УДК 004

А. В. ГАГАРИН

МОДИФИЦИРОВАННЫЕ ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ С ПРИМЕНЕНИЕМ НЕЙРОННЫХ СЕТЕЙ И ВЕРТИКАЛЬНЫХ СУБПОПУЛЯЦИЙ

Представлены модифицированные генетические алгоритмы, предназначенные для повышения эффективности поиска решения стандартным генетическим алгоритмом. В первом подходе используется радиально-базисная нейронная сеть, которая аппроксимирует поверхность целевой функции. Нейронная сеть обучается по ходу оптимизации и используется для прогноза предположительно лучшего решения, которое добавляется в популяцию потомков ГА. Второй алгоритм предназначен для оптимизации целевых функций, представляющих собой суперпозицию функций меньшей размерности. Для этого популяция разбивается на вертикальные субпопуляции и проводится параллельная оптимизация каждой из вложенных в целевую функций. Результаты тестов подтвердили эффективность обоих предложенных алгоритмов. *Генетические алгоритмы; гибридные генетические алгоритмы; искусственные нейронные сети; оптимизация функций*

ВВЕДЕНИЕ

В последнее время для решения задач оптимизации широко применяются генетические алгоритмы (ГА), к преимуществам которых перед традиционными методами локального поиска можно отнести отсутствие ограничений на оптимизируемую функцию и ее производные, способность выходить из локальных минимумов за счет использования целой популяции решений, отсутствие необходимости в априорной информации о предметной области [1,2,7]. Однако вычисление целевых функций зачастую является очень ресурсоемкой задачей и появляется необходимость в дополнительном увеличении скорости поиска ГА, руководствующегося лишь некоторыми эвристическими правилами. Развитие ГА в настоящее время идет по многим направлениям. К классическим можно отнести усовершенствования основных операторов ГА – селекции, скрещивания и мутации [1,8,9], распараллеливание [10], разработку самонастраивающихся ГА [11]. Одним из наиболее перспективных направлений является создание гибридных схем, когда ГА работает в паре с другим алгоритмом оптимизации, например, градиентным [5]. Такая комбинация позволяет ускорить сходимость робастного ГА методами, обычно использующими некоторое представление о поверхности оптимизируемой функции.

В данной работе предлагается гибридный ГА, в котором для построения этой поверхности используется радиально-базисная нейронная сеть (РБНС), способная аппроксимировать произвольные непрерывные функции, заданные на единичном гиперкубе [3,6]. РБНС выступает в качестве идентификатора неизвестной системы, а обучающая выборка для нее генерируется самим ГА. Отличительной особенностью предложенного алгоритма является его универсальность и полное использование информации, появляющейся в процессе оптимизации, для обучения РБНС.

Также предложена модификация стандартного ГА, позволяющая ускорить оптимизацию целевых функций, аргументами которых являются другие функции меньшей размерности. В этом случае популяция разбивается на вертикальные субпопуляции и проводится параллельная оптимизация каждой из вложенных в целевую функций.

Автор благодарит Надеждина О. В. и Савичева В. И. за неоценимую помощь, оказанную при разработке предложенных алгоритмов, постановке задач и обсуждении полученных результатов.

1. МОДИФИЦИРОВАННЫЙ ГА С ПРИМЕНЕНИЕМ НЕЙРОННОЙ СЕТИ

В последнее время появилось много работ, использующих совместно ГА и НС [12,13,14,15]. В них НС используется как идентификатор неизвестной системы, т. е. обучается моделировать реальное поведение исследуемого объекта. Множество обучающих примеров для НС, считается заданным до начала оптимизации (обычно реальная история наблюдения за объектом). Во время самой оптимизации значения целевой функции считаются неизвестными и вместо них используется нейросетевой прогноз. Затем проверяется оптимальность найденных параметров путем вычисления реальной целевой функции.

Однако такой способ использования НС означает потерю универсальности, работает только при доступном заранее обучающем множестве и игнорирует новые данные, появляющиеся в процессе оптимизации. В данной работе предложен гибридный универсальный ГА, который снабжает РБНС обучающими примерами непосредственно в процессе работы. На каждой итерации формируется множество пар $\{\mathbf{X}_i, u_i\}$, где $\mathbf{X}$ – вектор переменных (параметров), u – соответствующее ему значение целевой функции. Эти данные накапливаются и используются для обучения РБНС. Далее выполняется поиск лучшей особи на поверхности решений, построенной нейронной сетью. Для этого используется дополнительный ГА (рис. 1).

Предложенный гибридный ГА не зависит от истории наблюдений за объектом (хотя эту информацию можно использовать для предварительного обучения НС) и может быть применен для решения любой задачи оптимизации. С увеличением количества итераций ГА нейросетевой прогноз будет улучшаться, так как увеличивается накопленное количество известных точек на поверхности решений.

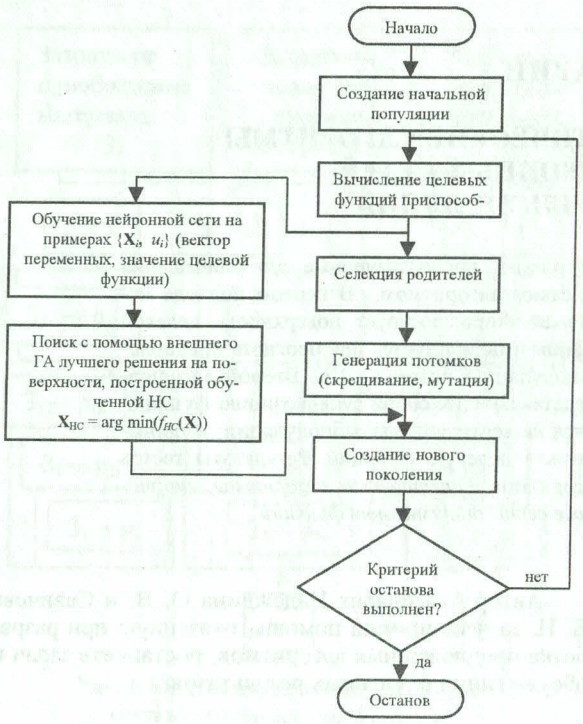

Рис. 1. Блок-схема предложенного гибридного ГА с использованием НС

2. МОДИФИЦИРОВАННЫЙ ГА С ВЕРТИКАЛЬНЫМИ СУБПОПУЛЯЦИЯМИ

В случае, когда целевая функция представляет собой суперпозицию сложных функций от меньшего числа переменных, задача оптимизации может быть переформулирована в пространстве меньшей размерности, что существенно увеличивает скорость поиска ГА.

Предположим, что целевая функция $f(x_1, x_2, \ldots, x_n)$ на самом деле имеет вид:

$$f(x_1, x_2, \ldots, x_n) = g\big(f_1(x_1, x_2, \ldots, x_i), f_2(x_{i+1}, x_{i+2}, \ldots, x_j), \ldots, f_m(x_k, x_{k+1}, \ldots, x_n)\big)$$

$$\text{или } f(\mathbf{X}) = g\big(f_1(\mathbf{X}^1), f_2(\mathbf{X}^2), \ldots, f_m(\mathbf{X}^m)\big). \quad (1)$$

Хромосома для ГА формируется в виде $(x_1, x_2, \ldots, x_n) = (\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^m)$. Очевидно, для вычисления результирующего значения целевой функции $f(\mathbf{X}) = u$ сначала вычисляются значения функций $\{f_1(\mathbf{X}^1) = u_1, f_2(\mathbf{X}^2) = u_2, \ldots, f_m(\mathbf{X}^m) = u_m\}$, над которыми затем выполняется преобразование с помощью функции $g()$. Задача состоит в том, чтобы при условии (1), используя результаты промежуточных расчетов $\{u_1, u_2, \ldots, u_m\}$, увеличить скорость поиска ГА.

В данной работе для решения этой задачи предлагается метод «вертикальных» субпопуляций (ГА-ВСП). Представим популяцию в виде матрицы, в строках которой находятся хромосомы. Предложенный метод предполагает вертикальное разбиение этой матрицы, т.е. субпопуляция определяется подмножеством генов хромосомы. Каждая субпопуляция соответствует своей функции f_i.

Суть метода заключается в политике случайной замены отрезков хромосом $\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^m$ на результаты оптимизации соответствующих вложенных функций. Для этого из матрицы популяции на каждой итерации главного ГА динамически формируются вертикальные субпопуляции. Далее, в каждой из них производится отбор, скрещивание и мутация для формирования нового «субпоколения». Отбор в субпопуляциях строится на основе рассчитанных промежуточных значений $\{u_1, u_2, \ldots, u_m\}$. Полученные потомки затем заменяют в случайно выбранных хромосомах из главной популяции соответствующие подмножества генов. Последний шаг будем называть миграцией. Миграция происходит в матрицу хромосом-потомков главного ГА.

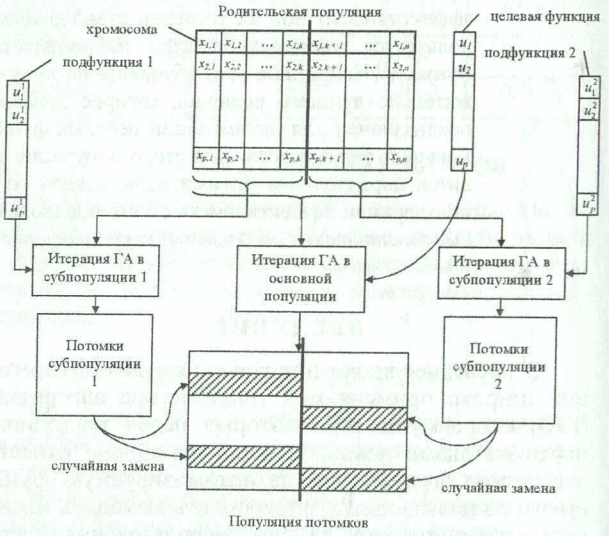

Рис. 2. Итерации ГА с двумя вертикальными субпопуляциями

Описанный алгоритм гарантирует сохранение глобальной устойчивости поиска наряду с ускорением, обеспечиваемым меньшей размерностью вложенных функций.

ГА-ВСП помимо разбиения $(\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^m)$ хромосомы на вертикальные субпопуляции требует задания долей мигрирующих из каждой субпопуляции генов. При стремлении долей к нулю, поведение алгоритма стремится к поведению обыкновенного ГА (случай, когда в целевой функции невозможно выделить подфункции или такая информация недоступна). Если же доли миграции стремятся к единице, поведение алгоритма приближается к поведению семейства раздельных параллельных ГА с совмещением результатов в одну хромосому (можно выделить независимые подфункции без взаимовлияния, функция $g()$ имеет простую форму). На практике при задании долей следует руководствоваться имеющимися априорными сведениями, учитывая два этих крайних случая.

3. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ ГА С ВЕРТИКАЛЬНЫМИ СУБПОПУЛЯЦИЯМИ (ГА + ВСП)

Для тестирования производительности алгоритмов использовалась задача минимизации тестовой функции. Результаты усреднялись по 5 запускам ГА. Параметры всех генетических алгоритмов устанавливались такими: размер популяции – 90 особей, вероятность скрещивания 0,6, вероятность мутации 0,1, оператор скрещивания – SBX [9], оператор мутации – неравномерная мутация Михалевича с параметром 5 [8], элитная стратегия отбора с выживани-

ем двух лучших особей. На каждом запуске выполнялось 1000 итераций ГА.

Для тестирования вертикальных субпопуляций были составлены две целевые функции, которые требуется минимизировать:

$$p_1(\mathbf{X}) = g_1(f_1(\mathbf{X}^1), f_2(\mathbf{X}^2), f_3(\mathbf{X}^3)) = \sum_{i=1}^{3} f_i(\mathbf{X}^i),$$

$$p_2(\mathbf{X}) = g_2(f_1(\mathbf{X}^1), f_2(\mathbf{X}^2), f_3(\mathbf{X}^3)) = \frac{f_1(\mathbf{X}^1)}{1+f_2(\mathbf{X}^2)} + f_3(\mathbf{X}^3),$$

$$\mathbf{X} = (\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3),$$

где $f_i(\mathbf{X}_i)$ – однотипные тестовые подфункции. Размерность вектора параметров $\mathbf{X}$ была принята равной 30, размерность каждого из векторов $\mathbf{X}_i$ равнялась 10.

Были протестированы следующие подфункции:

1) функция Расстригина:

$$f(\mathbf{X}) = 10n + \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i)),\ n = 10.$$

Так как функция p_1 является простой суммой своих подфункций, для нее доли миграции из всех трех субпопуляций были приравнены 0,8. Для p_2 очевидно, что минимизация во второй субпопуляции должна происходить как можно медленнее. Минимизация f_2 в данном случае оказывает влияние лишь на динамику, т.к. в точках минимума она может принимать любые значения. Исходя из этих соображений, доли миграции для трех субпопуляций были определены соответственно как 0,8, 0,1, 0,8.

Результаты минимизации с шагом в 200 итераций ГА представлены в табл. 1:

Таблица 1

Итерации / Алгоритм	200	400	600	800	1000
p_1					
ГА	33.2511	11.8482	9.3599	9.3527	9.3526
ГА + ВСП	2.0131	2.0294e-005	1.5908e-008	2.0705e-011	0
p_2					
ГА	0,3237	0.0173	0.0133	0.0133	0.0133
ГА + ВСП	0.0067	3.8346e-007	5.2174e-010	3.0290e-012	0

Очевидно, что при минимизации обеих тестовых целевых функций, метод вертикальных субпопуляций показал абсолютное превосходство над обычным ГА. Скорость движения к глобальному минимуму при использовании ВСП была намного выше, кроме того, этот минимум был достигнут, в то время как обычный ГА не смог улучшить свои показатели, начиная с 600-й итерации на второй тестовой функции.

2) функция Розенброка

$$f(\mathbf{X}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2],\ n = 10.$$

Доли миграции вертикальных субпопуляций для функции Розенброка были взяты такими же, как в предыдущем случае.

Результаты минимизации с шагом в 200 итераций ГА представлены в табл. 2:

Таблица 2

Итерации / Алгоритм	200	400	600	800	1000
p_1					
ГА	60.3539	39.5759	32.3703	22.4925	22.4174
ГА + ВСП	31.1646	19.7228	17.1595	14.5148	14.2964
P_2					
ГА	6.2098	5.9100	5.6581	5.3030	5.2518
ГА + ВСП	5.0016	4.8183	4.59	4.2385	4.1838

Необходимо отметить, что функция Розенброка имеет выраженный овражный характер и является одной из самой сложных для оптимизации. Однако метод вертикальных субпопуляций и в этом случае подтвердил свою эффективность.

4. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ ГИБРИДНОЙ СХЕМЫ ГА + НС

Для тестирования гибридного ГА использовались тестовые подфункции, описанные в предыдущем параграфе. Размерность каждой функции была равна двум. Параметры ГА были аналогичны параметрам из предыдущих экспериментов.

Алгоритм обучения РБНС заключался в начальном нахождении центров каждой радиальной функции с помощью быстрой кластеризации по *k* средним (*k-means*), после чего начиналась фаза обучения по алгоритму обратного распространения ошибки. Использовалась модифицированная версия алгоритма *k-means*, всегда сходящаяся к квазиоптимальным положениям центров [16]. Для дальнейшего обучения использовалась простая и эффективная модификация стандартного алгоритма обратного распространения, которая учитывает только знаки градиентов функции ошибки, а их значения игнорируются. Модификация известна под названием RPROP и детально описана в работе [17].

Ниже приведены графики оптимизации функции Расстригина от 2 переменных стандартным и гибридным ГА с использованием НС на протяжении первых 20 итераций.

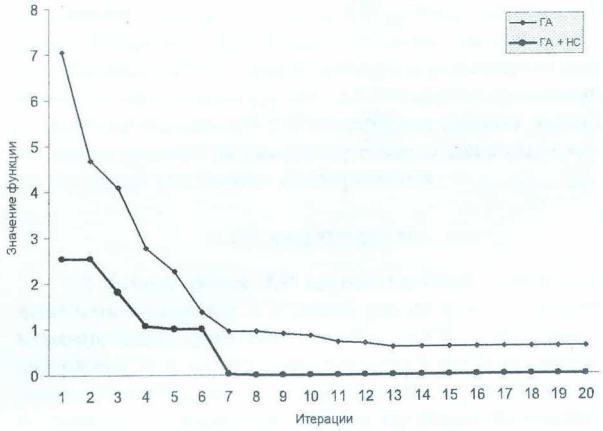

Рис. 3. Итерации ГА с двумя вертикальными субпопуляциями

ЗАКЛЮЧЕНИЕ

Предложенные модифицированные ГА позволяют значительно сократить количество итераций, необходимых для достижения заданной точности решения либо найти лучшее решение по сравнению со стандартным ГА при ограничении по времени. В общем случае, существенно уменьшается количество расчетов значений целевой функции. Это особенно важно в задачах оптимизации, где такие расчеты обходятся очень дорого (например, включают в себя численное моделирование объекта исследований). К таким задачам относится, в частности, определение неизвестных гидродинамических параметров нефтяного пласта. Регионализация этих параметров позволяет эффективно применять ГА с ВСП. Использование предложенных в данной работе алгоритмов в

нефтяной отрасли – задача дальнейших исследований.

СПИСОК ЛИТЕРАТУРЫ

1. **Гладков, Л. А.** Генетические алгоритмы / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик; ред. В. М. Курейчика. М. : Физматлит, 2006. 320 с.

2. **Вороновский, Г. К.** Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г. К. Вороновский, К. В. Махотило, С. Н. Петрашев, С. А. Сергеев. Харьков : Основа, 1997. 112 с.

3. **Хайкин, С.** Нейронные сети: полный курс / С. Хайкин. 2-е изд. М.: Издательский дом «Вильямс», 2006. 1104 с.

4. **Осовский, С.** Нейронные сети для обработки информации / С. Осовский М. : Финансы и статистика, 2004. 344 с.

5. **Тененев, В. А.** Гибридный генетический алгоритм с дополнительным обучением лидера / В. А. Тенев, Н. Б. Паклин // Интеллектуальные системы в производстве. Ижевск : Изд-во ИжГТУ, 2003. № 2. С. 181–206.

6. **Poggio, T.** A theory of networks for approximation and learning / T. Poggio, F. Girosi // Proc. of IEEE. Vol. 78. P. 1481–1497.

7. **Whitley, D.** A genetic algorithm tutorial / D. Whitley // Statistics and Computing. 1994. Vol. 4. P. 65–85.

8. **Herrera, F.** Tackling real-coded genetic algorithms: operators and tools for the behavior analysis / F. Herrera, M. Lozano, J. Verdegay // Artificial Intelligence Review. 1998. Vol. 12. № 4.

9. **Herrera, F.** Hybrid crossover operators for real-coded genetic algorithms: an experimental study // F. Herrera, M. Lozano, A. Sanchez // Soft Computing. 2005. Vol. 9, No. 4.

10. **Whitley, D.** The island model genetic algorithm: on separability, population size and convergence / D. Whitley, S. Rana, R.B. Heckendorn // J. of Computing and Information Technology. 1999. Vol. 7 (1). P. 33–47.

11. **Eiben, A. E.** Parameter control in evolutionary algorithms / A. E. Eiben, R. Hinterding, Z. Michalewicz // IEEE Trans. Evolutionary Computation. 1999. № 3 (2). P. 124–141.

12. **Javadi, A. A.** A hybrid intelligent genetic algorithm / A. A. Javadi, R. Farmani, T. P. Tan // Advanced Engineering Informatics. 2005. № 19.

13. **Ling Wang.** A hybrid genetic algorithm-neural network strategy for simulation optimization / Ling Wang // Applied Mathematics and Computation. 2005. № 170.

14. **Jan-Tai Kuo.** A hybrid neural-genetic algorithm for reservoir water quality management / Jan-Tai Kuo, Ying-Yi Wang, Wu-Seng Lung // Water research. 2006. № 40.

15. **Xiaodong Huo.** A novel channel selection method for CANDU refueling based on the BPANN and GA techniques / Xiaodong Huo, Zhongsheng Xie // Annals of nuclear energy. 2005. № 32.

16. **Chinrungrueng, C.** Optimal adaptive k-means algorithm with dynamic adjustment of learning rate / C. Chinrungrueng, C. H. Sequin // IEEE Trans. on Neural Networks. № 6. P. 157–169.

17. **Riedmiller, M.** A direct adaptive method for faster backpropagation learning: the RPROP algorithm / M. Riedmiller, H. Braun // In Proceedings of the IEEE International Conference on Neural Networks. 1993. P. 586–591.