

Министерство науки и высшего образования Российской Федерации

Национальный исследовательский университет ИТМО

Эволюционные вычисления

Весна

2024

Лабораторная работа №5

ПРОЕКТИРОВАНИЕ ЭВОЛЮЦИОННОГО АЛГОРИТМА ДЛЯ ЗАДАЧИ РАССТАНОВКИ ФЕРЗЕЙ

Цель работы

Целью данной работы является освоение всего цикла разработки эволюционных алгоритмов, начиная с анализа проблемы и проектирования, заканчивая настройкой параметров и анализом эффективности.

Краткие теоретические сведения

Перед разработкой эволюционного алгоритма для решения некоторой проблемы, необходимо задать себе ряд вопросов.

Какова цель разработки алгоритма?

- найти некоторое решение задачи;
- найти глобально оптимальное решение задачи;
- находить хорошие решения при частом запуске алгоритма;
- найти множество разнородных экземпляров хороших решений;
- разработать алгоритм, который будет эффективней существующих.

Необходимо проанализировать контекст решаемой проблемы. Насколько быстро должно быть получено решение? Что подразумевается под решением проблемы? Как оценивается качество решения? Существуют ли ограничения? Например, функция оценки качества решений может быть осложнена следующими составляющими:

- нестационарность, когда оценка качества одного и того же решения зависит от времени;
- зашумленность, в функции оценки качества присутствует компонент случайности;
- ограниченность, невозможность вычисления функции при определенных условиях;
- многокритериальность, необходимость нахождения компромисса между несколькими критериями оценки, а также нахождения множества недоминантных решений;
- интерактивность, в роли функции оценки качества выступает человек-эксперт.

После анализа проблемы можно приступить к проектированию алгоритма.

Существует большое множество эволюционных алгоритмов, кроме ГА:

- метод роя частиц (Particle Swarm Optimization), основан на модели движения стаи птиц;
- гравитационный алгоритм (Gravitational Search Algorithm), основан на принципах закона всемирного тяготения;
- муравьиный алгоритм (Ant Colony Optimization), основан на поведении муравьёв при поиске источников питания;
- гармонический поиск (Harmony Search), основан на принципах сочинения музыки;
- алгоритмы чёрной дыры или сверхновой;
- алгоритм капель воды и многие другие [4].

Нужно понять, как решения могут быть представимы в рамках алгоритмов и выбрать наиболее предпочтительный алгоритм. Также, следует заранее спроектировать, как будет осуществляться процесс вариации решений.

После реализации алгоритма, необходимо перейти к анализу его производительности и эффективности. В основном все эволюционные алгоритмы имеют итерационный характер, где на каждой итерации происходит вариация решений (популяции) и оценка их качества. Таким образом, для анализа эффективности алгоритма могут быть использованы следующие характеристики:

- шанс успеха, вероятность нахождения необходимого решения при множественном запуске алгоритма;
- среднее качество решений;
- скорость алгоритма.

Скорость алгоритма может быть представлена как:

- реальное время выполнения алгоритма;
- количество итераций алгоритма;
- количество вычислений фитнес функции.

В общем, количество вычислений фитнес функции является наиболее честным показателем скорости алгоритма, поскольку не зависит от размера популяции или от качества реализации самой программы.

Все эволюционные алгоритмы стохастичны, поэтому необходимо проводить серии запусков для получения устойчивого результата анализа эффективности. Также, важной проблемой является настройка и контроль параметров алгоритма. Настройка параметров является сложной и время-затратной задачей. Для её решения, в зависимости от цели разработки алгоритма, могут быть использованы следующие подходы:

- интуитивный подбор руками;
- поиск по сетке параметров;
- мета-эволюционный алгоритм;
- методы на основе моделирования (например, Sequential Parameter Optimization [3]).

При анализе эффективности алгоритма, требуется также оценить его возможности масштабирования, т.е. способность решения задачи большей размерности, а также влияния параметров при изменении размерности задачи.

В настоящее время существует большое число фреймворков и инструментов для построения эволюционных алгоритмов со встроенными примерами, схемами распределения, параллелизации и анализа алгоритмов.

В рамках лабораторных работ за основу взят фреймворк Watchmaker, из-за наиболее интуитивно понятного построения и легкой конструкции базовых схем эволюции. Можно привести несколько других примеров:

- Jenetics (<http://jenetics.io/>), Java, основан на эффективном использовании возможностей Java Streams;
- ECJ (<https://cs.gmu.edu/~eclab/projects/ecj/>), Java, включено огромное количество схем алгоритмов для решения разнородных задач;
- DEAP (<https://deap.readthedocs.io/en/master/>), популярная библиотека для языка Python, с возможностью параллелизации;
- JMetal (<http://jmetal.sourceforge.net/>), обширная библиотека для Java, также есть адаптация для C++.

В данной лабораторной работе решается классическая **задача расстановки ферзей**. Дана шахматная доска $N \times N$, на которой необходимо расставить N ферзей таким образом, чтобы они не били друг друга (не пересекались по горизонталям, вертикалям и диагоналям). Пример решения для задачи с $N=8$ представлена на рисунке 4.1.

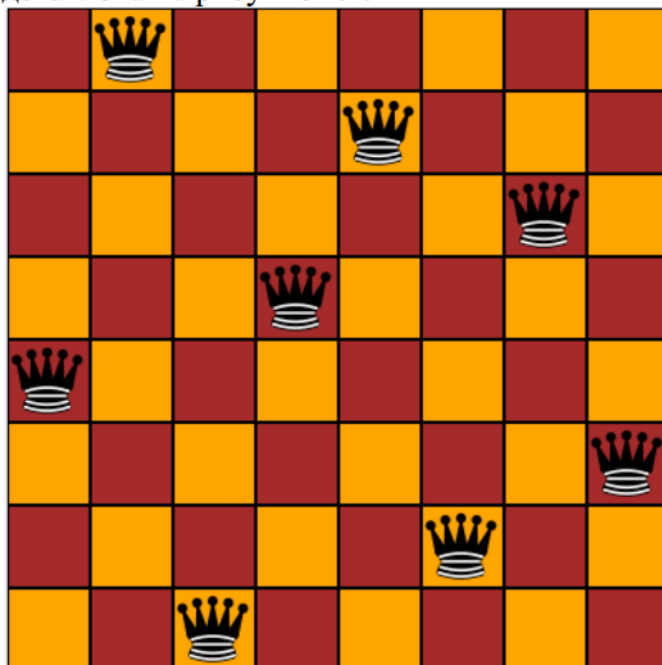


Рисунок 4.1 – Пример решения расстановки ферзей на доске 8×8 .

Ход работы

1. Выбрать фреймворк (можно продолжить в Watchmaker) для алгоритма или сделать свою реализацию.
2. Выбрать эволюционный алгоритм (ГА или другой).
3. Проанализировать задачу, выделить критерий оптимизации или ограничения.
4. Спроектировать решение поставленной задачи, вид решения, способы вариаций и обработки возможных ограничений.
5. Сформировать условия терминации или сходимости к алгоритму.
6. Установить характеристики для измерения эффективности алгоритма.
7. Настроить параметры алгоритма.
8. Провести серии запусков при разных значениях N и провести анализ эффективности.

В отчёте необходимо описать выполнение каждого пункта хода работы и заполнить результаты на основе таблицы 4.1.

Таблица 4.1 Результаты производительности алгоритма

N	Характеристика 1	Характеристика 2	...	Характеристика К
4				
8				
...				

Вопросы

1. Является ли задача оптимизационной или ограниченной?
2. Как растёт сложность задачи при увеличении размерности?

Литература

1. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. – 2013.
2. Eiben A. E., Smith J. Introduction to Evolutionary Computing. – 2015.
3. Bartz-Beielstein T., Lasarczyk C. W. G., Preuß M. Sequential parameter optimization //Evolutionary Computation, 2005. The 2005 IEEE Congress on. – IEEE, 2005. – Т. 1. – С. 773-780.
4. Rajakumar R., Dhavachelvan P., Vengattaraman T. A survey on nature inspired meta-heuristic algorithms with its domain specifications //Communication and Electronics Systems (ICCES), International Conference on. – IEEE, 2016. – С. 1-6.