

УДК 519.873

РАЗРАБОТКА МОДЕЛИ ФУНКЦИИ НА БАЗЕ ЭВОЛЮЦИОННОГО АЛГОРИТМА

А.А. Коленчук, А.Р. Комогорова

В работе рассматривается развитие методов эволюционного моделирования, и проводится их обзор, применительно к оптимизации функций. Проводится классификация методов по областям, традиционно использующим эволюционное моделирование и ряд схожих по происхождению решений. Описаны принципы и предпосылки использования приемов взятых из живой природы для оптимизации виртуальных моделей и функций. Вводится разделение метода оптимизации на составляющие части. Приводятся примеры реализации отдельных элементов алгоритма эволюционного моделирования. Анализируются сильные и слабые стороны метода, делаются выводы о предпосылках его использования.

Ключевые слова: оптимизация функций, эволюционное моделирование, популяция, геном, моделирование сложных систем.

С начала 70-х годов прошлого века в задачах синтеза моделей предпринимались попытки получения оптимальных решений, используя эволюционные принципы в моделировании.

Основной тезис эволюционного моделирования – замена процесса моделирования сложного объекта моделированием его эволюции [1–3]. В общем виде эволюционный алгоритм – оптимизационный метод, базирующийся на эволюции популяции особи, применительно к алгоритму модели.

Всё многообразие применений таких методов можно разделить на три класса:

- 1) модели происхождения молекулярно-генетических систем обработки информации;
- 2) модели, характеризующие общие закономерности эволюционных процессов;
- 3) модели искусственной «эволюции» с целью применения метода эволюционного поиска к практическим задачам оптимизации.

Исходя из различных областей применения и используемых методик на практике, задачи оптимизации также делят по типам алгоритмов:

1. Генетические алгоритмы, предназначенные для оптимизации функций дискретных переменных и акцентирующие внимание на рекомбинациях геномов.

2. Эволюционное программирование, ориентированное на оптимизацию непрерывных функций без использования рекомбинаций.

3. Эволюционные стратегии, ориентированные на оптимизацию непрерывных функций с использованием рекомбинаций.

4. Генетическое программирование, использующее эволюционный метод для оптимизации компьютерных программ.

Теперь рассмотрим эволюционное моделирование более конкретизировано [5]. В его основе лежат законы развития органического мира, сформулированные Дарвином [2]:

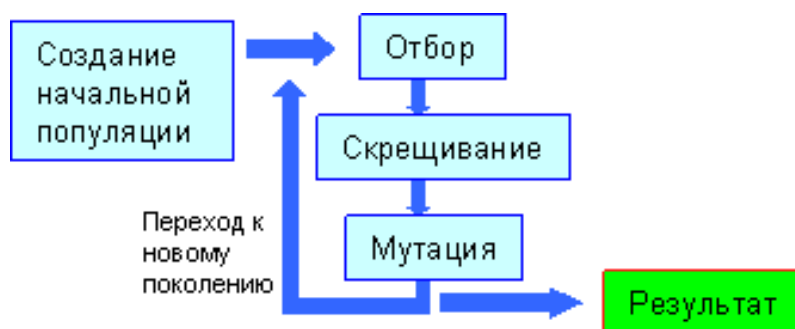
- наследственность, потомки сохраняют свойства родителей;
- изменчивость, потомки почти всегда не идентичны;
- естественный отбор, выживают наиболее приспособленные.

Эволюционные алгоритмы [1] оперируют совокупностью моделей, которые представляют собой одно из решений задачи. Этим алгоритм отличается от большинства других алгоритмов оптимизации, которые оперируют лишь с одним решением, улучшая его. Для удобства оперирования с моделями их структура записывается в виде строки фиксированной длины, состоящей из битов определяющих параметры модели [6]. Также отметим, что каждый запуск алгоритма дает одно решение близкое к оптимальному и таких решений может быть получено множество.

Самым очевидным и доступным для понимания можно считать метод группового учета аргументов [3, 6]. Он заключается в следующем:

- берется последнее поколение моделей;
- генерируется из них по определенным правилам новое поколение, которое становится последним поколением;
- отбирается из них F лучших, где F – ширина отбора (селекции);
- если не выполняется условие прекращения селекции (наступление вырождения), переход на п. 1, иначе лучшая модель объявляется искомым решением задачи;

На рис. представлена функциональная схема работы алгоритма [6].



Алгоритм эволюционного алгоритма оптимизации

Для решения задачи с помощью генетического алгоритма необходимо задать меру качества для каждой модели. Для этой цели используется

функция приспособленности. Функция приспособленности должна принимать неотрицательные значения на ограниченной области определения, при этом совершенно не требуются непрерывность и дифференцируемость, значение этой функции определяет, насколько хорошо подходит особь для решения задачи. Она должна отвечать представлениям разработчика об оптимальной модели.

На каждом шаге моделирования с помощью вероятностного оператора селекции выбираются два решения для получения гибридной модели.

Кратко опишем наиболее распространенные операторы [1, 4]:

1. Пропорциональный отбор. Каждой модели сопоставляют вероятность, равную отношению ее приспособленности к суммарной приспособленности популяции. Затем происходит отбор всех n моделей для дальнейшей обработки.

2. Турнирный отбор. Из популяции, содержащей m моделей, выбирается случайным образом t моделей и наиболее приспособленная модель записывается в промежуточный массив. Эта операция повторяется m раз. Строки в полученном промежуточном массиве затем используются для скрещивания.

3. Отбор усечением. Число моделей для скрещивания выбирается в соответствии с порогом T . Порог определяет, какая доля моделей, начиная с самой приспособленной, будет принимать участие в отборе.

4. Ранговый отбор. Для каждой модели ее вероятность попасть в промежуточную популяцию пропорциональна ее порядковому номеру в отсортированной по возрастанию приспособленности популяции.

5. Элитный отбор. Обязательное сохранение только одной лучшей особи, если она не прошла, как другие, через процесс отбора и мутации.

Существует большое количество разновидностей оператора скрещивания. Простейший одноточечный кроссовер может работать, например следующим образом. Выбирается одна из возможных точек разрыва. Под точкой разрыва понимают участок между соседними битами в строке. Две начальные модели разрываются на два сегмента по этой точке. Затем соответствующие сегменты различных моделей склеиваются и получаются два «генотипа» потомков [5].

После стадии кроссовера идет операция мутации, предназначенная для того, чтобы поддерживать разнообразие моделей в популяции. В простейшем случае в каждой модели, которая подвергается мутации, каждый бит с вероятностью P_m изменяется на противоположный.

После скрещивания и мутации моделей возникает проблема принятия решений: какие из новых моделей войдут в следующее поколение, а какие – нет, и что делать с предыдущими моделями. Возможны варианты её разрешения [6]:

1. Новые модели занимают место предыдущих. После этого наступает следующий этап.

2. Следующая популяция включает в себя предыдущие и новые модели. В простейшем случае, после каждого скрещивания включаются две лучших модели из четверки предыдущего и нового поколения. Более эффективным является механизм вытеснения, который заключается в удалении «похожих» моделей из популяции.

Схождением называется такое состояние популяции, когда все модели практически одинаковы и находятся в области некоторого экстремума.

Одним из эволюционных алгоритмов является СНС-алгоритм (Heterogenous recombination, Cataclysmic mutation), который был предложен Эшелманом и характеризуется следующими параметрами:

Для нового поколения выбираются N лучших. Дублирование не допускается.

Для скрещивания выбирается случайная пара, но не допускается сильное расхождение моделей, и используется разновидность однородного кроссовера.

СНС противопоставляет агрессивный отбор агрессивному кроссоверу, однако все равно малый размер популяции быстро приводит ее к более или менее одинаковым моделям. В таком случае СНС применяет cataclysmic mutation: все модели, кроме самой приспособленной, подвергаются сильной мутации (изменяется около трети битов). Таким образом, алгоритм перезапускается и далее продолжает работу, применяя только кроссовер.

При решении сложных задач особый интерес вызывают параллельные эволюционные алгоритмы, организованные как несколько параллельно выполняющихся процессов [6]. При прямом переносе имеющихся методов на параллельное выполнение возникнут трудности связанные с синхронизацией процессов и обменом информацией о состоянии моделей. Рассмотрим два метода изначально рассчитанных для параллельного выполнения процессов.

Островная модель (island model) заключается в разбиение моделей на подпопуляций. Каждая из подпопуляций будет развиваться отдельно с помощью некоего эволюционного алгоритма. Таким образом, можно сказать, что мы расселили модели по изолированным островам. Изредка процессы будут отправлять друг другу несколькими хороших особей, что позволит обмениваться генетическим материалом.

Из-за малого размера подпопуляций они будут склонны к преждевременной сходимости. Поэтому важно правильно установить частоту миграции. Очень частая миграция приведет к смешению всех подпопуляций, что сведет распараллеливание процессов на нет. В противной ситуации не будет предотвращаться преждевременное схождение подпопуляций.

Ячеистый эволюционный алгоритм (Cellular Genetic Algorithms) заключается в расположении процессов на сетке замкнутой размером. Каждый процесс может взаимодействовать только с четырьмя своими соседями (сверху, снизу, слева, справа). В каждой ячейке находится ровно одна модель. Каждый процесс будет выбирать лучшую модель среди своих соседей, скрещивать с ней модель из своей ячейки и заменять её.

По мере работы такого алгоритма возникают эффекты, похожие на островную модель. Сначала все особи имеют случайную приспособленность. Спустя несколько поколений образуются небольшие области похожих особей с близкой приспособленностью. По мере работы алгоритма эти области растут и конкурируют между собой.

Метод эволюционного моделирования в задачах оптимизации применим прежде всего к сложным системам, прямой анализ которых затруднен в силу большого объема описания, или к моделированию систем строение которых доподлинно не известно [6]. На практике в пользу применения метода говорит его универсальность независимо от вида модели и возможность реализации алгоритма на многопоточной основе.

Библиографический список

1. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. – Х.: ОСНОВА, 1997. – С. 112.
2. Исаев, С.А. Популярно о генетических алгоритмах / С.А. Исаев. – URL: <http://algolist.manual.ru/ai/ga/ga1.php>.
3. Каширина, И.Л. Введение в эволюционное моделирование: учебное пособие / И.Л. Каширина. – Воронеж, 2007. – С. 40.
4. Яминов, Б. Генетические алгоритмы / Б. Яминов. – СПб.: Санкт-Петербургский государственный университет, 2005. – URL: <http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005>.
5. Емельянов, В.В. Теория и практика эволюционного моделирования / В.В. Емельянов, В.В. Курейчик, В.М. Курейчик. – М.: ФИЗМАТЛИТ, 2003.
6. Стариков, А. Лаборатория BaseGroup. Генетические алгоритмы – математический аппарат / А. Стариков. – URL: <http://www.basegroup.ru/genetic/>.

[К содержанию](#)