

О САМОНАСТРОЙКЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА**Хритonenко Д.И.****Научный руководитель д-р техн. наук Семенкин Е.С.
Сибирский государственный аэрокосмический университет
имени академика М. Ф. Решетнева,**

Рассматривается самонастройка генетического алгоритма для решения задачи однокритериальной безусловной оптимизации. Разработанная программа протестирована на ряде задач и сравнивается с базовым алгоритмом.

При решении задач с помощью генетических алгоритмов (ГА) возникает проблема выбора их параметров. Для каждой решаемой задачи существует свой, оптимальный в некотором смысле набор параметров (по времени работы, числу вычислений целевой функции и т.д.), который в свою очередь может изменяться во время работы алгоритма. Разумеется, изначально ни набор этих параметров, ни закон их изменения (если он вообще существует) неизвестен. Таким образом, перед нами ставится задача в отыскании этих неизвестных параметров, с целью увеличения производительности программы.

В данной работе настройка параметров будет производиться на основе работы А.С. Мясникова: «Островной генетический алгоритм с динамическим распределением вероятностей выбора генетических операторов».

Выделим 3 группы (острова) операторов у ГА: селекции, скрещивания, мутации. Каждая группа представляет собой набор следующих операторов:

Группы операторов	Селекция	Скрещивание	Мутация
Операторы	<ul style="list-style-type: none"> • Пропорциональная • Ранговая • Турнирная 	<ul style="list-style-type: none"> • Одноточечное • Двухточечное • Равномерное 	<ul style="list-style-type: none"> • Слабая • Средняя • Сильная

Таблица 1. Разбиение операторов ГА по группам

Внутри каждой группы операторы будут конкурировать за возможность использования на следующей итерации алгоритма.

Введем понятие прогрессивной хромосомы. *Прогрессивная хромосома* – хромосома, которая превосходит по пригодности среднее значение пригодности в текущей популяции. Необходимо также сделать утверждение: количество прогрессивных хромосом, полученных при помощи оператора селекции равно суммарному числу прогрессивных хромосом, полученных при проведении скрещивания и мутации. Вероятность выбора оператора внутри группы будет высчитываться следующим образом:

$$P_i = \frac{q_i}{\sum_j q_j}$$

q_i – число прогрессивных хромосом, полученных при помощи i -го оператора. При работе алгоритма на каждой итерации работает лишь один набор генетических операторов. Поэтому необходимо некоторое число итераций для накопления выборки, на основании которой будут пересчитаны вероятности.

Тестирование разработанной программы проводилось на ряде задач для следующих размерностей: 2,3,5,10,20,40. Для тестирования и получения результатов использовалась программа (которая поддерживается разработчиками и в настоящее время), предоставленная с того же сайта, где находится данный набор тестовых задач и результаты тестирования других алгоритмов, что исключает неверное представление

результатов. Результаты тестирования, а также сравнения с базовым алгоритмом (Simple GA) и методом Монте-Карло представлены ниже:

Алгоритм	Размерность:	2D	3D	5D	10D	20D	40D
	Островной ГА	147	278	504	887	5651	12000
	Simple GA	556	1903	2160	1704	18336	18425
	Monte Carlo	3550	439674	361942	N/A	N/A	N/A

Таблица 2. Результаты тестирования алгоритмов для точности 0.1

Алгоритм	Размерность:	2D	3D	5D	10D	20D	40D
	Островной ГА	380	1380	2695	5543	9728	10390
	Simple GA	3032	2108	2912	6052	15300	21160
	Monte Carlo	35418	243174	N/A	N/A	N/A	N/A

Таблица 3. Результаты тестирования алгоритмов для точности 0.001

Критерий эффективности:

$$I = \frac{ERT}{ERT_{best}}$$

Здесь ERT – число вычислений целевой функции для достижения заданной точности (по значению этой целевой функции), ERT_{best} – соответственно лучшее число вычислений целевой функции, которое предоставляется программой. В данной работе проводилось усреднение критерия эффективности по всем функциям (их количество – 24). Стоит заметить, что предложенный модифицированный ГА выигрывает на большинстве тестовых задач у базового алгоритма – Simple GA, а, следовательно, данную модификацию можно считать успешной.

Предложенный подход не позволяет настраивать такие параметры как: *размер популяции*, *размер турнира* и т.д. В дальнейшем планируется перенос данного подхода на алгоритм генетического программирования, эффективность которого будет оценена на ряде задач классификации.