

СИМБИОЗ КВАНТОВЫХ ВЫЧИСЛЕНИЙ И ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

И. В. Кухарчук

Кафедра электронных вычислительных машин, Белорусский государственный университет информатики и радиоэлектроники

Минск, Республика Беларусь

E-mail: ihar.kukharchuk@gmail.com

Данная статья посвящена анализу объективности использования генетических алгоритмов в квантовых вычислениях. Рассматриваемые вычисления, ровно как и алгоритмы, основаны на вероятностной модели, что позволяет добиться более высокой производительности вычислений. К тому же структура построения программ, придерживающихся данной концепции, обладает широкими возможностями для параллелизма, позволяя наиболее эффективно использовать ресурсы вычислительных систем.

ВВЕДЕНИЕ

При решении оптимизационных и комбинаторно-логических задач проектирования, конструирования и искусственного интеллекта на основе графовых моделей часто включают в общую архитектуру один или несколько механизмов эволюционного моделирования со стратегиями и методами бионического поиска [1]. Генетические алгоритмы используют накопленную в процессе эволюции информацию для получения оптимальных решений. Квантовые же подпрограммы чаще всего используются с точки зрения параллельных вычислений. Согласно принципу суперпозиции вся графовая модель или система способна как бы одновременно существовать во многих возможных состояниях. В данной статье предлагается технология, построенная на симбиозе квантового и бионического поиска. Предложены принцип и алгоритм такого поиска. Данное описание конечного приложения даёт возможность увеличить количество рассмотренных вариантов с отсутствием или незначительными потерями во времени работы, а также предотвратить скорую сходимость алгоритмов.

I. ПОСТАНОВКА ЗАДАЧИ

Для реализации симбиоза была выбрана классическая задача коммивояжёра, которая формулируется следующим образом: дан полный взвешенный граф $G(X, V)$ порядка n , где $X = x_1, x_2, \dots, x_n$ – множество вершин; $V \leq X * X$ – множество рёбер. В данном графе нужно найти Гамильтонов цикл, имеющий наименьший суммарный вес входящих в него рёбер:

$$\begin{cases} Q(x) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \\ \sum_{i=1}^N x_{ij} = 1, \forall j = \overline{1, N} \\ \sum_{j=1}^N x_{ij} = 1, \forall i = \overline{1, N} \end{cases}$$

где c_{ij} – вес ребра (i, j)

$$x_{ij} = \begin{cases} 1, & \text{если есть переход из } i \text{ в } j \\ 0, & \text{если перехода из } i \text{ в } j \text{ нет} \end{cases}$$

Необходимым решением данной задачи будет любой маршрут между вершинами графа («городами»), который должен пересекать все вершины без исключения и ни одну не пересекать больше одного раза.

II. КОНЦЕПЦИЯ

Как отмечено на рисунке 1, вычислительным ядром предстаёт квантовая подпрограмма, генетический алгоритм в свою очередь подготавливает данные для обработки и анализирует результаты [2].



Рис. 1 – Общий вид алгоритма

Входными данными является граф. Генетический препроцессор готовит популяцию, которая будет отправлена на процедуру скрещивания. Квантовый препроцессор формирует оракул. Квантовый алгоритм производит последовательные скрещивания над входной популяцией. Генетический постпроцессор производит селекцию и выборку искомого результата. Если конечный результат не достигнут, постпроцессор формирует новую популяцию генотипов. Выходными данными при решении данной задачи является наиболее оптимальный маршрут. Рассмотрим более подробно каждый из компонентов предложенной архитектуры.

III. ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Примем к решению задачу о нахождении маршрута между четырьмя городами. Необходимо закодировать маршрут. Кодом являются номера городов, расположенных последовательно, начиная с самого первого и заканчивая предпоследним, так как маршрут замкнут. Одна из возможных последовательностей отображена на рисунке 2.

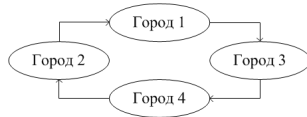


Рис. 2 – Возможное расположение городов

В данном случае города получают номера от 1 до 4 и формируется ряд вида: $\boxed{1} \boxed{3} \boxed{4} \boxed{2}$. Теперь необходимо представить решение в виде хромосомы. Хромосома кодируется в виде битового вектора [3]. Длина гена в битах будет равна: $L = \log_2 N$. Для нашего примера $L = \log_2 4 = 2$, то есть для кодирования одного гена понадобится 2 бита. Кодировем всю возможную последовательность:

Таблица 1 – Кодирование городов

00	01	10	11
1	2	3	4

Для данной задачи достаточной является популяция, равная 16 особям. Селекция производится посредством оценки результата функции приспособленности (оценка длины и стоимости маршрута).

IV. КВАНТОВЫЙ АЛГОРИТМ

Квантовый алгоритм участвует лишь в процессе скрещивания, предотвращая сходимость особей и повышая их разнообразие. Построение оракула сводится к совокупности необходимого числа матриц отрицания вдоль главной диагонали формирующей матрицы. Квантовый оракул для нахождения оптимального маршрута в данном случае состоит из двух последовательных гейтов, представленных в виде матриц [4]:

$$U_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad U_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Согласно алгоритму необходимо несколько раз применить квантовую подпрограмму для обеспечения разнообразия особей. Схема квантовой подпрограммы выглядит следующим образом:

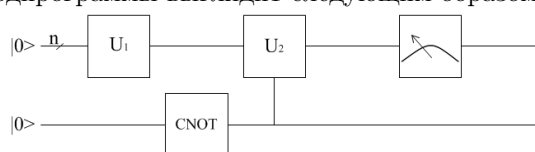


Рис. 3 – Схема квантовой подпрограммы

V. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Были проведены эксперименты по количеству запусков квантовой подпрограммы и генетического алгоритма в качестве «обёртки» над ним. Важно провести «обучение» схемы с целью получения коррелирующих значений по необходимому количеству запусков с идеей добиться удовлетворяющей вероятности правильного решения задачи, так как обе применённые концепции относятся к вероятностным.

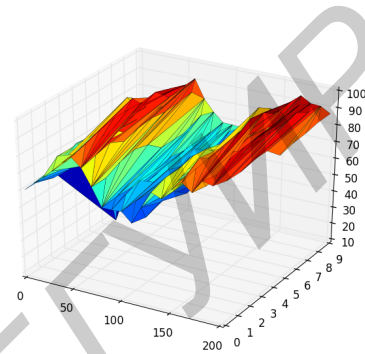


Рис. 4 – Результаты запусков алгоритма

На рисунке 4 вы можете увидеть количество запусков алгоритма [1;200], количество запусков квантовой подпрограммы [1;10] и по шкале Z вероятность получения верного ответа. Результат таков, что для получения вероятности порядка 95% достаточно генерировать порядка 20 поколений с шестью запусками квантовой подпрограммы. Алгоритм может быть улучшен посредством отброса сгенерированных хромосом, которые содержат по два и более одинаковых городов в результате скрещивания. На текущий момент данным хромосомам присваивается наименьшее значение функции приспособленности, что влечёт за собой дополнительные издержки. Однако алгоритм уже на текущий момент может составить конкуренцию имеющимся аналогам. Задача легко масштабируется с учётом наличия необходимого числа кубитов.

1. Горемыкин, А. О. Параллельный квантовый генетический алгоритм для решения оптимизационных задач на эксафлопсных суперкомпьютерах / А. О. Горемыкин, Н. Н. Попова // Программные системы и инструменты –2014№ 15. –С. 173–183.
2. Курейчик, В. М. Квантовые и генетические алгоритмы – новая технология эволюционного поиска / В. М. Курейчик // ВЮНЦ –2015№ 2. –С. 41–50.
3. Algebraic and Number Theoretic Algorithms: Verifying Matrix Products [Electronic resource] / Ed. S. Jordan. 2015. – Mode of access: <http://math.nist.gov/quantum/zoo/>. – Date of access: 15.09.2015.
4. Душкин, Р. В. Квантовые вычисления и функциональное программирование / Р. В. Душкин –Москва: Литкон, 2014. – 341 с.