

СТРУКТУРА ГЕНЕТИЧЕСКОГО АЛГОРИТМА ОПТИМИЗАЦИИ СОСТАВА ПРОФЕССИОНАЛЬНЫХ ГРУПП ПРОГРАММИСТОВ

Ждановский А.М., Прихожий А.А.

*Белорусский национальный технический университет, Минск, Беларусь
arseni_zhdanouski@epam.com, prihozhy@yahoo.com*

Генетические алгоритмы [1-7] – это мета-эвристика, используемая для решения слабо структурированных комбинаторных задач моделирования и оптимизации в различных прикладных областях. Она основана на высокоуровневом представлении эволюционного процесса поиска решения. Генетический алгоритм оперирует популяцией искусственных хромосом. Хромосома представляет решение поставленной задачи и характеризуется фитнес функцией, показывающей насколько хорошим является решение. Начальное состояние популяции формируется путем случайной генерации хромосом. Генетический алгоритм выполняет отбор, основываясь на значении фитнес функции, производит скрещивание и мутацию выбранных хромосом, формирует поколение наследников и обновляет популяцию. Выбрав родительские хромосомы, он выполняет скрещивание путем рекомбинации отрезков хромосом, или выполняет мутацию путем модификации отдельных генов. Полученные потомки он включает в состав следующего поколения. Этот процесс повторяется итерационно, доля успешных решений растет, увеличивается среднее значение фитнес функции по хромосомам популяции. Процесс продолжается, пока не будет получено приемлемое решение, достигнуто состояние стагнации или выполнено требуемое число итераций алгоритма.

Генетический алгоритм состоит из отдельных блоков, это его сильная сторона при адаптации к конкретным решаемым задачам. Каждый из блоков является достаточно универсальным, и может быть адаптирован с небольшими модификациями к различным применениям алгоритма. Основные блоки алгоритма ответственны за кодирование хромосом, вычисление фитнес функции, выполнение операций отбора, скрещивания и мутации.

Проблема комплектации программистских групп для реализации проектов в области ИТ рассматривалась в известной литературе. Технология Agile [8] рассчитана на распределенную работу групп программистов с использованием адаптивного планирования, эволюционного развития, постоянного совершенствования, быстрого и гибкого реагирования на изменения, активного взаимодействия групп разработчиков и заказчиков. В основе технологии лежит множество рекомендаций, носящих качественный характер. Для поддержки работы коллективов программистов разработана агентная распределенная архитектура [9]. Анализ публикаций показывает, что в известной литературе не предлагается количественных методов оптимизации размера и состава команд программистов.

В работах [10, 11] предложен формальный метод распределения множества программистов по группам. Этот метод заранее не фиксирует число групп, число программистов в каждой группе, включение программиста в конкретную группу. При этом, он учитывает требования к проекту, над которым будут работать программистские группы. Метод допускает использование при оптимизации различных критерии качества распределения программистов по группам. Параметры групп и распределения программистов по группам являются объектом оптимизации с целью улучшения характеристик и возможностей каждой группы и всех групп в целом при работе над конкретным проектом. В работах [10, 11] акцент сделан на численном моделировании характеристик каждого программиста, используемых при работе над проектом языков, технологий и инструментов программирования, на моделировании характеристик каждой группы и характеристик всех групп, работающих над проектом. В них не рассмотрена структура разрабатываемых алгоритмов оптимизации.

Данная статья описывает и анализирует различные аспекты построения и функционирования генетического алгоритма оптимизации распределения программистов по группам с учетом требований к квалификации, предъявляемых проектом. Основной цикл генетического алгоритма представлен на рисунке 1. Исходная популяция размером 500 хромосом генерируется таким образом, что в каждой хромосоме все программисты случайным образом с равномерным распределением вероятности распределяются по группам в количестве от 4 до 8. Оптимальное количество групп заранее неизвестно, поэтому алгоритм может как увеличивать, так и сокращать число групп. С учетом оцениваемых характеристик каждой группы и требований проекта, группы делятся на работоспособные и неработоспособные резервные. В процессе своей работы, генетический алгоритм стремится сократить число и размер неработоспособных групп и увеличить количество программистов в работоспособных группах. Он также стремится сбалансировать квалификацию групп по каждой технологии и средствам программирования.

Основной цикл алгоритма генерирует поколения хромосом посредством операций селекции, скрещивания, мутации и обновления популяции. Селекция родительских хромосом выполняется посредством правила рулетки. Селекция хромосом, включаемых в обновленную популяцию, выполняется с учетом функции полезности, но с сохранением генетического многообразия с целью избегания явления стагнации в эволюционном процессе.

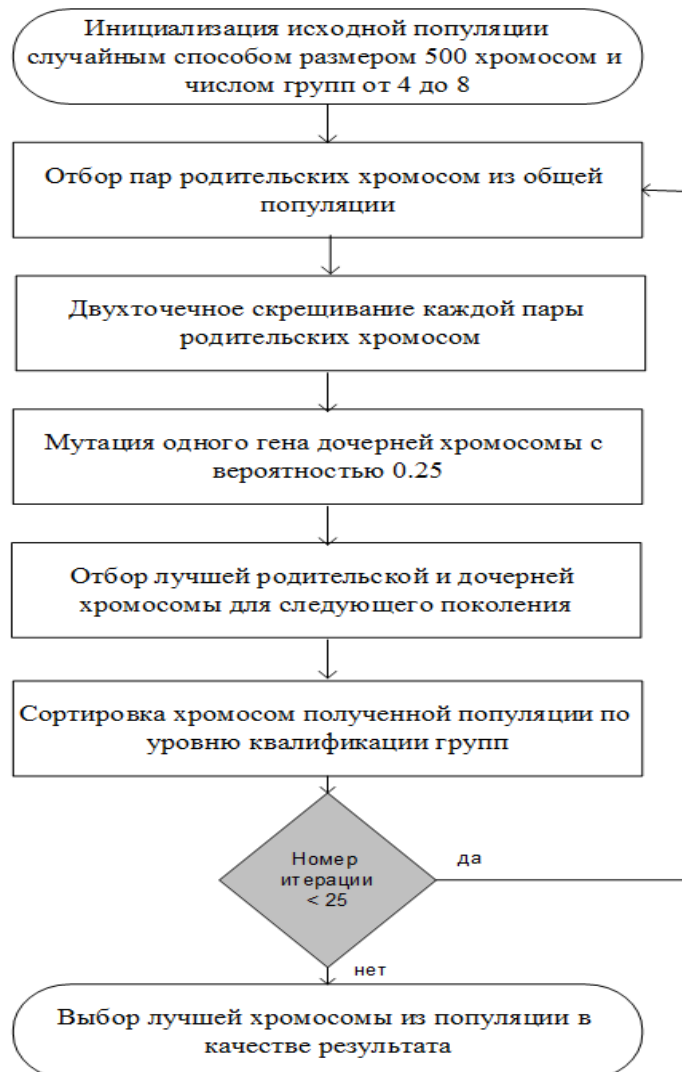


Рисунок 1 – Генетический алгоритм оптимизации распределения программистов по группам

В текущей реализации генетического алгоритма, операция скрещивания хромосом реализуется двухточечным кроссовером, который, как показывают эксперименты, обеспечивает достаточно эффективную рекомбинацию участков хромосом и позволяет сгенерировать большое многообразие в количестве и составе групп программистов. Генетический алгоритм построен таким образом, что двухточечный кроссовер легко заменяется многоточечным, в частности, однородным кроссовером,

Операция мутации изменяет значения генов в хромосоме и позволяет переводить программиста (или нескольких программистов одновременно) из одной группы в другие. Замечательным свойством реализации мутации в нашем алгоритме является обеспечение создания новых групп и, таким образом, обеспечение увеличения общего числа групп программистов. Следует заметить, что операция скрещивания не обладает подобным свойством, она может только сократить число групп.

Генетический алгоритм формирует новое поколение, включая в него лучшие родительские и дочерние хромосомы после выполнения каждой генетической операции. Хромосома с наибольшим значением функции полезности представляет результат работы генетического алгоритма.

Для вычисления функции полезности хромосом (вариантов распределения программистов по группам), мы использовали метод, предложенный в работах [10, 11]. Этот метод учитывает индивидуальную квалификацию каждого программиста, актуальность и популярность технологий и инструментов программирования, обеспечивает интегрированную оценку квалификации каждой группы с учетом требований, предъявляемых проектом. Квалификация группы складывается из средней квалификации по всем ее программистам и используемым технологиям, квалификации лучших представителей группы по каждой из технологий, пороговых значений уровня квалификации программистов, лучших представителей и всей группы в целом. Требования к выполняемому программистами проекту учитывают направленность проекта, используемые языки, инструменты и технологии программирования, опыт разработчиков.

Экспериментальное исследование генетического алгоритма проводилось как на реальных выборках программистов, так и на случайно сгенерированных выборках. В частности, экспериментальные результаты получены на реальной выборке из 24 программистов, получивших высшее образование в университетах Республики Беларусь (БНТУ, БГУ, БГУИР и др.). Квалификация программистов оценивалась по 16 технологиям программирования. Качество распределения программистов по группам оценивалось средней взвешенной квалификацией одной группы и количеством сформированных групп. Произведение этих двух параметров дает суммарную квалификацию всех групп. Основным требованием к проекту явилась относительная пороговая средняя взвешенная квалификация одной группы. При значениях этой квалификации в диапазоне от 0.40 до 0.75 с шагом 0.05 число сформированных групп уменьшалось согласно последовательности 9, 8, 8, 8, 6, 5, 3, 2, а средняя квалификация одной группы изменялась согласно последовательности 0.602, 0.631, 0.630, 0.626, 0.683, 0.696, 0.757 и 0.785. Суммарная квалификация всех групп сократилась с 5.42 до 1.57. При этом число не включенных в группы программистов (резерв) увеличилось с 1 до 18, что обусловлено повышением требований к квалификации, приводящим к невозможности сформировать достаточно большое число групп.

Генетический алгоритм оптимизации распределения программистов по группам достаточно легко поддается распараллеливанию методами, предложенными в работе [12].

Выводы. Проблема оптимизации распределения программистов по группам является слабо структурированной, многокритериальной, комбинаторной. Численные методы решения этой проблемы не освещены в известной литературе. В статье исследуется структура генетического алгоритма в целом и генетических операций в частности для решения проблемы. Структура обеспечивает учет особенностей решаемой задачи, разнообразия исходной информации, сложности построения интегрированной метрики для оценки квалификации группы, высокой

неопределенности, проявляющейся в отсутствии априорной информации об оптимальном числе групп и предпочтительном включении программиста в ту или иную группу. Предложена программная реализация генетического алгоритма, позволившая провести эксперименты, подтвердившие высокую практическую значимость разработки.

Список литературы:

1. Barricelli, N.A. Symbio genetic evolution processes realized by artificial methods / N.A. Barricelli // *Methodos*, 1957, pp. 143–182.
2. McCall, J. Genetic algorithms for modelling and optimization / J. McCall // *Journal of Computational and Applied Mathematics*, Vol. 184, 2005, pp. 205–222.
3. Lamini, C. Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning / C. Lamini, S. Benhlina, A. Elbekri // *Procedia Computer Science*, Vol. 127, 2018, pp. 180-189.
4. Thomas, D. A Genetic Algorithm Approach to Autonomous Smart Vehicle Parking system / D. Thomas, B.C. Kooor // *Procedia Computer Science*, Vol. 125, 2018, pp. 68–76.
5. Assi, M. Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem / Assi, M., Halawi, B., Haraty, R.A. // *Procedia Computer Science*, Vol. 126, 2018, pp. 899–906.
6. Sergeeva, M., Dynamic airspace configuration by genetic algorithm / M. Sergeeva, D. Delahaye, C. Mancel, A. Vidosavljevic // *journal of traffic and transportation engineering* 2017; 4 (3): pp. 300-314.
7. Прихожий, А. Эвристический генетический алгоритм оптимизации вычислительных конвейеров / А.А. Прихожий, А.М. Ждановский, О.Н. Карасик, М. Маттавелли // *Доклады БГУИР*, 2017, № 1, с. 34-41.
8. Joshi, S. Agile Development - Working with Agile in a Distributed Team Environment / S. Joshi // *MSDN Magazine*, 2012, Vol.27, No.1, pp.1-6.
9. Müller, J.P., Rao, A.S., Singh, M.P. A-Teams: An Agent Architecture for Optimization and Decision-Support, *Proceedings 5th International Workshop, ATAL'98 Paris, France, July 4–7, 1998*, pp. 261-276.
10. Прихожий, А.А. Метод оценки квалификации и оптимизация состава профессиональных групп программистов / А.А. Прихожий, А.М. Ждановский // *Системный анализ и прикладная информатика*. – № 2. – 2018. – С. 4-12.
11. Prihozhy, A. Genetic algorithm of optimizing the size, staff and number of professional teams of programmers / A. Prihozhy, A. Zhdanouski // *Open Semantic Technologies for Intelligent Systems: Research Paper Collection, Issue 3*. – Minsk, BSUIR, 2019. – P. 305–310.
12. Prihozhy, A.A. Analysis, transformation and optimization for high performance parallel