УДК 004-021

Исследование генетического алгоритма в SciLab

Колесников Алексей Александрович
Приамурский государственный университет им. Шолом-Алейхема
Студент

Баженов Руслан Иванович
Приамурский государственный университет им. Шолом-Алейхема
к.п.н., доцент, зав. кафедрой информационных систем, математики и методик обучения

Аннотация
В статье рассматривается метод решения функций генетическим алгоритмом с помощью средств SciLab.
Ключевые слова: SciLab, генетический алгоритм, минимум функции.

Research of genetic algorithm in SciLab

Kolesnikov Aleksey Aleksandrovich
Sholom-Aleichem Priamursky State University
Student

Bazhenov Ruslan Ivanovich
Sholom-Aleichem Priamursky State University
Candidate of pedagogical sciences, associate professor, Head of the Department of Information Systems, Mathematics and teaching methods

Abstract
The article discusses the method of solving functions the genetic algorithm by means of SciLab.
Keywords: SciLab, genetic algorithm, minimum function.

В наше время существует множество различных способов решения задач с разным количеством неопределенных параметров, как, например, линейные уравнения, системы уравнений с двумя и более неизвестными. И для каждой задачи имеется свой метод решения. Однако довольно сложно решать задачи со всеми неизвестными. Для решения таких задач был придуман генетический алгоритм.

Генетический алгоритм является одним из наиболее популярных методов решения многокритериальных задач. В статьях В.А.Тененёва[1], К.Ю.Брестера и Е.С.Семенкина[2], Н.Ю.Паротькина и В.Г.Жукова[3] с помощью генетического алгоритма решаются задачи многокритериальной оптимизации. В.В.Возыка применяет для построения дерева Штейнера[4], а

Т.С.Емельянова решает транспортные задачи[5][6]. В статье В.А.Тененёва и А.В.Тененёвой используется для обучения нечетких нейронных сетей[7]. Также генетический алгоритм применяется в статье I.Ya.Lvovich и др. для строительства объектов с необходимыми средними значениями характеристик [8] и в статье V.D.Tsoukalas и N.G.Fragiadakis для прогнозирования риска в судостроительной отрасли [9].

Генетический алгоритм - это метод поиска решений практически любой оптимизационной задачи, моделирующий процессы природной эволюции.

Генетический алгоритм состоит из нескольких этапов:

1. Инициация
2. Выращивание
3. Оценивание
4. Селекция
5. Скрещивание
6. Мутация

Для решения задач в SciLab будет использоваться гибкий генетический алгоритм. Последовательность вызова функции выглядит следующим образом:

[pop_opt,fobj_pop_opt,pop_init,fobj_pop_init]=optim_ga(ga_f,pop_size,nb_generation,p_mut,p_cross,Log,param)

Рассмотрим каждый параметр функции:

- ga_f - функция, которую будем оптимизировать. Функция может быть в виде либо линейного уравнения, например, $y = f(x)$, либо в виде списка $y = list(f, p1, p2, ..)$.
- pop_size - количество особей в популяции (значение по умолчанию: 100).
- nb_generation - количество итераций (поколений) (по умолчанию: 10).
- p_mut - вероятность мутации (по умолчанию: 0.1).
- p_cross - вероятность кроссовера (значение по умолчанию: 0.7).
- Log - если %T, то функция будет выводить значения после каждой итерации.
- Param - список дополнительных параметров:
    1. codage_func - функция, которая будет выполнять кодирование и декодирования особи (по умолчанию: coding_ga_identity).
    2. init_func - функция инициализации популяции (по умолчанию: init_ga_default).
    3. dimension - параметр, используемые в функции инициализации для определения начальной популяции.
    4. minbounds and maxbounds - параметры, определяющие минимум и максимум переменной X соответственно.

5. crossover_func - функция, которая будет выполнять кроссовер (сцепление) между двумя особями (по умолчанию: crossover_ga_default).
6. mutation_func - функция, которая будет выполнять мутацию одной особи (по умолчанию: mutation_ga_default).
7. selection_func - функция, которая будет выполнять отбор особей в конце поколения (по умолчанию функция: selection_ga_elitist).
8. nb_couples - количество пар, которые будут выбраны для выполнения кроссовера и мутации (значение по умолчанию: 100).
9. pressure - значение эффективности худшей особи (значение по умолчанию: 0.05).
10. output_func - функция обратного вызова, которая вызывается после каждого поколения, если Log это %T (По умолчанию: output_ga_default).

- pop_opt - оптимальная особь.
- fobj_ pop_opt - набор значений функции, связанных с pop_opt (параметр является необязательным).
- pop_ init - начальная популяция особей.
- fobj_ pop_init - набор значений функции, связанных с pop_init (параметр является необязательным).

Для примера использования генетического алгоритма исследуем минимизирование функции $f(x) = x^2$ с изменяющимися параметрами. По умолчанию область значений x берется в промежутке [0;1]. Пример решения функции показан на рисунке 1.

```
-->function y=f(x)
-->y=sum(x.^2)
-->endfunction

-->PopSize=100;

-->NbGen=10;

-->Proba_mut=0.1;

-->Proba_cross=0.7;

-->Log=%T;

-->ga_params=init_param();

-->ga_params=add_param(ga_params,"dimension",3);

-->[pop_opt,fobj_pop_opt]=..
-->optim_ga(f,PopSize,NbGen,Proba_mut,Proba_cross,Log,ga_params);
optim_ga: iteration 1 / 10
    min / max value found = 0.011330 / 2.123777
optim_ga: iteration 2 / 10
    min / max value found = 0.005061 / 0.653990
optim_ga: iteration 3 / 10
    min / max value found = 0.004230 / 0.191636
optim_ga: iteration 4 / 10
    min / max value found = 0.001879 / 0.057040
optim_ga: iteration 5 / 10
    min / max value found = 0.000639 / 0.015700
optim_ga: iteration 6 / 10
    min / max value found = 0.000336 / 0.005520
optim_ga: iteration 7 / 10
    min / max value found = 0.000076 / 0.002376
optim_ga: iteration 8 / 10
    min / max value found = 0.000019 / 0.000985
optim_ga: iteration 9 / 10
    min / max value found = 0.000019 / 0.000442
optim_ga: iteration 10 / 10
    min / max value found = 0.000005 / 0.000188
```

Рисунок 1 - Пример решения функции

Итоговые значение x внесем в таблицу измерений (табл. 1). Проведем ещё несколько вычислений изменяя входные параметры функции.

Таблица 1 - Данные об изменениях

Pop_Size	**Nb_gen**	**P_mut**	**P_cross**	**Min x**	**Max x**
100	10	0.1	0.7	0.000005	0.000188
100	10	0.1	0.8	0.000005	0.000042

100	10	0.1	0.9	0.000005	0.000153
100	10	0.3	0.7	0.000012	0.000341
100	10	0.5	0.7	0.000049	0.000607
100	15	0.1	0.7	0.000000	0.000006
100	20	0.1	0.7	0.000000	0.000000
150	10	0.1	0.7	0.000035	0.001252
200	10	0.1	0.7	0.000295	0.005187

Используя значения X таблицы 1 найдем среднюю погрешность вычислений относительно "0" и внесем полученные результаты в таблицу 2.

Таблица 2 - Данные о погрешностях

Min x	**Max x**	**Ср. значение x, (Xmax+Xmin)/2**	**Погрешность, % Xср*100%**
0.000005	0.000188	0.000097	0.0097
0.000005	0.000042	0.000024	0.0024
0.000005	0.000153	0.000079	0.0079
0.000012	0.000341	0.000172	0.0172
0.000049	0.000607	0.000328	0.0328
0.000000	0.000006	0.000003	0.0003
0.000000	0.000000	0	0
0.000035	0.001252	0.000644	0.0644
0.000295	0.005187	0.002741	0.2741

Исходя из вычисленных данных таблицы 2 следует, что при увеличении числа итераций, средняя погрешность вычислений стремится к нулю, а при увеличении остальных параметров, погрешность растет. Следовательно, чем больше число поколений и, чем меньше начальная популяции, вероятность мутации и кроссинговера, тем ближе результат вычисления будет к реальному.

Для решения многопараметрических уравнений в используется функция - "list" в которой пишутся все используемые в уравнении параметры. Решение для $3*x^2-8$ показано на рисунке 2.

```
-->function y=f(x,a1,a2)
-->y=a1*sum(x.^2)+a2
-->endfunction

-->PopSize= 100;

-->Proba_cross= 0.7;

-->Proba_mut= 0.1;

-->NbGen= 10;

-->Log= %T;

-->ga_params = init_param();

-->ga_params = add_param(ga_params,"dimension",3);

-->ga_params=add_param(ga_params,"minbound",[-2;-2]);

-->ga_params=add_param(ga_params,"maxbound",[2;2]);

-->a1=3;

-->a2=-8;

-->myobjfun = list(f,a1,a2);

-->[pop_opt,fobj_pop_opt] = ..
-->optim_ga(myobjfun,PopSize,NbGen,Proba_mut,Proba_cross,Log,ga_params);
optim_ga: iteration 1 / 10
    min / max value found = -7.982748 / -4.628011
optim_ga: iteration 2 / 10
    min / max value found = -7.999967 / -7.417788
optim_ga: iteration 3 / 10
    min / max value found = -7.999967 / -7.869558
optim_ga: iteration 4 / 10
    min / max value found = -7.999967 / -7.959888
optim_ga: iteration 5 / 10
    min / max value found = -7.999998 / -7.990325
optim_ga: iteration 6 / 10
    min / max value found = -7.999998 / -7.998401
optim_ga: iteration 7 / 10
    min / max value found = -7.999998 / -7.999781
optim_ga: iteration 8 / 10
    min / max value found = -7.999999 / -7.999967
optim_ga: iteration 9 / 10
    min / max value found = -8.000000 / -7.999991
optim_ga: iteration 10 / 10
    min / max value found = -8.000000 / -7.999998
```

Рисунок 2 - Пример решения функции с параметрами

Таким образом, в статье был рассмотрен генетический алгоритм с помощью средств программы SciLab. Была подробно рассмотрен пример выполнения алгоритма на примере квадратичной функции.

Рассмотрение алгоритма показало простоту применения, но также раскрыло его сложность - относительно большую погрешность вычислений.

Библиографический список

1. Тененёв В.А. Решение задачи многокритериальной оптимизации генетическими алгоритмами // Интеллектуальные системы в производстве . 2006. №2. С. 103-109.
2. Брестер К.Ю., Семенкин Е.С. О решении задач многокритериальной оптимизации самонастраивающимся генетическим алгоритмом // Актуальные проблемы авиации и космонавтики. 2012. №8. С. 290-291.
3. Паротькин Н.Ю., Жуков В.Г. О решении задачи многокритериальной оптимизации дифференцированным генетическим алгоритмом // Актуальные проблемы авиации и космонавтики. 2013. №9. С. 337-338.
4. Возыка В.В. Построение дерева Штейнера генетическим алгоритмом // Известия ЮФУ. Технические науки . 2002. №3 (26). С. 213-214.
5. Емельянова Т.С. Об одном генетическом алгоритме решения транспортной задачи // Известия ЮФУ. Технические науки. 2007. №1 (73). С. 65-70.
6. Емельянова Т.С. Об одном генетическом алгоритме решения транспортной задачи с ограничением по времени // Известия ЮФУ. Технические науки. 2008. №4 (81). С. 45-50.
7. Тененёв В.А., Тененёва А.В. Обучение нечетких нейронных сетей генетическим алгоритмом // Интеллектуальные системы в производстве. 2010. №1. С. 76-85.
8. Lvovich I.Ya., Lvovich Ya.E., Preobrazhenskiy A.P., Choporov O.N., Sakharov Yu.S. The Use of Genetic Algorithm for Construction Objects with Necessary Average Values Scattering Characteristics // Procedia Computer Science. 2017. №103. С. 378-383.
9. Tsoukalas V.D., Fragiadakis N.G. Prediction of occupational risk in the shipbuilding industry using multivariable linear regression and genetic algorithm analysis // Safety Science. 2016. №83. С. 12-22.