

ИНСТРУМЕНТАЛЬНАЯ СРЕДА ДЛЯ ИССЛЕДОВАНИЯ ЭВОЛЮЦИОННЫХ СТРАТЕГИЙ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВЫХ МЕТАМОДЕЛЕЙ

Павел Афонин

Аннотация: В статье представлена инструментальная среда для исследования эволюционных стратегий, которые используют механизм аппроксимации целевой функции с помощью аппарата нейронных сетей. Приводится описание алгоритма эволюционной стратегии и подходы к построению метамоделей. Рассмотрены существующие на сегодняшний день алгоритмы оптимизации на основе эволюционных стратегий и метамоделей. Отмечается актуальность применения механизмов адаптации в таких алгоритмах. Описаны основные функции и возможности инструментальной среды. Средством реализации является программный пакет MatLab v.7.1.

Ключевые слова: инструментальная среда, эволюционная стратегия, нейронная сеть, метамодел, оптимизация.

Conference: The paper is selected from XVth International Conference "Knowledge-Dialogue-Solution" KDS 2009, Varna, Bulgaria, June-July 2009

Введение

В последние несколько лет эволюционные алгоритмы нашли широкое применение для решения большого числа сложных задач оптимизации, таких как: задачи оптимального проектирования, задачи оптимального управления, задачи оптимизации на основе имитационного моделирования и многих других задач, требующих применения методов глобальной оптимизации. Однако, во многих случаях, требуется большое число расчетов целевой функции для сходимости эволюционного алгоритма. Проблема усугубляется тем, что для многих задач целевая функция представлена неявно, т.е. в виде некоторой расчетной модели, имитационной модели или компьютерной программы, что требует проведения длительных расчетов, которые во многих случаях невозможно выполнить за приемлемое время.

Основным способом решения данной проблемы является использование метамоделей. Метамоделю принято называть приближенную математическую модель, полученную в результате экспериментов с моделью системы с целью замещения последней при оптимизации. Применяются следующие подходы к построению метамоделей: полиномы, кригинг модели и нейронные сети.

Таким образом, актуальной задачей является разработка эффективных эволюционных алгоритмов оптимизации, которые используют метамодел и, как следствие, разработка инструментария для проведения исследования таких алгоритмов.

В статье приводится описание инструментальной среды для исследования эволюционных стратегий, которые используют механизм аппроксимации целевой функции с помощью аппарата нейронных сетей. В качестве эволюционного алгоритма выбрана эволюционная стратегия, потому что этот алгоритм работает достаточно быстро и обладает возможностью адаптации параметров. В качестве метамоделей используются нейронные сети, поскольку они являются мощным аппаратом для аппроксимации сложных зависимостей. В статье дано описание алгоритма эволюционной стратегии, подходы к построению метамоделей и существующие алгоритмы оптимизации на основе эволюционных стратегий и метамоделей. Отмечается актуальность применения механизмов адаптации в таких алгоритмах. Инструментальная среда реализована с помощью программного средства MatLab v.7.1. Представлены итоги работы и планы для будущей работы.

Описание эволюционной стратегии

Задачей эволюционной стратегии (ЭС) является минимизация целевой функции $F(x_1, x_2, \dots, x_n)$, где x_i – вещественные переменные ($i=1, 2, \dots, n$). В ЭС каждая особь характеризуется:

- функцией пригодности (ФП), которая зависит от целевой функции оптимизационной задачи;
- строкой-хромосомой, включающей:
 - вектор $x(x_1, x_2, \dots, x_n)$, который представляет собой некоторое решение оптимизационной задачи;
 - среднеквадратическое отклонение (шаг мутации) $\sigma(\sigma_1, \sigma_2, \dots, \sigma_m)$, $1 \leq m \leq n$, от которого зависит величина мутации;
 - угол ротации $\alpha(\alpha_1, \alpha_2, \dots, \alpha_k)$, $\alpha_j \in [-\pi, \pi]$, $k = n^*(n-1)/2$, который является необходимым параметром для реализации коррелируемой мутации, позволяющей учитывать ландшафт целевой функции.

В процессе работы алгоритма ЭС происходит адаптация значений шага мутации и угла ротации. Таким образом, происходит двухуровневый процесс оптимизации: на одном уровне осуществляется поиск оптимального значения переменных, на другом – настройка параметров ЭС.

Далее опишем общую схему эволюционной стратегии [3]:

Шаг 1. *Инициализация.* Задается начальная популяция из μ особей. В большинстве случаев используется случайная генерация начальной популяции по равномерному закону распределения. Также возможно равномерное задание популяции, задание в области предполагаемого оптимума, задание методом латинских квадратов и др. Значение шага мутации определяется, как правило, разработчиком. Значение угла ротации выбирается случайным образом на интервале $[-\pi, \pi]$ с помощью равномерного закона распределения.

Шаг 2. *Расчет ФП особей.* Для всех μ особей популяции осуществляется расчет ФП. Данный шаг выполняется только в случае применения $(\mu+\lambda)$ -селекции (см. шаг 6).

Шаг 3. *Скрещивание.* Результатом данного шага является генерация λ потомков из μ родителей. Для получения одного потомка сначала происходит случайный выбор двух родителей из популяции с вероятностью $1/\mu$ для каждого. Затем осуществляется рекомбинация, при которой потомок наследует фенотипические признаки родителей. Рекомбинация реализуется для всей строки-хромосомы, включая шаг мутации и угол ротации.

Шаг 4. *Мутация.* Реализуется мутация λ потомков. Сначала производится изменение параметров стратегии: шага мутации с помощью логнормального закона распределения и угла ротации с помощью нормального закона распределения. После этого реализуется мутация переменных решения на основе случайного вектора, полученного с помощью измененных параметров ЭС:

$$\sigma'_i = \sigma_i \cdot \exp(\tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1));$$

$$\alpha'_j = \alpha_j + \beta \cdot N(0, 1);$$

$$x' = x + N(0, C');$$

где $N(0, 1)$ – значение случайной величины, полученное с помощью нормального закона распределения с математическим ожиданием 0 и среднеквадратическим отклонением 1; τ' – постоянный коэффициент (рекомендуемая величина: $\tau' \approx 1/(2 \cdot n)^{1/2}$); τ – коэффициент, варьируемый при каждой мутации (рекомендуемая величина: $\tau \approx 1/(2 \cdot n^{1/2})^{1/2}$); β – постоянный коэффициент (рекомендуемая величина: $\beta \approx 5^\circ$); C' – ковариационная матрица, полученная после мутации значений σ и α ; $N(0, C')$ – случайный вектор, полученный из ковариационной матрицы n -мерного нормального распределения.

Шаг 5. *Расчет ФП потомков.* Осуществляется расчет ФП для λ потомков текущей популяции.

Шаг 6. *Селекция.* Существует два типа селекции: (μ, λ) -селекция и $(\mu+\lambda)$ -селекция. В случае (μ, λ) -селекции в следующую популяцию выбираются лучшие μ родителей (имеющие лучшие значения ФП)

только из множества λ потомков. При $(\mu+\lambda)$ -селекции новая популяция образуется из объединенного множества родителей и потомков. Более предпочтительной является (μ, λ) -селекция, поскольку реализует механизм выхода из локальных оптимумов.

Шаг 7. Проверка условия останова ЭС. Применяются классические критерии останова для популяционных алгоритмов, такие как: максимальное число итераций (расчетов ФП), максимальное время работы алгоритма, число поколений без изменения лучшего значения ФП, малая разница между лучшим и средним значением ФП и др. Если условие останова не выполняется, то осуществляется переход к шагу 3 для реализации следующего поколения.

Способы построения метамоделей

Полиномы. Наиболее часто используются полиномы второй степени. Число коэффициентов такой модели рассчитывается как: $(n+1)*(n+2)/2$, где n – число переменных. Для расчета неизвестных коэффициентов полинома можно использовать метод наименьших квадратов или градиентный метод. Основным недостатком метода наименьших квадратов является значительные временные затраты на расчет коэффициентов модели в случае решения задачи большой размерности.

Кригинг модели. Представляют собой комбинацию глобальной модели и локальных «отклонений»:

$$y(x) = g(x) + Z(x),$$

где: $g(x)$ является функцией глобальной модели для целевой функции; $Z(x)$ представляет собой функцию Гаусса с нулевым математическим ожиданием и ковариацией, моделирующей локальные отклонения от глобальной модели. Обычно, функция $g(x)$ задается полиномом или, во многих случаях, задается как коэффициент β .

Расчет параметров модели осуществляется с помощью метода максимального правдоподобия.

Основным достоинством кригинг моделей является возможность расчета доверительного интервала без дополнительных вычислений. Однако необходимо выполнять матричные преобразования для расчета выхода модели, что значительно увеличивает время вычислений с ростом размерности задачи.

Нейронные сети. Являются мощным аппаратом для аппроксимации сложных зависимостей [4]. Здесь применяются три типа сетей: многослойный персептрон, сети на основе радиальных базисных функций и машины опорных векторов.

Многослойный персептрон прямого распространения состоит из: входного слоя, состоящего из множества сенсорных элементов; одного или нескольких скрытых слоев вычислительных нейронов (как правило, с сигмоидальной функцией активации) и выходного слоя нейронов. Обучение такой сети выполняется с помощью алгоритма обратного распространения ошибки (BP-алгоритм). Для повышения эффективности решения задачи с помощью многослойного персептрона применяют модификации BP-алгоритма и методы оптимизации структуры сети для конкретной задачи.

Сеть на основе радиальных базисных функций (RBF-сеть) состоит из трех слоев. Входной слой включает сенсорные элементы, которые связывают сеть с внешней средой. Промежуточный слой является единственным скрытым слоем сети. Он состоит из радиальных элементов, каждый из которых воспроизводит гауссову поверхность отклика и выполняет нелинейное преобразование входного пространства. Для большинства задач размерность скрытого слоя значительно превышает размерность входного слоя. Выходной слой RBF-сети включает нейроны с линейными функциями активации и выполняет линейное преобразование пространства скрытого слоя нейронов.

Машина опорных векторов представляет собой линейную систему, которая решает задачу разделения объектов в пространстве признаков с помощью гиперплоскости. Теория таких систем исходит из теории статистического обучения. Основными достоинствами машины опорных векторов являются: отсутствие локального минимума ошибки в процессе обучения, а также то, что ошибка обобщения не зависит от размерности задачи.

Эволюционные алгоритмы оптимизации на основе метамоделей

Интеграция метамоделей в эволюционные алгоритмы. Существует два основных подхода к интеграции метамоделей в эволюционные алгоритмы. В первом подходе сначала определяется оптимум по метамоделю, а затем реализуется расчет оптимизируемой функции в точке оптимума. Новые данные используются для построения новой (более точной) метамоделю и процесс определения оптимума и обновления метамоделю повторяется. Второй подход основан на концепции *эволюционного контроля* [5], в рамках которого применяются два метода: эволюционный контроль на уровне особей и эволюционный контроль на уровне поколений. В методе эволюционного контроля на уровне особей часть особей текущей популяции рассчитывается с помощью целевой функции. Для остальных особей в популяции расчет ФП реализуется с использованием метамоделю. В методе эволюционного контроля на уровне поколений все особи некоторой популяции рассчитываются или с помощью целевой функции или с помощью метамоделю. Особи, ФП которых рассчитывается с помощью целевой функции, называются *контролируемые*, а для которых с помощью метамоделю – *неконтролируемые*.

Эволюционный контроль на уровне особей. В настоящее время данный подход представляется интересным и малоисследованным [1, 2, 5, 6]. Здесь главной задачей является определение того, какие особи в каждом поколении будут рассчитываться с помощью целевой функции, а какие с помощью метамоделю. Опишем два базовых метода эволюционного контроля на уровне особей: метод на основе оценки лучших особей (*best* стратегия) и метод с применением предварительного отбора (*pre-selection* стратегия).

В *best* стратегии [5], $\lambda' = \lambda$ потомков рассчитываются с помощью метамоделю и лучшие λ^* рассчитываются с помощью ФП. После обновления метамоделю оставшиеся $\lambda' - \lambda^*$ потомков снова рассчитываются по метамоделю. Лучшие μ особей из λ потомков становятся родителями следующего поколения.

В *pre-selection* стратегии [6], $\lambda' > \lambda$ потомков генерируются из μ родителей с помощью операторов рекомбинации и мутации и затем рассчитываются с помощью метамоделю. Лучшие $\lambda^* = \lambda$ особей предварительно отбираются из λ' потомков и рассчитываются с помощью ФП.

По результатам исследований, описанных в работе [2], сделано важное заключение о том, что стабильность эволюционной стратегии на основе эволюционного контроля на уровне особей может быть улучшена, если родители для следующего поколения выбираются из контролируемых особей, подобно тому, как это реализовано в *pre-selection* стратегии.

Однако на сегодняшний день остается открытым вопрос о том, сколько особей и какие особи текущего поколения должны контролироваться.

Механизмы адаптации. Основным механизмом адаптации является следующее правило: если качество метамоделю *улучшается* в процессе поиска, то *большее* число особей должно рассчитываться с использованием метамоделю.

В работе [6] предложено правило определения числа особей для предварительного отбора λ_{pre} в зависимости от качества селекции, рассчитываемого по метамоделю. В работе [2] описаны механизм адаптации на основе расчета среднеквадратического отклонения, механизм адаптации на основе селекции и механизм адаптации на основе корреляции.

Существует точка зрения [2, 6], что при использовании метамоделю в эволюционных алгоритмах важным является только правильная селекция, а не ошибка аппроксимации метамоделю.

Описание инструментальной среды

На рисунке 1 представлено диалоговое окно инструментальной среды для исследования эволюционных стратегий с использованием нейросетевых метамоделю. Текущая версия поддерживает алгоритмы, реализующие метод эволюционного контроля на уровне особей. Среда разработана в среде MatLab 7.1, программный код функций и алгоритмов содержится в 5-ти m-файлах.

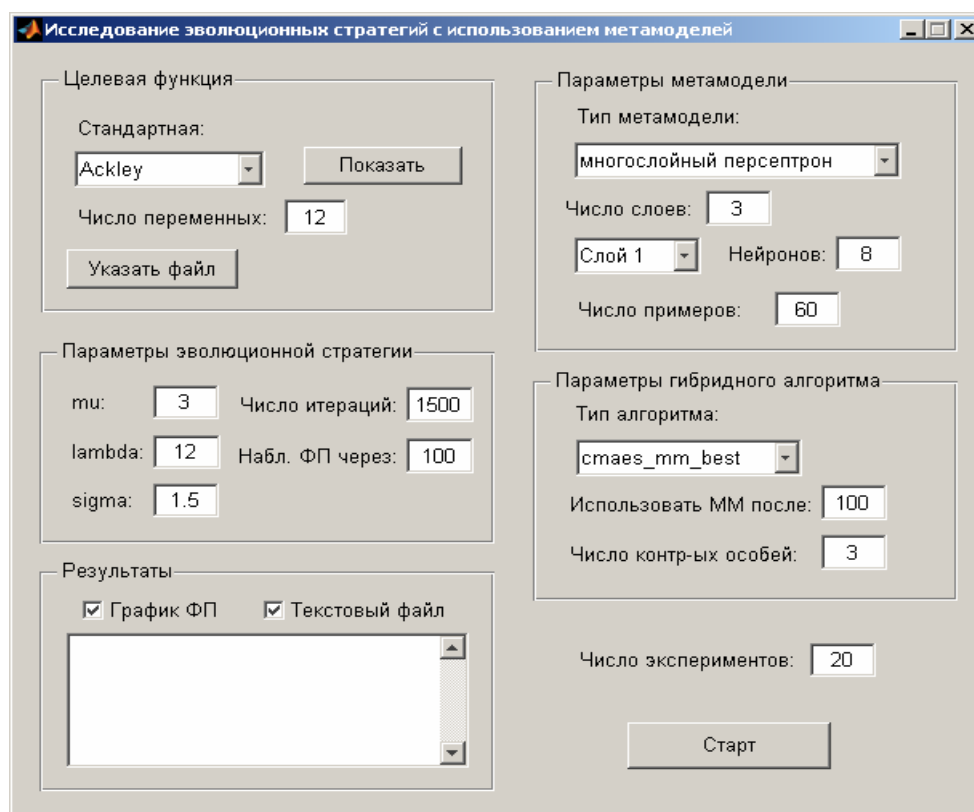


Рис.1. Диалоговое окно инструментальной среды

Опишем возможности и соответствующие блоки инструментальной среды:

Целевая функция. Задаются параметры оптимизируемой функции. Можно выбрать стандартную тестовую функцию (вкладка *стандартная*) или загрузить из файла (указать М-файл, в котором содержится определение функции). Предусмотрены стандартные тестовые функции: *Sphere*, *Schwefel*, *Ackley*, *Rosenbrock*, *Rastrigin* и др. Имеется возможность просмотра в отдельном окне ландшафта (поверхности) стандартных тестовых функций в трехмерной системе координат. В файле можно задать определение как простой функции (функция задается аналитически), так и сложной (функция задается как компьютерная программа, работающая по определенному алгоритму). Также можно задать размерность исследуемой функции (параметр *число переменных*).

Параметры эволюционной стратегии. Задаются параметры алгоритма эволюционной стратегии: тип стратегии (μ, λ) и шаг мутации σ , а также число итераций (расчетов ЦФ) работы алгоритма и интервал, с которым осуществляется наблюдение значений ФП в процессе работы алгоритма, необходимый для сбора статистических данных по результатам отдельных экспериментов.

Параметры метамоделей. Задаются параметры нейросетевой метамоделей. Во вкладке *тип метамоделей* можно выбрать тип нейронной сети, на основе которой будет реализовано построение метамоделей: многослойный перцептрон, сеть на основе радиальных базисных функций, машина опорных векторов. Можно задавать параметры нейронной сети и число примеров обучения. Так, например, для многослойного перцептрона можно задать число скрытых слоев и число нейронов в каждом скрытом слое.

Параметры гибридного алгоритма. Текущая версия поддерживает следующие алгоритмы: *rigestmaes* (базовый алгоритм эволюционной стратегии без использования метамоделей), *pre-selection* стратегия, *best* стратегия и вновь разработанный алгоритм «*my_cmaesmm*», в котором число контролируемых особей для одного поколения может изменяться в зависимости от точности модели, а выбор μ родителей для следующего поколения всегда осуществляется из контролируемых особей. В будущем, планируется опубликовать результаты исследований эффективности данного алгоритма по сравнению с другими

стратегиями. Также исследователю предоставляется возможность задавать число контролируемых особей. Имеется возможность задать число примеров для обучения НС после которого начинается построение метамоделей.

Результаты. Имеется возможность проведения серии экспериментов с заданными параметрами, а также расчет средних значений по результатам серии экспериментов. Планируется реализация расчета доверительных интервалов и среднеквадратических ошибок. Результаты можно представить в виде текстового файла, в котором описаны средние и лучшие значения ФП для каждого эксперимента (имеется возможность задать число экспериментов) через определенное число итераций. В отдельном блоке приводятся результаты работы алгоритмов, которые можно использовать для окончательной оценки их эффективности. Результаты по каждому эксперименту сохраняются в массиве данных, который можно использовать в дальнейшем.

Заключение

Статья посвящена разработке инструментальной среды для исследования эволюционных стратегий, которые используют механизм аппроксимации целевой функции с помощью аппарата нейронных сетей. В качестве эволюционного алгоритма выбрана эволюционная стратегия, а в качестве метамоделей используются нейронные сети. Рассмотрены эволюционные алгоритмы оптимизации на основе метамоделей. Инструментальная среда реализована с помощью программного средства MatLab v.7.1.

В дальнейшем планируется расширение среды и добавление следующих возможностей: включение алгоритмов, реализующих метод эволюционного контроля на уровне поколений, добавление других способов построения метамоделей (полиномы и кригинг модели), включение других эволюционных алгоритмов, в частности, различных схем реализации генетического алгоритма.

Библиография

1. M. Emmerich, A. Giotis, M. Özdenir, T. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. In *Parallel Problem Solving from Nature*, number 2439 of *Lecture Notes in Computer Science*, pp. 362-370, 2002.
2. L. Gräning, Y. Jin, and B. Sendhoff. Individual-based Management of Meta-models for Evolutionary Optimization with Application to Three-Dimensional Blade Optimization. *Evolutionary Computation in Dynamic and Uncertain Environments*, pp. 225-250, 2007.
3. N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2), pp. 159–196, 2001.
4. S. Haykin. *Neural Networks – A Comprehensive Foundation*. Prentice-Hall, 1994.
5. Y. Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation*, 6(5), pp. 481-494, 2002.
6. H. Ulmer, F. Streichert, and A. Zell. Evolution strategies with controlled model assistance. In *Congress on Evolutionary Computation*, pp. 1569–1576, 2004.
7. П. Афонин. Система оптимизации на основе имитационного моделирования, генетического алгоритма и нейросетевых метамоделей // *Proceedings of the XIII-th International Conference «Knowledge-Dialogue-Solutions» – Varna, 2007. – V.1. – P.60-63.*

Информация об авторе

Павел Афонин – к.т.н.; Московский Государственный Технический Университет им. Баумана; 2-я Бауманская, д.5, Москва, Россия; e-mail: pavlafon@yandex.ru