

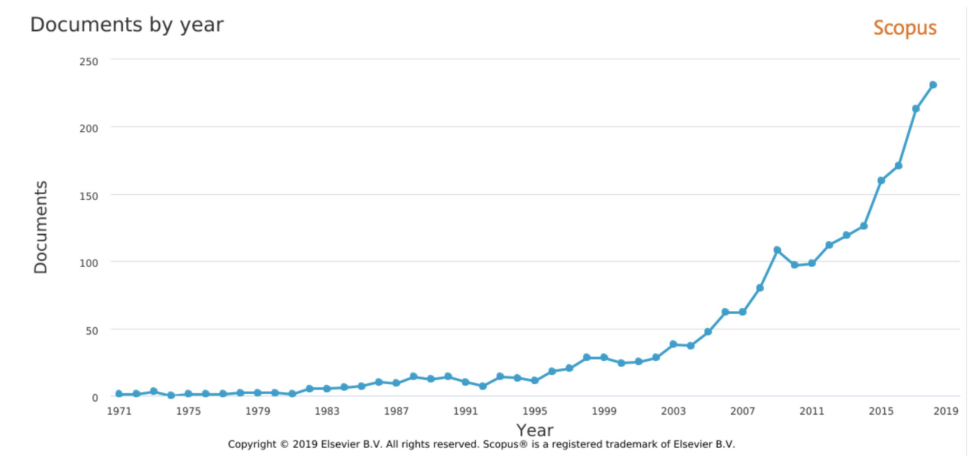
Московский государственный университет имени М. В. Ломоносова
Факультет Вычислительной математики и кибернетики

Разработка и исследование методов анализа эффективности работы эволюционных и роевых алгоритмов

Чепурнов А. В., Ершов Н. М.

Крупноразмерные задачи оптимизации

- Может не быть информации о ландшафте целевой функции (чёрный ящик), только область определения;
- Число параметров может исчисляться десятками тысячами;
- Находить значения целевой функции может быть вычислительно сложно;
- Имеется ограничение на количество вычислений целевой функции («бюджет»).



Рост интереса к теме

Цели исследования

- «Проклятие размерности» значительно затрудняет решение крупноразмерных задач оптимизации;
- Алгоритмы оптимизации обладают низкой обобщаемостью на широкие классы таких задач;
- Сложно прогнозировать, какой алгоритм (и с какой конфигурацией) наиболее эффективно решит данную задачу.

Цель нашего исследования — разработка метода прогнозирования эффективности работы алгоритмов оптимизации, основанного на проведении статистического анализа гибридными методами фильтрации данных.

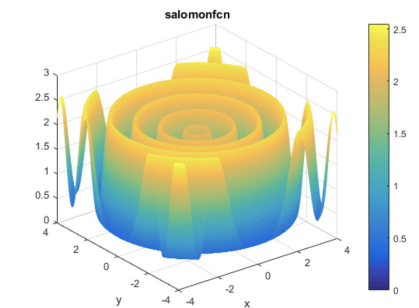
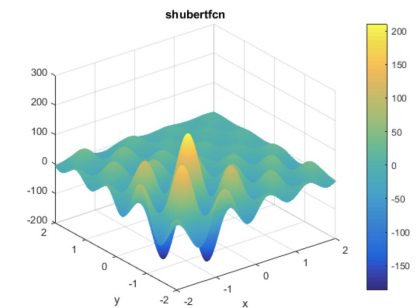
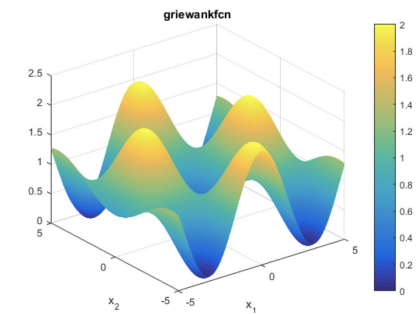
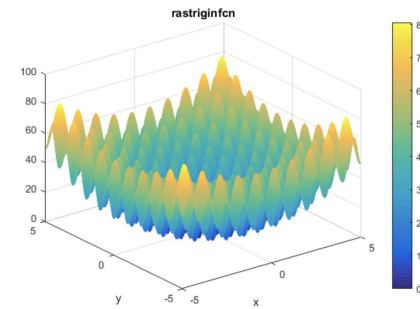
Эволюционные и роевые алгоритмы

- Particle Swarm Optimization (PSO) — метод роя частиц
- A Competitive Swarm Optimizer (CSO) — метод конкурирующего роя
- Bacterial Foraging optimization (BF) — алгоритм бактериального поиска
- Genetic Algorithm (GA) — генетический алгоритм
- Artificial Bee Colony optimization (ABC) — алгоритм пчелиной колонии
- Differential evolution (DE) — алгоритм дифференциальной эволюции

Бенчмарки

IEEE CEC, YABBOB, Nevergrad, OptimSuite

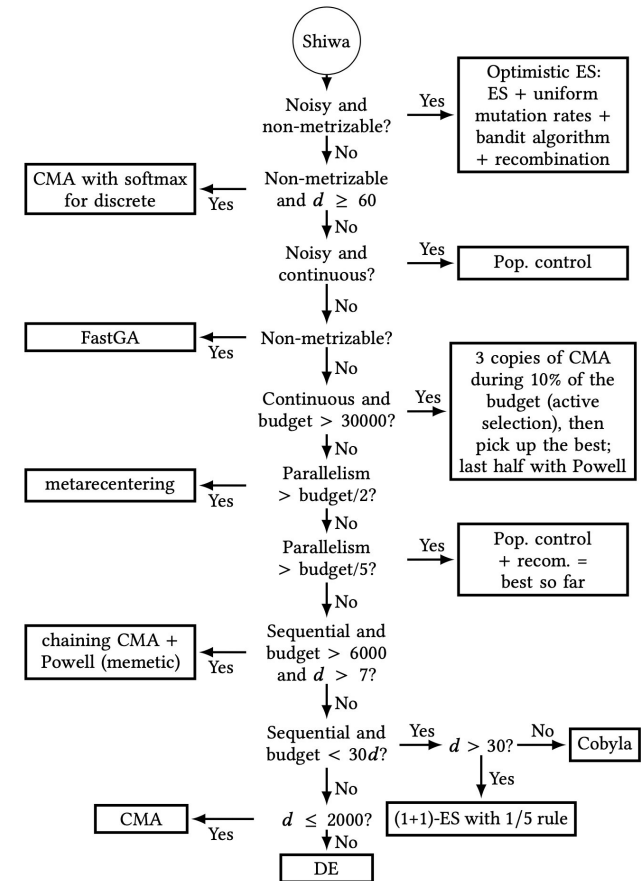
- Комбинации «классических» тестовых аналитических функций (варьируется сложность ландшафта);
- Real-world MLDA (machine learning and data analysis);
- Задачи оптимизации энергетических систем;
- Задачи визуализации графов;
- Traveling Salesman Problem;
- ...



Системы выбора алгоритмов

Shiwa (2020), ABBO Wizard (2021)

1. Пассивный выбор алгоритмов — дерево решений по общим характеристикам задачи;
2. Активный выбор алгоритмов — по численным результатам алгоритмов для задачи;
3. Выбор гиперпараметров алгоритма — отдельно (grid search, байесовская оптимизация).



Дерево решений системы Shiwa

Предлагаемые подходы к решению

Для предсказания эффективности предлагается использовать обширную статистику работы алгоритмов на максимально широком классе задач.

Эти данные позволяют выделить кластеры алгоритмов, проявляющих схожие свойства.

1. Для новых задач потребуется выполнить вычисления для набора ключевых алгоритмов из различных кластеров;
2. Затем становится возможным прогнозирование эффективности алгоритмов (с учётом гиперпараметров) методами рекомендательных систем.

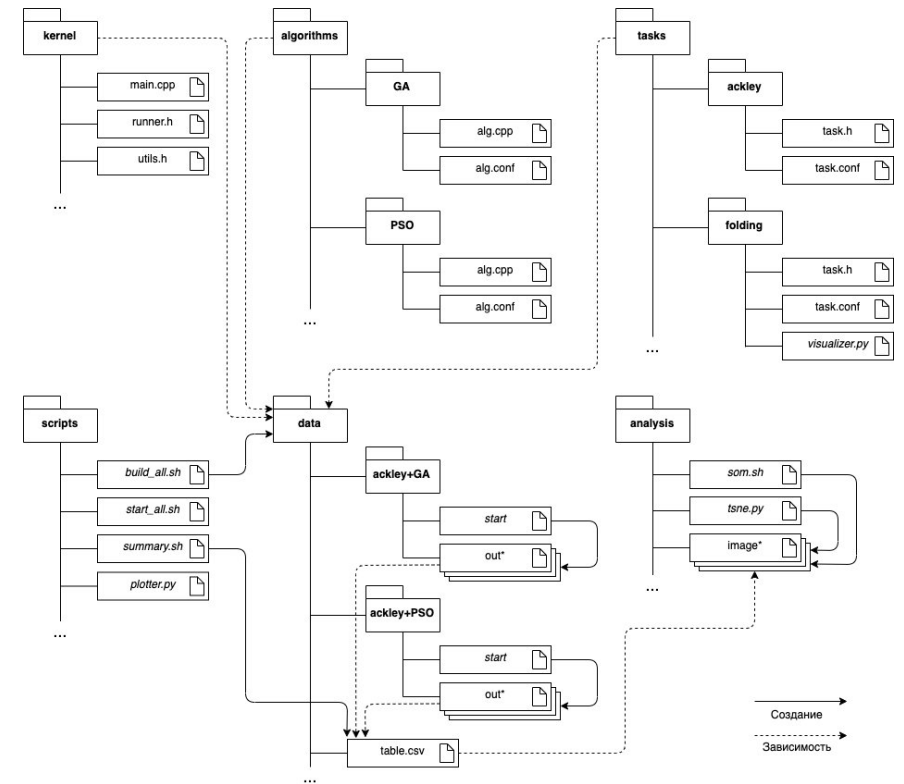
Подход даёт возможность учитывать свойства алгоритмов для задач разных типов.

Сбор статистики об эффективности работы алгоритмов

- Создана программная система тестирования алгоритмов и задач в параллельном режиме на языке C++;
- С помощью системы осуществлён сбор обширной информации о работе эволюционных и роевых алгоритмов;
- Посчитаны оценки эффективности их работы для каждой задачи:

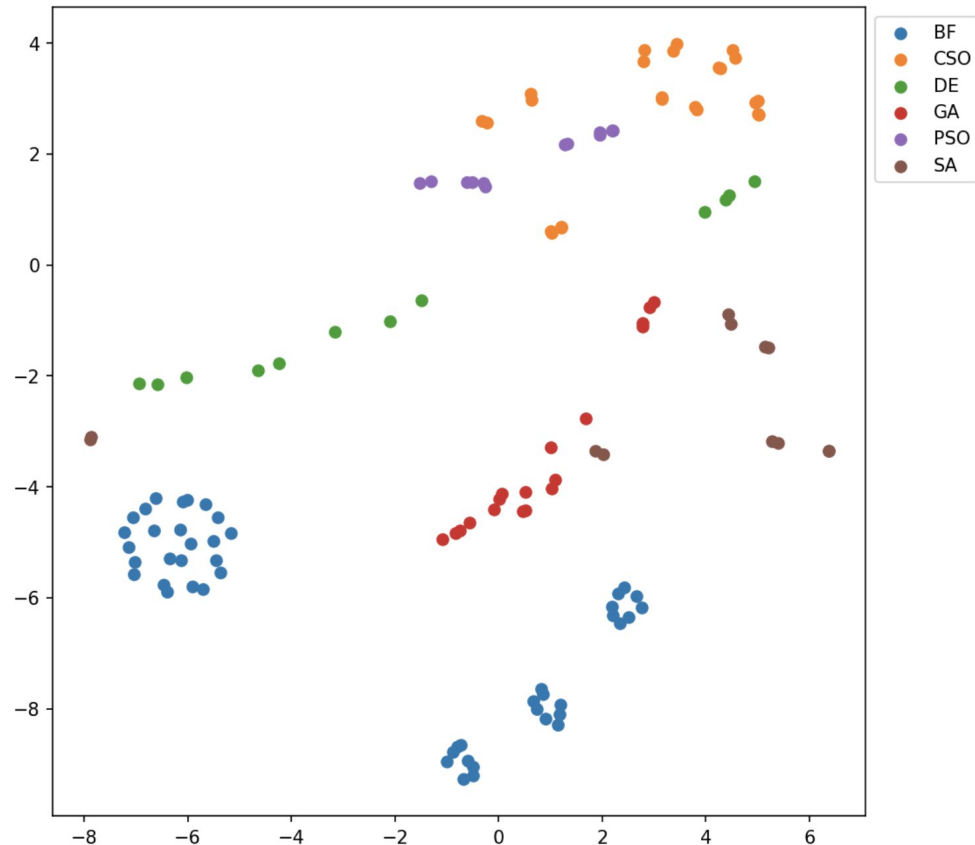
$$m_{ij} = \sqrt{f_{ij} r_{ij}}, \text{ где } f_{ij} \text{ и } r_{ij} \text{ — лучшее полученное}$$

алгоритмом значение целевой функции и его ранг, нормированные в (0, 1).



Файловая структура программного комплекса

Проблема «холодного старта»

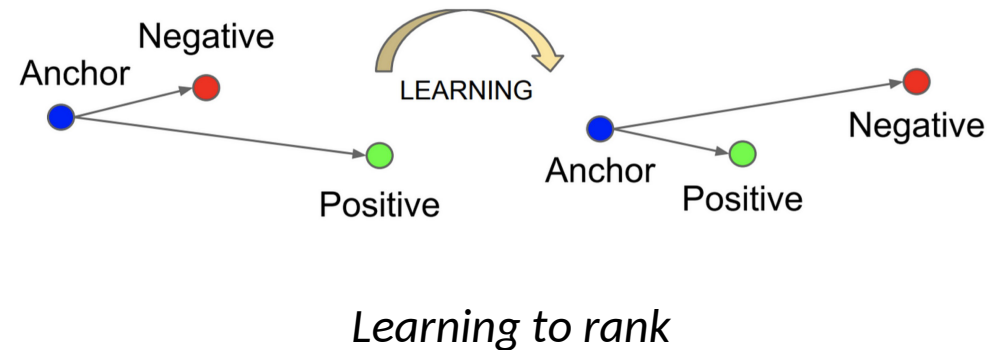


Визуализация пространства алгоритмов
методом *tSNE*

- Удалось установить, что алгоритмы хорошо поддаются кластеризации по собранным данным;
- Кластеры, которые можно увидеть на визуализациях, полученных нелинейными проекциями на плоскость, выявляются K-means и DBSCAN;
- Для пассивного выбора предлагается использовать алгоритмы-центроиды всех кластеров.

Прогнозирование оценок эффективности

- Проведён анализ методов факторизации с использованием искусственных нейронных сетей для прогнозирования эффективности алгоритмов;
- Удалось установить, что ранжирующая модель факторизации (learning to rank) позволяет делать прогнозы для разреженной матрицы оценок эффективности с меньшей ранговой ошибкой, чем классическая матричная факторизация.



Ранговая ошибка для матричной факторизации

α	K = 5	K = 10	K = 15	K = 20
25%	0,49 (0,26)	0,44 (0,23)	0,42 (0,21)	0,39 (0,20)
50%	0,58 (0,29)	0,52 (0,25)	0,49 (0,24)	0,47 (0,24)
75%	0,76 (0,38)	0,68 (0,36)	0,65 (0,33)	0,62 (0,33)

Матричная факторизация с регуляризацией

α	K = 5	K = 10	K = 15	K = 20
25%	0,46 (0,25)	0,41 (0,22)	0,40 (0,19)	0,37 (0,19)
50%	0,53 (0,26)	0,47 (0,26)	0,47 (0,21)	0,47 (0,21)
75%	0,67 (0,35)	0,61 (0,33)	0,61 (0,30)	0,59 (0,31)

Learning to rank

α — доля помеченных неизвестными элементами матрицы M (оценок эффективности)

K — количество признаков

В скобках указана доля неверно отранжированных пар среди всех пар алгоритмов

Направления дальнейшей работы

- Расширение статистики о работе алгоритмов за счёт интеграции системы с рассмотренными бенчмарками и добавления в систему новых алгоритмов;
- Сравнение результатов с Shiwa и ABBO Wizard для крупноразмерных задач;
- Исследование других моделей прогнозирования оценок эффективности, в том числе гибридных методов рекомендаций;
- Создание удобного клиента для пользователей.

Спасибо за внимание!