих оптимальных значений параметров. Мы рассматриваем проблему регрессии параметров алгоритма, а не классификации задачи по наилучшему подходящему алгоритму.

Для этого исследования мы собрали набор данных оптимальных значений параметров для генетического алгоритма (1 + (λ, λ)) [9] с четырьмя статическими параметрами на нескольких экземплярах тестовой задачи W-model [10]. Мы вычислили особенности ландшафта для тех же экземпляров задач, используя пакет flacco в R [11]. Используя нейронную сеть, обученную на этом наборе данных, мы можем рекомендовать значения параметров для генетического алгоритма (1 + (λ, λ)) для разных экземпляров задач оптимизации, для которых мы можем вычислить особенности ландшафта их функции приспособленности.

Экспериментальные результаты на различных экземплярах задачи W-model показывают, что предложенный подход, основанный на анализе ландшафта, может помочь определить хорошие значения для статических параметров генетического алгоритма (1 + (λ, λ)). Основным свидетельством этого является то, что алгоритм со значениями параметров, предложенными нашим методом, превосходит другие рассматриваемые варианты (1 + (λ, λ)) GA.

Генетический алгоритм (1 + (λ, λ)). Генетический алгоритм (1 + (λ, λ)), предложенный в [12], представляет собой двухэтапный генетический алгоритм, который показал свою эффективность на нескольких теоретических и практических задачах. Алгоритм пытается исследовать пространство поиска, используя высокие скорости мутаций на первом этапе. На втором этапе он применяет механизм скрещивания для противодействия

возможным негативным последствиям высокой скорости мутаций для отдельных особей. Выбор параметров в $(1 + (\lambda, \lambda))$ GA существенно влияет на его производительность и мотивирует исследования по его обобщению и, в частности, по настройке параметров.

В данной работе рассматривается генетический алгоритм $(1 + (\lambda, \lambda))$ с четырьмя параметрами $\lambda_1, \lambda_2, k, c$, аналогичный рассмотренному в [13]. Сам алгоритм работает следующим образом:

- ◆ на первом этапе каждой итерации алгоритм создает λ_1 мутантных особей, применяя оператор мутации со скоростью мутации k/n , где n – размер задачи;
- ◆ лучший мутант выбирается для скрещивания с родительской особью;
- ◆ на втором этапе создаются λ_2 особи с помощью оператора скрещивания, который берет биты у мутантной особи с вероятностью c ;
- ◆ лучшая особь заменяет родителя, если ее значение приспособленности равно или больше родительского;
- ◆ процесс повторяется до тех пор, пока не будет найден оптимум.

Тестовые задачи. В этой статье мы используем две тестовые задачи для разработки и оценки предлагаемого подхода: задачу W-model и задачу о выполнимости булевой формулы в максимизационной постановке MAX-3SAT.

Задача W-model[10] – это параметризованная задача оптимизации, основанная на известной задаче OneMax: поиске скрытого оптимума на основе количества совпадающих битов в двоичной строке.

W-model вводит несколько настраиваемых слоев, которые влияют на ландшафт целевой функции задачи и, таким образом, делают ее пригодной для сравнительного анализа. Настраиваемые слои задачи W-модели представляют из себя:

- ◆ нейтральность – вводит области с одинаковыми значениями функции пригодности [14];
- ◆ эпистаз – вводит зависимость между различными отдельными генами особей [15];
- ◆ жесткость – вводит так называемые горы и долины в ландшафт целевой функции [16];
- ◆ фиктивность – вводит переменные, не влияющие на значение приспособленности особей.

W-model позволяет нам создавать экземпляры задач с различными свойствами ландшафта функции пригодности, что важно для сбора набора данных о производительности алгоритма оптимизации, необходимого в нашем подходе.

Также, мы рассматриваем задачу о выполнимости булевой формулы в максимизационной постановке MAX-3SAT[17]. Эта задача изучается в эволюционных вычислениях как теоретически, так и практически, и было показано, что $(1 + 1)$ ЭА[18] и $(1 + (\lambda, \lambda))$ GA показывают время выполнения, подобное тому, которое они показывают на задаче OneMax [19].

Основная задача в задаче о выполнимости булевой формулы в максимизационной постановке состоит в том, чтобы найти такую битовую строку $x = x_i, i = 1, \dots, n$, чтобы максимизировать количество удовлетворяемых выражений в булевой формуле в КНФ (конъюнктивной нормальной форме) с тремя переменными в каждом выражении.

Обучающий набор данных. Набор обучающих данных состоит из среднего времени выполнения (числа вычислений целевой функции, необходимого для нахождения оптимума) параметризованного $(1 + (\lambda, \lambda))$ GA с разными значениями параметров на разных экземплярах задачи W-model.

Для $(1 + (\lambda, \lambda))$ GA рассматриваются следующие параметры:

$$\lambda_1 \in [2, 3, 4, \dots, 9]$$

$$\lambda_2 \in [2, 3, 4, \dots, 9]$$

$$k \in [1, 3, 5, 7, 9]$$

$$c \in [0.01, 0.03, 0.05, 0.07, 0.09]$$

Мы вычислили среднее время выполнения $(1 + (\lambda, \lambda))$ GA со всеми возможными комбинациями этих параметров на экземплярах задачи W-model со следующими параметрами:

$n \in [8, 16, 32, 64, 128]$
 $dummy \in [1, 1 - 1/n, 1 - 2/n, \dots, 0.8]$
 $neutrality \in [0, 1, 2, \dots, n \times 0.2]$
 $epistasis \in [0, 2, 3]$

Для каждой комбинации параметров $(1 + (\lambda, \lambda)) GA$ $\lambda_1, \lambda_2, k, c$, и параметров задачи W-model $n, dummy, neutrality, epistasis$ 25 раз запускается генетический алгоритм $(1 + (\lambda, \lambda))$. Затем выбирается набор параметров с наименьшим средним временем выполнения в качестве наилучшего набора параметров для конкретного экземпляра задачи.

Для каждого экземпляра задачи W-model из набора данных о производительности мы использовали флассо для вычисления 25 отдельных векторов из 35 характеристик ландшафта функции приспособленности. Каждый вектор признаков рассчитывается из $n/4$ случайных особей, что следует учитывать при анализе эффективности предложенного подхода.

Комбинируя признаки с данными о производительности мы получаем набор данных лучших параметров $(1 + (\lambda, \lambda)) GA$ для определенных признаков ландшафта целевой функции, вычисленных флассо, и размера задачи $n: \{n, features\} \Rightarrow \{\lambda_1, \lambda_2, k, c\}$. Далее мы можем использовать эти данные для обучения нейронной сети, которая сможет предлагать определенные значения параметров для заданного набора признаков ландшафта функции приспособленности.

Обучение нейронной сети. Мы обучили FNN (нейронную сеть с прямой связью) с двумя скрытыми слоями с 32 и 16 узлами соответственно. Также рассматривались и другие архитектуры: сети с количеством скрытых слоев от 1 до 4, в которых было от 4 до 32 нейронов, результаты тестирования и валидации которых оказались хуже выбранной.

Были рассмотрены различные алгоритмы оптимизации, такие как Adam, SGD (стохастический градиентный спуск) и BFGS с ограниченной памятью (Бройден-Флетчер-Гольдфарб-Шанно). Алгоритм Adam показал наименьшую среднеквадратичную ошибку, поэтому он был выбран для окончательного обучения модели. Мы предоставляем вычисленные характеристики ландшафта функции приспособленности экземпляра задачи в качестве входных данных и получаем рекомендуемые значения параметров алгоритма оптимизации для $(1 + (\lambda, \lambda)) GA$ в качестве выходных данных.

Эксперименты. В этом разделе мы представляем наши эмпирические данные об эффективности предложенного подхода. Мы оцениваем время работы алгоритмов, т.е. количество оценок функции приспособленности, необходимых для нахождения оптимального решения.

Для каждой рассматриваемой тестовой задачи мы сравниваем производительность алгоритма с предложенными нейронной сетью параметрами, который мы называем $(1 + (\lambda, \lambda)) GA$ tuned, со следующими вариантами $(1 + (\lambda, \lambda)) GA$, рассмотренными в [20]:

$(1 + (\lambda, \lambda)), \lambda = 4$: $(1 + (\lambda, \lambda)) GA$ с выбором параметров по умолчанию

$\lambda_2 = \lambda_1, k = \lambda_1, c = 1/\lambda_1$ для $\lambda_1 = 4$;

$(1 + (\lambda, \lambda)), \lambda \leq 2 \log n$ и $(1 + (\lambda, \lambda)), \lambda \leq n$: $(1 + (\lambda, \lambda)) GA$ с динамическим λ , настроенный по правилу одной пятой с указанной верхней границей;

$(1 + (\lambda, \lambda)), \lambda \sim \text{pow}(2.5)$: $(1 + (\lambda, \lambda)) GA$ с λ , выбранным из степенного распределения с $\beta = 2.5$;

Мы также рассматриваем параметры в наборе обучающих данных, которые показывает наилучшую общую производительность среди всех вариантов выбора параметров. Эти значения параметров: $\lambda_1 = 2, \lambda_2 = 2, k = 8, c = 0.02$, мы обозначаем $(1 + (\lambda, \lambda)) GA$ с такими параметрами как $(1 + (\lambda, \lambda))$, single best. Для задачи W-model мы также рассмотрели эволюционный алгоритм (ЭА) (1+1) и алгоритм случайного локального поиска (RLS) в качестве дополнительной основы для сравнения алгоритмов.

Результаты на W-model. Сначала мы оценили производительность предложенного подхода на экземплярах W-model с размером задачи $n = 512$. Обратим внимание, что для обучающих данных мы использовали только экземпляры W-model с размером $n = 128$. Увеличение значения n также увеличивает количество различных значений в фиктивном слое и увеличивает верхнюю границу слоя нейтральности.

Примеры параметров для конфигураций W-model представлены в табл. 1. all-default представляет собой $(1 + (\lambda, \lambda))$ GA со статическими параметрами $\lambda_2 = \lambda_1, k = \lambda_1, c = 1/\lambda_1$ при $\lambda_1 = 4$, рекомендованными авторами алгоритма. all-singlebest – $(1 + (\lambda, \lambda))$ GA с параметрами, показавшими в среднем лучшую производительность среди всех рассмотренных наборов параметров на конфигурациях W-model из набора данных для обучения модели, то есть $\lambda_1 = 2, \lambda_2 = 2, k = 8, c = 0.02$. Данные значения параметров не меняются на различных экземплярах задачи. Каждая конфигурация W-model определяется параметрами: n – размерность, dum, neu, epi – параметры слоев dummy, neutrality, epistasis соответственно. Для экземпляров задачи W-model представлены рекомендуемые обученной нейронной сетью параметры.

Таблица 1

Параметры для W-model

Конфигурация задачи				Параметры $(1 + (\lambda, \lambda))$ GA			
n	dum	neu	epi	λ_1	λ_2	k	c
all-default				4	4	0.25	4
all-singlebest				2	2	0.02	8
256	0.8	50	1	4	2	0.071	9
256	0.95	0	0	3	5	0.052	1
512	0.90	80	1	5	4	0.012	4
512	0.99	25	2	4	5	0.069	9
256	1	0	3	3	3	0.034	3
512	0.85	0	2	5	7	0.045	2

Модель предлагает различные параметры $(1 + (\lambda, \lambda))$ GA для разных конфигураций задачи W-model. Анализируя зависимость предлагаемых параметров от конфигурации задачи можно заметить, что для больших значений размерности задачи n предлагаются большие значения параметров λ . Также при больших значениях параметра слоя нейтральности, модель рекомендует большие значения параметра k .

Результирующие показатели производительности показаны в табл. 2. Для каждого алгоритма приводится его средний ранг среди всех рассмотренных алгоритмов $Mean_r$, медиана ранга $Median_r$, стандартное отклонение ранга Std_r и средняя разница в числе вычислений функции приспособленности $Mean_d$ для алгоритма по сравнению с $(1 + (\lambda, \lambda))$ GA tuned, с учетом вычислений для сбора выборки для анализа ландшафта. Отрицательные значения разницы указывают на то, что конкретный алгоритм работает лучше, чем $(1 + (\lambda, \lambda))$ GA tuned, а положительные значения указывают на то, что алгоритм работает хуже.

Таблица 2

Сравнение алгоритмов на задаче W-model с размерностью 512

Алгоритм	$Mean_r$	$Median_r$	Std_r	$Mean_d$
RLS	2.93	3	1.83	-154.48
$(1 + (\lambda, \lambda))$, tuned	3.13	3	1.52	0
$(1 + (\lambda, \lambda))$, $\lambda \leq n$	3.40	3	1.91	204.54
$(1 + (\lambda, \lambda))$, $\lambda \leq 2 \log n$	3.86	4	1.58	247.22
$(1 + (\lambda, \lambda))$, $\lambda \sim \text{pow}(2.5)$	4.53	5	1.20	399.10
$(1 + (\lambda, \lambda))$, $\lambda = 4$	5.09	5	2.14	706.25
$(1 + 1)$ EA	6.32	6	1.35	886.95
$(1 + (\lambda, \lambda))$, single best	6.56	7	1.48	1272.10

Из результатов видно, что в среднем $(1 + (\lambda, \lambda))$ GA tuned является лучшим из всех рассмотренных вариантов $(1 + (\lambda, \lambda))$ GA. В большинстве случаев разница в производительности больше, чем количество оценок, необходимых для вычисления характеристик ландшафта функции приспособленности, и она увеличивается с размером задачи n . Вер-

сии $(1 + (\lambda, \lambda))$ GA с динамической настройкой параметров улучшают свою производительность с увеличением размера задачи, что может означать, что им либо нужно время, чтобы найти правильные значения параметров, либо они значительно выигрывают от изменения λ на разных этапах оптимизации. Тем не менее, $(1 + (\lambda, \lambda))$ GA tuned имеет лучшую производительность, чем они, что говорит о том, что он может выбирать хорошие значения параметров перед запуском оптимизации.

Кроме того, существует значительная разница в производительности $(1 + (\lambda, \lambda))$ GA tuned и $(1 + (\lambda, \lambda))$ GA single best. На $n = 512$ алгоритм single best показывает худшую производительность из всех, в то время как $(1 + (\lambda, \lambda))$ GA tuned имеет отличную производительность. Это наблюдение заставляет нас поверить, что нейронная сеть способна обучаться выбору параметров в зависимости от особенностей ландшафта целевой функции, а не просто выбирает лучшие в среднем параметры из обучающего набора данных.

Также были изучены конкретные диапазоны параметров W-model, чтобы понять преимущества и недостатки обученной модели и предлагаемого метода. Разделив диапазоны параметров на интервалы, был проведен сравнительный анализ производительности рассмотренных алгоритмов на соответствующих конфигурациях задач W-model на основе средних рангов алгоритмов и средней разницы числа вычислений функций приспособленности. В среднем предложенный подход имеет лучшую производительность на всех рассмотренных интервалах значений параметров.

Наконец, отметим, что алгоритмы $(1 + 1)$ EA начинают заметно снижать производительность с увеличением размерности задачи, а алгоритм RLS не в состоянии найти оптимум на нескольких экземплярах задачи W-model. Возможно, эти результаты объясняются тем, что параметры, предлагаемые нейронной сетью для алгоритма $(1 + (\lambda, \lambda))$, позволяют ему повторять поведение этих простых алгоритмов, тем самым быстрее решая конкретные экземпляры задач, на которых $(1 + (\lambda, \lambda))$ GA с его типичными значениями параметров уступает по производительности алгоритмам RLS и $(1 + 1)$ EA.

Результаты на MAX-3SAT. Также, мы оценили предложенный подход на задаче MAX-3SAT, которая также не использовалась для сбора обучающего набора данных. Выражения булевой формулы в опытах генерировались случайным образом и независимо друг от друга. Всего было рассмотрено около 1000 случайных конфигураций с размерностью задачи $n \in [256, 512]$. Результаты для задачи MAX-3SAT представлены в табл. 3.

Таблица 3

Сравнение алгоритмов на задаче MAX-3SAT

Алгоритм	$Mean_r$	$Median_r$	Std_r	$Mean_d$
$(1 + (\lambda, \lambda))$, tuned	2.45	2	1.10	0
$(1 + (\lambda, \lambda))$, $\lambda \leq n$	2.57	2	1.21	68.22
$(1 + (\lambda, \lambda))$, $\lambda \leq 2 \log n$	2.79	3	1.29	104.54
$(1 + (\lambda, \lambda))$, $\lambda \sim \text{pow}(2.5)$	3.82	4	1.21	209.10
$(1 + (\lambda, \lambda))$, $\lambda = 4$	4.90	5	0.98	299.50
$(1 + (\lambda, \lambda))$, single best	5.07	5	1.32	355.52

Как в других результатах, предложенный $(1 + (\lambda, \lambda))$ GA tuned оказывается более эффективен, чем другие варианты $(1 + (\lambda, \lambda))$ GA. В отличие от задачи W-model, наблюдаются более устойчивые результаты и меньший разброс рангов. Это можно еще раз объяснить тем, что экземпляры MAX-3SAT с разной конфигурацией хоть и имеют разные характеристики ландшафта, но все же больше похожи друг на друга, чем разные конфигурации W-model.

Выводы. В этой статье мы изучаем подход к автоматической настройке параметров с использованием анализа ландшафта функции приспособленности и машинного обучения. Этот подход направлен на использование анализа ландшафта целевой функции для переноса знаний об оптимальных параметрах из экземпляров задач, оцененных ранее, на новые экземпляры задач с аналогичными характеристиками ландшафта. В результате это

позволяет нам пропустить этап обучения, обычно присутствующий в подходах к настройке, за счет использования данных, полученных на задачах с различными особенностями ландшафта.

Мы оценили предложенный подход на генетическом алгоритме $(1 + (\lambda, \lambda))$ с четырьмя параметрами и на нескольких задачах оптимизации. Мы собрали данные о производительности алгоритма на разных экземплярах задачи W-model с различными характеристиками ландшафта функции приспособленности и обучили нейронную сеть на этих данных. Сеть способна предложить хорошие значения параметров для различных задач оптимизации на основе особенностей их ландшафта. Кроме того, мы отмечаем значительную разницу в производительности между параметрами, предложенными нейронной сетью, и лучшими в среднем параметрами из обучающего набора данных, что говорит о том, что алгоритм машинного обучения способен извлекать зависимости между особенностями ландшафта и выбором параметров.

Исходя из наших результатов, мы с оптимизмом смотрим на то, что применение анализа ландшафта целевой функции и машинного обучения для настройки параметров может помочь нам разработать более эффективные инструменты для решения задачи выбора параметров. В дальнейшем мы планируем расширить этот подход к управлению параметрами путем переоценки характеристик ландшафта и рекомендуемых параметров во время процесса оптимизации.

Для будущих исследований у нас есть несколько целей. Прежде всего, важно дополнительно оценить предлагаемый подход на большем числе задач. Это может не только помочь нам лучше понять его преимущества и недостатки, но и предоставить больше данных для обучения, что всегда хорошо в машинном обучении. Кроме того, мы считаем важным изучить различные стратегии выборки для оценки характеристик ландшафта и их влияния на наши результаты. Наконец, разработка простого инструмента для настройки параметров с использованием нашего подхода сделает наши результаты легко доступными для других, а также значительно упростит сбор данных о производительности различных алгоритмов, что является важным требованием для эффективности нашего подхода.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Lobo F.J., Lima C.F., Michalewicz Z. (ed.). Parameter setting in evolutionary algorithms. Springer Science & Business Media, 2007, Vol. 54.
2. Mersmann O., Preuss M., Trautmann H. Benchmarking evolutionary algorithms: Towards exploratory landscape analysis, 2010.
3. Mersmann O. et al. Exploratory landscape analysis, *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 2011, pp. 829-836.
4. Lopez-Ibanez, M., Dubois-Lacoste J., Caceres L.P., Stutzle T., Birattari M. The irace package: Iterated racing for automatic algorithm configuration, *Operations Research Perspectives*, 2016, 3, pp. 43-58.
5. Hutter F., Hoos H.H., Leyton-Brown K. Sequential model-based optimization for general algorithm configuration, *Proceedings of Learning and Intelligent Optimization*. Springer, 2011, pp. 507-523.
6. Ochoa G., Malan K. Recent advances in fitness landscape analysis, *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2019, pp. 1077-1094.
7. Kerschke P., Trautmann H. Automated algorithm selection on continuous black-box problems by combining exploratory landscape analysis and machine learning, *Evolutionary computation*, 2019, pp. 99-127.
8. Janković A., Doerr C. Adaptive landscape analysis, *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2019, pp. 2032-2035.
9. Bassin A., Buzdalov M. The $(1 + (\lambda, \lambda))$ Genetic Algorithm for Permutations, *Proceedings of Genetic and Evolutionary Computation Conference Companion*. ACM, 2020, pp. 1669-1677.
10. Weise T., Wu Z. Difficult features of combinatorial optimization problems and the tunable w-model benchmark problem for simulating them, *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018, pp. 1769-1776.
11. Kerschke P., Trautmann H. The R-Package FLACCO for exploratory landscape analysis with applications to multi-objective optimization problems, *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2016, pp. 5262-5269.
12. Doerr B., Doerr C., Ebel F. From black-box complexity to designing new genetic algorithms, *Theoretical Computer Science*, 2015, Vol. 567, pp. 87-104.
13. Dang N., Doerr C. Hyper-parameter tuning for the $(1+(\lambda, \lambda))$ GA, *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019, pp. 889-897.

14. Kimura M. [et al.]. Evolutionary rate at the molecular level, *Nature*, 1968, Vol. 217, No. 5129, pp. 624-626.
15. Cordell H.J. Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans, *Human molecular genetics*, 2002, Vol. 11, No. 20, pp. 2463-2468.
16. Weise T. [et al.]. A tunable model for multiobjective, epistatic, rugged, and neutral fitness landscapes, *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, 2008, pp. 795-802.
17. Garay M.R. Computers and intractability. A guide to the theory of NP-completeness, 1979.
18. Beyer H.G. Evolution strategies, *Scholarpedia*, 2007, Vol. 2, No. 8, pp. 1965.
19. Eshelman L. On crossover as an evolutionarily viable strategy, *Proceedings of the Fourth International Conference on Genetic Algorithms*, 1991, pp. 61-68.
20. Schumer M., Steiglitz K. Adaptive step size random search, *IEEE Transactions on Automatic Control*, 1968, Vol. 13, No. 3, pp. 270-276.

Статью рекомендовал к опубликованию д.т.н., профессор В.В. Курейчик.

Пикалов Максим Вадимович – Национальный исследовательский университет ИТМО; e-mail: pikmaksim@gmail.com; г. Санкт-Петербург, Россия; тел.: 88126070283; аспирант.

Письмеров Алексей Максимович – e-mail: alekseipismerov@gmail.com; аспирант.

Pikalov Maxim Vadimovich – ITMO University; e-mail: pikmaksim@gmail.com; St. Petersburg, Russia; phone: +78126070283; graduate student.

Pismerov Aleksei Maximovich – e-mail: alekseipismerov@gmail.com; graduate student.

УДК 004.932.2

DOI 10.18522/2311-3103-2024-2-228-236

П.А. Гессен, А.И. Лизин, В.А. Павлова, В.Д. Саенко, В.А. Тупиков

РЕКУРСИВНЫЙ АЛГОРИТМ АНАЛИЗА И ВОССТАНОВЛЕНИЯ КОНТУРОВ В СИСТЕМАХ НАВИГАЦИИ И НАВЕДЕНИЯ

В целях разработки алгоритма обнаружения объектов для встраиваемых вычислительных систем оптико-электронных комплексов, был проведен анализ имеющегося мирового научно-технического опыта, направленного на улучшение процесса выделения контуров. На основе проведенного анализа, авторами статьи был разработан новый метод коррекции контурных изображений. Этот метод реализует подход, позволяющий объединять разорванные контуры и применять фильтрацию по различным параметрам для оптимальной анализа контуров. Первым этапом работы алгоритма является применение размытия изображения, за которым следует применение алгоритма детекции границ Кенни. Затем происходит утоньшение контуров и фильтрация контурного изображения для удаления самых слабых контуров. Следующими этапами являются создание и обработка каждого отдельного контура, а также фильтрация выбросов. Заключительным этапом является соединение и поиск точек перегиба контура. В рамках работы выделены как преимущества, так и недостатки классических методов выделения контуров в контексте использования их в алгоритмах обнаружения объектов. Авторами исследования был проведен анализ двух классических морфологических операторов - дилатации и эрозии, а также существующих основных вариаций их применения таких как открытие и закрытие, как методов объединения контуров. В результате сравнительного анализа результатов работы морфологических операторов дилатации и эрозии, а также основных вариаций их применения, с рекурсивным алгоритмом анализа и восстановления контуров было выявлено преимущество последнего в части сохранения целостности морфологических признаков объектов. Авторами предложены также идеи дальнейшего развития рекурсивного алгоритма анализа и восстановления контуров, а также его дальнейшего применения в задачах обнаружения объектов на изображениях.

Контурный анализ; обработка изображений; восстановление контуров.