

Автоматическая генерация параллельных алгоритмов машинного обучения и эволюционных вычислений*

Н.М. Ершов, Н.Н. Попова

Московский государственный университет им. М.В. Ломоносова,
факультет вычислительной математики и кибернетики

Целью проводимых исследований является разработка программной системы для автоматической генерации алгоритмов машинного обучения и эволюционных вычислений [1]. Одной из составляющих этой системы должен быть специальный веб-сервис, предназначенный для автоматической генерации параллельных программ, реализующих классические эволюционные алгоритмы оптимизации [2]. Отличительной особенностью таких методов является то, что все они являются *метаэвристиками*: каждый из этих алгоритмов представляет собой, по сути, целое семейство алгоритмов, объединенных некоторой общей идеей, но сильно отличающихся друг от друга на нижнем уровне (например, реализацией тех или иных операторов). Целью работы разрабатываемой онлайн-службы является предоставление пользователю возможности собрать свой вариант оптимизационного алгоритма, используя для этого набор стандартных схем и элементов. Результатом работы системы должен быть программный код (например, на языке программирования C++) выбранного и настроенного пользователем алгоритма оптимизации. Этот код должен а) быть работающим; б) быть хорошо задокументированным; в) содержать минимальный набор необходимых вспомогательных структур и функций (без использования сложных классов или шаблонов классов и функций). У пользователя системы, таким образом, появляется полный доступ к полученному коду, он может использовать его, например, в качестве некоторой заготовки для разработки своей собственной реализации того или иного алгоритма.

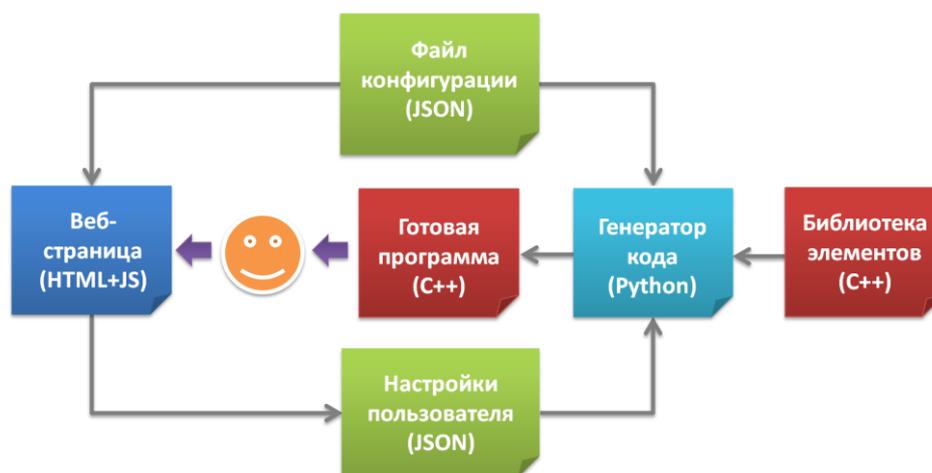


Рис. 1. Архитектура системы

Настоящая работа посвящена разработке той части системы, которая отвечает за автоматическую генерацию *параллельных* кодов рассматриваемых эволюционных и популяционных алгоритмов оптимизации. Известно, что такие алгоритмы обладают хорошей степенью встроенного параллелизма [3], т.к. все они оперируют с популяциями решений, которые эволюционируют практически независимо друг от друга, и, следовательно, могут обрабатываться параллельно. В работе рассматривается несколько подходов к автоматическому распараллеливанию такого рода алгоритмов.

Простейшим подходом к распараллеливанию, не требующим практически никакой модификации исходного последовательного алгоритма, является *островная* модель. При таком под-

* Работа выполнена при финансовой поддержке РФФИ (грант №14-07-00628 А).

ходе на каждом процессоре системы запускается отдельная последовательная версия алгоритма, работающая со своей субпопуляцией. Взаимодействие между субпопуляциями осуществляется с помощью специального оператора *миграции*. Более сложной с точки зрения автоматической генерации кода является модель *master-slave*. В этом случае алгоритм запускается на выделенном процессоре (*master*), а реализация операторов алгоритма, которые могут выполняться независимо для разных решений, производится на рабочих (*slave*) процессорах. Еще один подход к распараллеливанию основан на делении *генома* на фрагменты, с обработкой каждого фрагмента на отдельном процессоре системы (*геномная* модель распараллеливания). Операции, которые требуют доступа ко всему геному (т.е. ко всем компонентам каждого решения), должны выполняться на отдельном процессоре. Примером такой операции является вычисление целевой функции. Также анализируется возможность автоматического построения мелкозернистой параллельной реализации эволюционных алгоритмов в так называемой *клеточной* модели.

Для выполнения автоматической генерации параллельной версии выбранного и настроенного пользователем алгоритма системе может потребоваться некоторая вспомогательная информация, которая должна быть заложена при составлении конфигурационных данных. Например, для реализации островной модели нужны следующие параметры: 1) топология связей между отдельными субпопуляциями (линейная, дерево, двумерная решетка, гиперкубическая и т.п.); 2) частота выполнения миграции; 3) размер миграции. Кроме того, системе может потребоваться информация о типе каждого оператора с точки зрения его параллельной реализации в той или иной модели распараллеливания. Например, если рассматривается геномная модель распараллеливания, то важно знать, является ли данный оператор локальным относительно генома (мутация или скрещивание в генетическом алгоритме при использовании двоичного кодирования) или глобальным (вычисление целевой функции).

В настоящее время разработан прототип системы, поддерживающий работу с наиболее популярными эволюционными и популяционными алгоритмами (генетическими, муравьиными, методом роя частиц и т.д.). Включена возможность генерации параллельных кодов алгоритмов на основе островной и геномной моделей. Ведется работа по реализации других моделей распараллеливания.

Литература

1. Ершов Н., Попова Н. Автоматическая генерация эволюционных алгоритмов под целевую архитектуру вычислительной системы // Тихоновские чтения. Научная конференция, Москва, факультет ВМК МГУ имени М.В. Ломоносова, 27-31 октября 2014 г. Тезисы докладов. — МАКС Пресс Москва, 2014. — С. 11.
2. Ершов Н. М., Попова Н. Н. Разработка онлайн-репозитория эволюционных алгоритмов оптимизации // Научный сервис в сети Интернет: труды XVII Всероссийской научной конференции (21-26 сентября 2015 г., г. Новороссийск). — ИПМ им. М.В. Келдыша М, 2015. — С. 104–108.
3. Omar A.C. Cortes, Eveline de Jesus V. Sa, Jackson A. da Silva, Andrew Rau-Chaplin, An Automatic Code Generator for Parallel Evolutionary Algorithms: Achieving Speedup and Reducing the Programming Efforts, The Ninth International Conference on Advanced Engineering Computing and Applications in Sciences, 2015