

А.В.Гуща. Л.П.Оганян, А.Н.Нардид.

Инструментальное программное средство пректирования генетических алгоритмов (DEvol)¹.

1 Введение

Для многих классов задач алгоритм решения задач еще найден или является слишком дорогим по вычислительным ресурсам. Большинство из этих задач можно представить как задачи оптимизации, где в пространстве возможных решений ведется поиск оптимального. Генетические алгоритмы построены по примеру эволюции в животном мире, где постоянно генерируются решения, проверяются на удачность и либо отбрасываются, либо передаются в будущие поколения. Применяя данный подход к алгоритмам можно получить результат быстрее, чем с помощью полного перебора вариантов. Данное утверждение доказывалось в работе [2]. В данной работе на основе генетических алгоритмов и генетического программирования построен инструмент для поиска решений для широкого класса задач оптимизации. Программный продукт обладает следующими особенностями:

- Возможность написания модулей для расширения возможностей продукта.
- Повторное использование полученных решений для будущих задач.
- Возможность масштабирования с помощью сетей Интернет и кластеризации.

2 Формализация задачи

Для решения задачи с помощью данного программного продукта исходную задачу нужно разделить на несколько компонент:

- Функция оценки. Это алгоритм по которому каждому решению задачи ставится в соответствие некоторое число, которое отображает пригодность и оптимальность решения исходной задачи.
- Множество операторов. Из исходной задачи нужно вычлнить минимальные действия, вычисления, которые могут быть использованы в потенциальных решениях. На основе набора операторов создается внутренний язык описания решений.
- Множество типов. Из исходной задачи выделяются элементарные сущности, которыми могут быть использованы в операторах. Это могут быть числа, строки, логические переменные, сложные сущности материального мира.

После декомпозиции исходной задачи необходимо описать условия и параметры проведения эволюционного процесса. Конкретные значения параметров эволюции выбираются эмпирически или на основе одной из стандартной стратегии. Условия проведения отбора описываются как отдельный модуль, представляющий из себя набор правил, определяющих порядок обработки и окружения решений задач.

¹ В данном материале приводятся результаты исследований, выполненных при поддержке гранта РГНФ №12-04-12039в

3 Описание процесса получения решения задачи

Потенциальные решения задачи называются индивидами. Поведение индивидов кодируется на основе набора операторов и типов для данной задачи. Условия окружения индивидов называются миром, в котором происходит эволюционный процесс поиска лучших индивидов. Индивиды объединяются в популяции, число которых задается в параметрах системы. Эволюция популяций проходит независимо, что позволяет получить разные решения и обработать наибольшее число возможных вариантов.

Далее будет приведен общий алгоритм эволюционного процесса, который может быть изменен для решения специфических задач. Изначально популяции заполняются индивидами, исходный код которых генерируется случайно (Описание внутреннего представления индивидов приведено в разделе 4). Эволюционный процесс разделяется на поколения. В начале каждого поколения проходит один или несколько циклов симуляции окружения индивидов и их поведения, после этого с помощью функции оценки каждому индивиду ставится в соответствие мера приспособленности (*fitness*). На основе этого параметра будет создана популяция для следующего поколения.

Генерация новых индивидов для новой популяции производится с помощью набора генетических операторов: мутация, кроссинговер, копирование. Оператор мутации случайным образом меняет исходный код индивида в нескольких местах. Обеспечивает поддержание генетического разнообразия в популяции. Оператор кроссинговера скрещивает исходные коды двух индивидов и получает два новых индивида. Позволяет находить новые решения на основе предыдущих. Оператор копирования переносит несколько лучших индивидов в новую популяцию, гарантирует, что следующее поколение будет не хуже предыдущего, то есть не произойдет деградации популяции.

После генерации новой популяции наступает новое поколение и процесс повторяется до тех пор, пока не наступит предел поколений (установленный разработчиком) или разработчик сочтет удовлетворительным полученное решение. Схема эволюционного процесса представлена на рисунке 1.

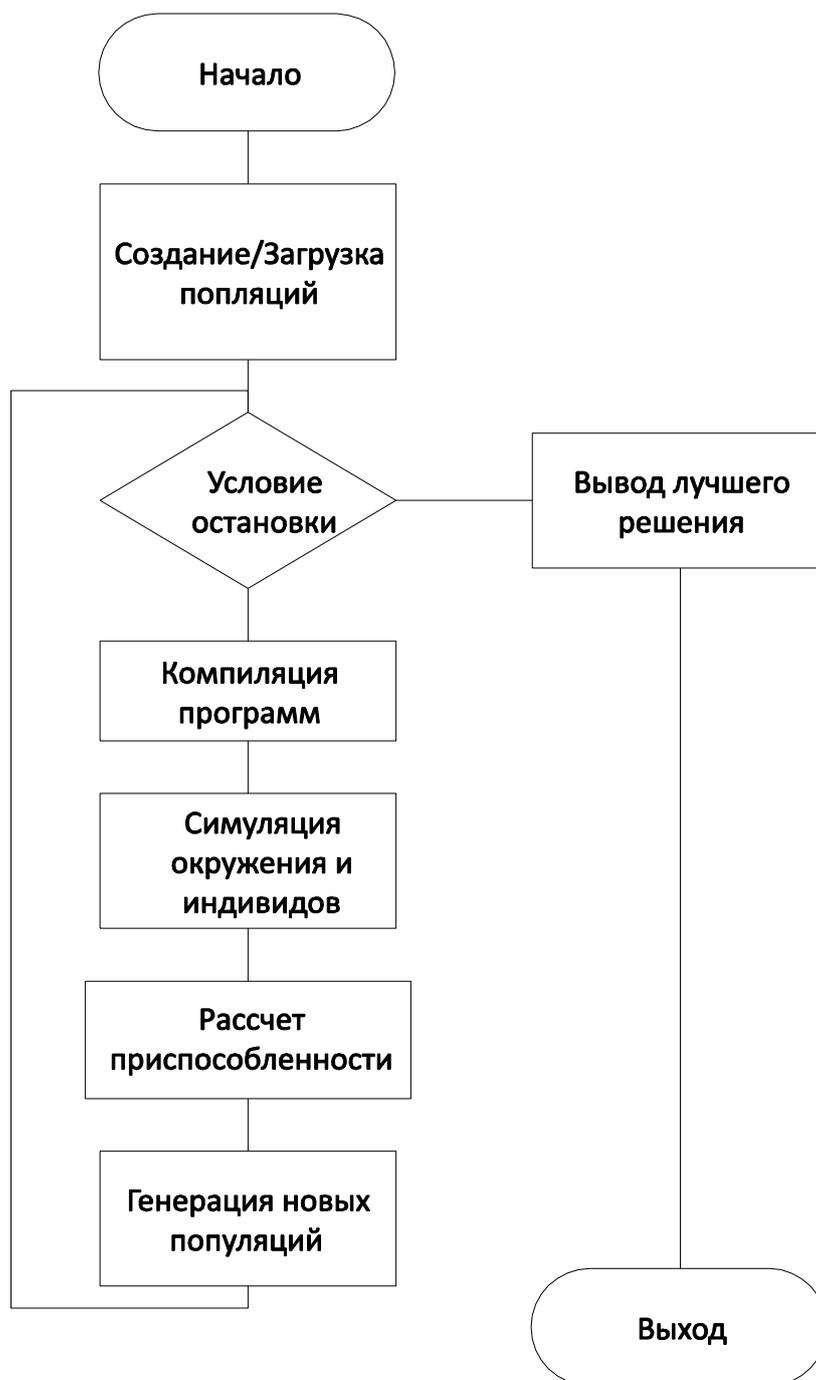


Figure 1: Общая схема эволюционного процесса.

4 Внутренний язык представления индивидов

На основе формализованной задачи создается внутренний проблемно-ориентированный (DSL) язык описания индивидов. Каждая программа является набором деревьев, в узлах которых находятся операторы, в листьях экземпляры типов — операнды. Пример дерева представлен на рисунке 2.

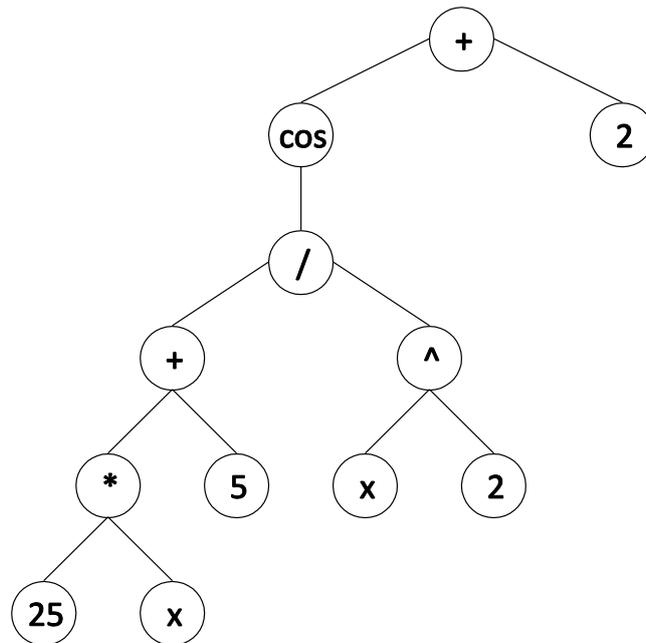


Figure 2: Пример разбиения выражения $\cos\left(\frac{25x+5}{x^2}\right)+2$.

Разработчик описывает глобальные условия для выполнения кода — гены. Каждый ген является контейнером для деревьев и запускается при определенных событиях в окружении индивида или с помощью специальных операторов. Пример разбиения на гены для классической задачи муравья [1] представлен на рисунке 3.

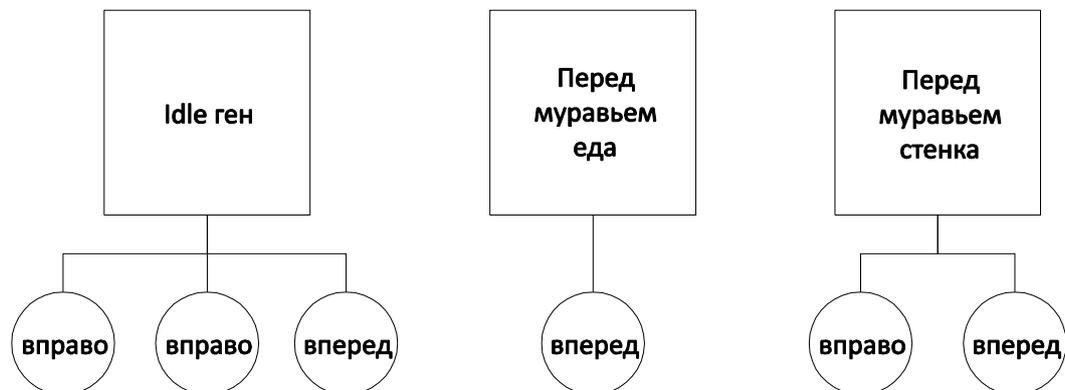


Figure 3: Пример схемы генов для задачи муравья.

Каждый индивид имеет специальный ген, который выполняется при каждом шаге симуляции (idle ген). Для простых задач модель индивида может состоять только из одного idle гена.

Каждый генетический оператор гарантирует семантическую корректность полученной программы (соответствие типов операндов). Схема работы операндов приведена на рисунке 4.

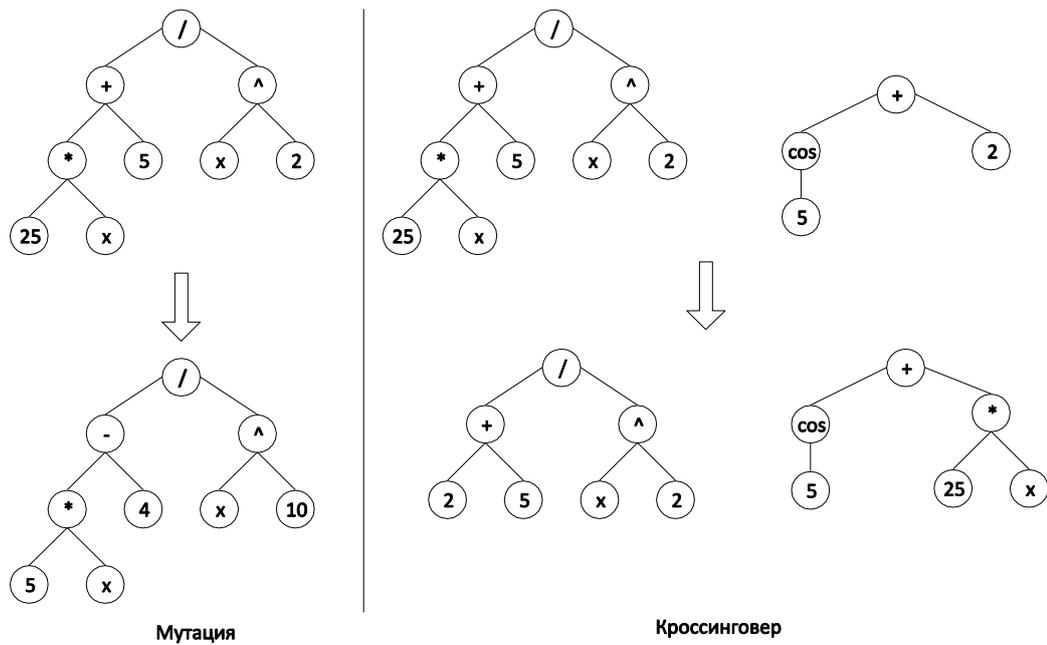


Figure 4: Пример преобразования деревьев с помощью генетических операторов.

5 Техническое описание программного продукта

Программный продукт написан на языке D. Выбранный язык обеспечил быстродействие и скорость разработки продукта. Целевые платформы: Windows, GNU/Linux. Система разработана на основе модульной архитектуры с ядром. Большинство модулей системы могут быть изменены для конкретных нужд разработчика, что позволяет расширить круг задач, решаемые системой. Схема модульной архитектуры представлена на рисунке 5.

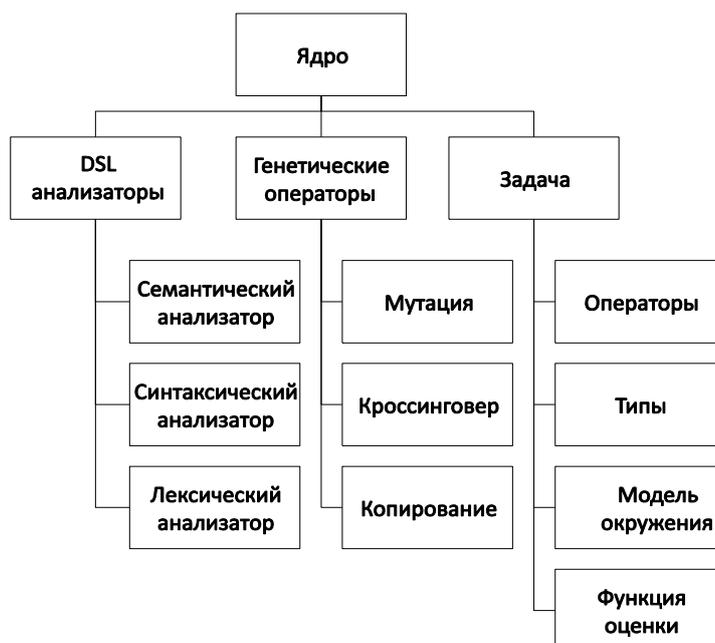


Figure 5: Схема модулей программного продукта.

Для записи, хранения, обработки полученных решений был создан лексический, синтаксический и семантические анализаторы сгенерированных DSL языков. Итоговый код индивидов транслируется в язык D для последующей компиляции и выполнения. Схема работы анализаторов представлены на рисунке 6.

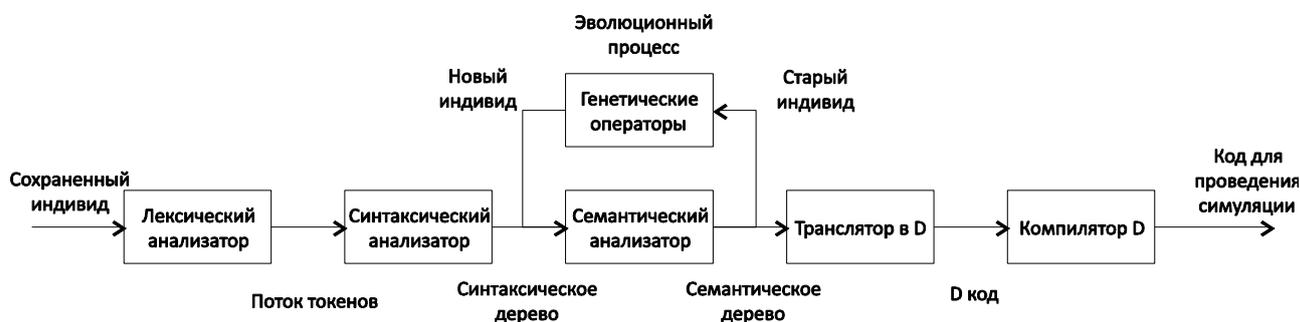


Figure 6: Схема работы анализаторов.

6 Заключение

Разработан инструмент для автоматизации решения задач с помощью генетических алгоритмов. Пользователь может настраивать программный продукт для расширения класса задач, решаемых системой. В будущем планируется улучшить набор возможностей по следующим пунктам:

1. Расширение возможностей внутренних DSL языков с помощью генов, генерируемых автоматически.
2. Создание инструментов автоматического объединения ЭВМ с установленным программным продуктом в единую вычислительную сеть.
3. Создание расширенной стандартной библиотеки операторов и типов.
4. Возможность модернизации системы с помощью полученных решений задач.

Список литературы

- [1] Koza, John (1992), Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press. ISBN 0-262-11170-5
- [2] Holland, John H (1975), Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor