# Применение алгоритмов машинного обучения для повышения эффективности алгоритма многокритериальной оптимизации

Баркалов К.А., Козинов Е.А.

# 1. Введение

# 2. Постановка задачи

In the most general form, the multi-objective optimization (MMO) problem can be formulated as follows

|  |  |
| --- | --- |
|  | (1) |

где есть objective functions (критерии эффективности), есть вектор варьируемых параметров, а есть размерность решаемой задачи multi-objective оптимизации. Множество возможных значений параметров (search domain) обычно представляет собой *N*-мерный гиперкуб

|  |  |
| --- | --- |
|  | (2) |

при заданных граничных векторах *a* и *b*.

Не уменьшая общности, предполагается, что objective functions должны быть минимизированы для повышения показателей эффективности принимаемых решений *.* При этомпредполагается, что , *,* являются многоэкстремальными представлены в виде time-consuming “black-box” computational procedures. Предполагается также, что objective functions , *,* удовлетворяют условию Липщица

|  |  |
| --- | --- |
|  | (3) |

where , are the Lipschitz constants and denotes the Euclidean norm in . Условие (3) означает, что при небольших вариациях параметра *y*∈*D* соответствующие изменения значений functions , *,* являются ограниченными.

# 3. Метод решения

В рамках предлагаемого подхода решение задач MMO выполняется в несколько этапов. В рамках первого этапа задача MMO сводится к решению серии скалярных задач глобальной оптимизации. В рамках второго этапа осуществляется совместное решении серии задач глобального поиска. Далее рассматривается общая схема редукции задач MMO, а также приводится метод решения порождаемых задач глобального поиска.

1. **Scalarization of multiple objective functions**. В самом общем виде, задача глобальной оптимизации, порождаемая при скаляризации multiple objective functions задачи МMО, может быть представлена в виде

|  |  |
| --- | --- |
| , , | (4) |

где есть скалярная многоэкстремальная функция, получаемая в результате скаляризации objective functions , *,* а есть вектор параметров используемой свертки functions. В литературе приводятся много методов построения функции : минимаксная свертка, метод оптимизации по образцу, метод -ограничений и т.д. В выполненных вычислительных экспериментах раздела 5 без уменьшения общности предлагаемого подхода использовалась минимаксная свертка частных критериев

|  |  |
| --- | --- |
| , , | (5) |

где , , функции оптимизируемых критериев для decision , а коэффициенты есть показатели значимости каждого критерия. Обычно можно предполагать, что область возможных значений коэффициентов представляется собой множество

|  |  |
| --- | --- |
| *s*. | (6) |

Следует отметить, что в силу (3) скалярная функция *F*(,y) из (5) также удовлетворяет условию Липщица c некоторой константой *L*, т.е.

|  |  |
| --- | --- |
|  | (7) |

# 4. Методы повышения эффективности многокритериального поиска

# 5. Результаты вычислительных экспериментов

Вычислительные эксперименты проводились на суперкомпьютере «Лобачевский» Нижегородского государственного университета (операционная система – CentOS 7, система управления – SLURM). Один узел суперкомпьютера располагает 2-я процессорами Intel Sandy Bridge E5-2660 2.2 GHz, 64 Gb RAM. Центральный процессор является 8-и ядерным (т.е. всего на узле доступно 16 ядер CPU). Для получения исполняемого программного кода использовался компилятор Intel C++ 17.0.0, python 3.9 and scikit-learn 0.24.2. The numerical experiments were performed using the Globalizer system [XXX].

Первая серия экспериментов была выполнена для сравнения алгоритма MGSA с рядом широко известных алгоритмов многокритериальной оптимизации на примере решения тестовой двухкритериальной задачи [XXX]

|  |  |
| --- | --- |
|  | (35) |

В ходе экспериментов для решения задачи (35) выполнялось построение численной аппроксимации области Парето. Качество аппроксимации оценивалось с помощью показателей the hypervolume and distribution uniformity indexes (HV). [37, 40]. Первый из этих показателей характеризует полноту аппроксимации (большее значение соответствует более полному покрытию области Парето), а второй – равномерность покрытия (меньшее значение соответствует более равномерному покрытию области Парето).

В рамках данного эксперимента сравнивались пять алгоритмов многокритериальной оптимизации: the Monte-Carlo (MC) method, the genetic algorithm SEMO from the PISA library [42], the Non-uniform coverage (NUC) method [40], the bi-objective Lipschitz optimization (BLO) method [39] и алгоритм MGSA, предложенного в данной статье. For the first three algorithms, the numerical results were used from [41]. The results of the BLO method were presented in [39].

Для MGSA было решено 25 задач (4) при разных значениях коэффициентов свертки , равномерно распределенных в Λ. При проведении экспериментов параметр надежности был задан , а параметр точности . В полном виде результаты выполненных экспериментов представлены в таблице 1.

Таблица 1. Сравнение эффективности алгоритмов многокритериальной оптимизации

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Метод решения** | MC | SEMO | NUC | BLO | **MGSA**  **()** | **MGSA**  **()** |
| **Количество  итераций метода** | 500 | 500 | 515 | 498 | **302** | **269** |
| **Количество найденных точек  Парето границы** | 67 | 104 | 29 | 68 | **92** | **79** |
| **HV индекс** | 0.300 | 0.312 | 0.306 | 0.308 | **0.312** | **0.312** |
| **DU индекс** | 1.277 | 1.116 | 0.210 | 0.175 | **0.101** | **0.103** |

Как показывают результаты выполненных экспериментов, использование базового алгоритма MGSA (т.е. при ) дает заметное преимущество по сравнению с рассмотренными методами многокритериальной оптимизации даже при решении сравнительно простых задач МКО. Использование методов машинного обучения позволяет дополнительно сократить число выполняемых испытаний глобального поиска.

Во второй серии экспериментов проводилось решение двухкритериальных двумерных задач MMO, т.е. , . В качестве критериев задачи использовались многоэкстремальные функции, определяемых соотношениями [XXX]:

|  |  |
| --- | --- |
|  | (36) |

where

,

are defined in the range , and the parameters are the independent random numbers distributed uniformly. На рис. 1 показаны линии уровней нескольких функций из данного семейства.

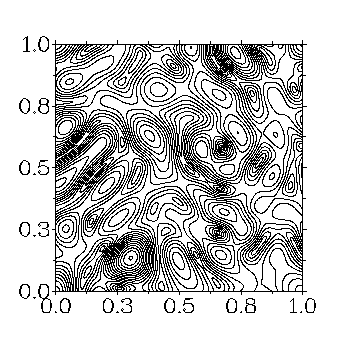
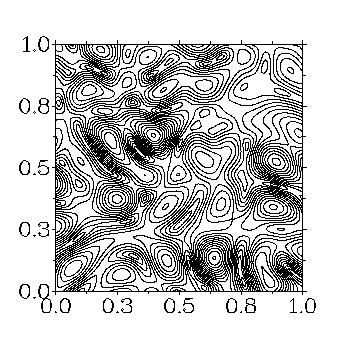
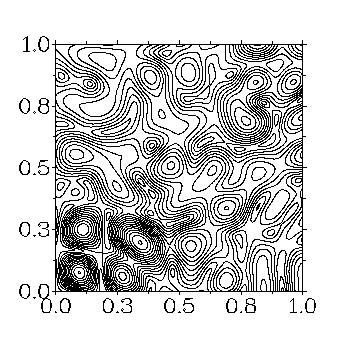


Fig. 2. Линии уровня нескольких многоэкстремальных функций   
используемых во второй серии вычислительных экспериментов

На начальном этапе решения серии задач MMO необходимо было выбрать значение параметра из (XXX). Выбранное значение показателя из (XXX) должно обеспечивать приемлемое качество решения задач с точки зрения показателей HV и DU, а также сокращение количество испытаний решения серии задач. Для более обоснованного выбора значения параметра из (XXX) рассматривалась одна задача из серии. Параметры точности , надежности и количества решаемых скалярных задач (4) равное 50 были зафиксированы. Значения параметра из (XXX) менялось в диапазоне [0..0.1]. Изменение показателей HV и DU, а также количество испытаний в зависимости от параметра показаны на рис. 2.

|  |  |
| --- | --- |
|  |  |

Fig. 2. Изменение показателей построенной области Парето в одной из задач MMO в зависимости от выбранного показателя   
(слева – изменение числа итераций, справа – изменение показателей HV и DU)

Изменение заполнения области поиска при использовании алгоритмов машинного обучения показано на рис. 3.

|  |  |
| --- | --- |
|  |  |

Fig. 3. Точки испытаний в зависимости от выбранного значения параметра

Исходя из рисунка 2 видно, что увеличение параметра из (XXX) приводит к существенному уменьшению числа испытаний, однако, вместе с тем с увеличением значения параметра из (XXX) ухудшаются значения параметров HV и DU. Исходя из проведенной серии экспериментов было зафиксировано значение . На рисунке 3 видно, что использование значения параметра позволяет уменьшить число испытаний вне области Парето, вместе с тем в окрестности области Парето плотность точек испытаний остается высокой.

Для построения более обоснованных выводов об эффективности разработанного подхода было выполнено решение 100 многокритериальных задач, формируемых с использованием многоэкстремальных функций семейства (36). В таблице 2 приведены усредненные результаты вычислительных экспериментов.

Таблица 2. Сравнение эффективности алгоритмов многокритериальной оптимизации  
с использованием и без использования алгоритмов машинного обучения   
при решении серии из 100 задач MMO

|  |  |  |
| --- | --- | --- |
|  | MGSA (α=0.0) | MGSA (α=0.01) |
| Average number of iterations | 1902,4 | 658,6 |
| Average value of  DU indexe (less is better) | 1,32 | 1,55 |
| Average value of  HV indexe (more is better) | 92,1 | 91,9 |
| Reducing the number of iterations | 1 | 2,9 |

Исходя из таблицы 2 видно, что за счет применения алгоритмов машинного обучения удалось добиться существенного сокращения количества испытаний (почти в 3 раза) для решения серии задач MMO с сохранением схожих значений показателей HV и DU.

# Литература