Прошян Г.А. M4150

Лабораторная работа №3. Генетический алгоитм для задачи оптимизации непрерывной функции.

Цель работы

Получние навыков разработки и анализа эволюционных операторов эволюционных алгоритмов генетического алгоритма для решения задачи оптимизации непрерывной веществennозначной функции.

Оборудование и программное обеспечение

- Java JDK версии 1.8 и выше
- Watchmaker framework версии 0.7.1 (https://github.com/dwdyer/watchmaker)
- Шаблон проекта: https://gitlab.com/itmo_ec_labs/lab2

Условие

- Область определения всех переменных целевой функции: [-5, 5]
- Область определения целевой функции [0, 10]
- Заведомо известно, что глобальный оптимум = 10.

Инициализация полпуляции (MyFactory.java)

Инициализация выполняется с помощью генерации случайного числа в интервале [0, 1] типа double методом `random.nextDouble()` и последующей линейной трансформации для изменения диапозона значений.

Кроссовер (MyCrossover.java)

Так как дискретный кроссовер является частным случаем арифметического, был реализован арифметический. Арифметический кроссовер вычисляется по формуле:

$$z_i = \alpha \cdot x_i + (1 - \alpha) \cdot y_i$$

Метод `mate` принимает на вход родителей `p1`, `p2` и число точек, для которых производится кроссовер `crossoverPointsNum`, а также объект `random`. Для каждого дочернего элемента мы копируем в него соответсвующего родителя, генерируем случайным образом индексы, для которых будет произведен кроссовер (число индекссов = `crossoverPointsNum`) и для данных индексов производим вычисление по формуле выше.

Были опробованы различные значения `alpha`. Во всех экспериментах, представленных в данном отчета `alpha` = 0.5

Мутация (MyMutation.java)

Была имплементирована равномерная мутация. Каждый ген особи заменяется на новое случайное число с вероятностью `mutation_prob`. На основе незадукументированных эмперических исследований, оптимальные значения `mutation_prob` для решения задачи данной лабораторной лежат в интервале [0.005, 0.01]. Во всех экспериментах, представленых в данном отчете `mutation_prob` = 0.005.

Эксприменты

Производился подбор populationSize < 100, generetions < 10'000, максимизирующие фитнесс-функцию. Очевидно, увеличение числа итераций всегда приводит к неуменьшению фитнесс-функции, поэтому данный параметр не максимизировался, а скорее наоборот, было интересно при каком минимальном занчении числа итераций получится превысить порог в 9.5

Размер проблемы	Размер популяции	Количество итераций	Результат
2	30	15	9.816
10	45	650	9.769
20	30	1500	9.893
50	60	5000	9.721
100	70	9999	9.567

Ответы на вопросы

1. Что важнее кроссовер или мутация?

Оба механизма являются крайне важными и только комбинируя их можно получить наиболее значимые результаты. Кроссовер является основным генетическим оператором, однако без оператора мутации алгоритм будет застревать в локальных оптимумах. Если все же необходимо выбрать один из операторов, то я выберу кроссовер, так как именно благодаря нему особи "развиваются и эволюционируют". Также в проведенных мной опытах было получено, что "отключение" (замена на dummy заглушку) кроссовера привело к более низким результатам, чем "отключение" мутации.

Выключение кроссовера:

Размер проблемы	**Размер популяции**	**Количество итераций**	**Результат**
2	30	15	6.929
10	45	650	5.348
20	30	1500	5.010
50	60	5000	4.783
100	70	9999	4.426

Выключение мутации:

Размер проблемы	**Размер популяции**	**Количество итераций**	**Результат**
2	30	15	9.973
10	45	650	7.014
20	30	1500	6.127
50	60	5000	5.395
100	70	9999	4.995

2. Как влияет значение параметра "размер популяции" на производительность и эффективность алгоритма?

   Увеличение размера популяции приводит к повышению числа вычислений. В проведенных экспериментах повышение размера популции зачастую приводило к повышению значения фитнесс-функции, но не всегда.

3. Важно ли знать область определения переменных целевой функции?

Важно, так как в контексте решения решения реальных практических задач оптимизация фитнесс функции имеет смысл только если гены входят в область определения. Если оптимальное найденное решение имеет невозможные гены, оно для нас бесполезено, и его нахождение было бессмысленной тратой ресурсов. Так наример, если хотим построить максимально устойчивую балку спуткника, а получено решение, которое технически невозможно построить/ отлить в реальном мире, вычисления будут напрасными.