

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего  
профессионального образования «Южно-Уральский государственный университет (национальный  
исследовательский университет)» Высшая школа электроники и компьютерных наук  
Кафедра системного программирования

## **Применение метода проектирования Q-эффективных программ к методу стохастического градиентного спуска для обучения нейронных сетей**

Рецензент:  
доцент кафедры ТУиО  
ФГБОУ ВО «ЧелГУ»,  
к.ф.-м.н. С.А. Никитина

Научный руководитель:  
доцент кафедры СП,  
к.ф.-м.н. В.Н. Алеева

Автор работы:  
студент группы КЭ-229  
А.С. Сапожников

# ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

## Цель

Применение метода проектирования Q-эффективных программ к методу стохастического градиентного спуска для обучения нейронных сетей.

## Задачи

1. Изучить параллельные компьютерные архитектуры и реализацию ресурса параллелизма на основе концепции Q-детерминанта.
2. Изучить метод стохастического градиентного спуска, применяемый для оптимизации целевой функции нейронной сети.
3. Изучить метод обратного распространения ошибки, применяемого для вычисления градиентов нейронной сети.
4. Изучить архитектуру суперкомпьютера «Торнадо ЮУрГУ».

# ЦЕЛЬ И ЗАДАЧИ ИССЛЕДОВАНИЯ

5. Разработать Q-эффективную реализацию алгоритма, выполняющего метод стохастического градиентного спуска для обучения нейронных сетей.
6. Разработать Q-эффективную реализацию алгоритма, выполняющего метод обратного распространения ошибки, применяемого для вычисления градиентов нейронной сети.
7. Разработать Q-эффективную программу для системы с общей памятью.
8. Разработать Q-эффективную программу для системы с распределенной памятью.
9. Протестировать разработанные Q-эффективные программы.
10. Провести экспериментальное исследование разработанных Q-эффективных программ.

# КОНЦЕПЦИЯ Q-ДЕТЕРМИНАНТА

Пусть  $\alpha$  – численный алгоритм для решения алгоритмической проблемы  $\vec{y} = F(N, B)$  и  $M = \{1, \dots, m\}$ . Предположим, что алгоритм  $\alpha$  состоит в том, чтобы для каждого  $i \in M$  нужно найти  $y_i$ , как значение Q-терма  $f_i$ .

- $N$  – параметр размерности алгоритмической проблемы.
- $B$  – множество входных данных.
- $\vec{y}$  – выходные данные.
- $Q$  – множество операций используемых алгоритмом  $\alpha$ .
- Q-терм – это понятие концепции Q-детерминанта.

Тогда множество Q-термов  $\{f_i | i \in M\}$  называется Q-детерминантом алгоритма  $\alpha$ . Также система уравнений  $y_i = f_i$  для всех  $i \in M$  называется представлением алгоритма  $\alpha$  в форме Q-детерминанта.

# МЕТОД ПРОЕКТИРОВАНИЯ Q-ЭФФЕКТИВНЫХ ПРОГРАММ

Метод использует расширенную модель концепции Q-детерминанта и состоит из этапов:

1. Построение Q-детерминанта алгоритма.
2. Описание Q-эффективной реализации алгоритма.
3. Разработка параллельной программы для выполнимой Q-эффективной реализации алгоритма.

# МЕТОД СТОХАСТИЧЕСКОГО ГРАДИЕНТНОГО СПУСКА

Метод стохастического градиентного спуска (СГС) заключается в вычислении новых весов  $W^{*(l)}$  и смещений  $B^{*(l)}$  по формуле:

$$W^{*(l)} = W^{(l)} - \varepsilon \cdot \nabla_W C^{(l)},$$
$$B^{*(l)} = B^{(l)} - \varepsilon \cdot \nabla_B C^{(l)},$$

где  $B^{(l)} = \{b_1, \dots, b_r\}$  – вектор смещений слоя  $l$ ;

$$W^{(l)} = \begin{pmatrix} w_{11} & \cdots & w_{1r} \\ \vdots & \ddots & \vdots \\ w_{h1} & \cdots & w_{hr} \end{pmatrix} \text{ – матрица весов слоя } l;$$

$\varepsilon$  – скорость обучения.

# МЕТОД ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ

Метод обратного распространения ошибки заключается в вычислении меры влияния нейронов на величину ошибки следующим образом:

$$\delta^L = \frac{\partial C}{\partial A^L} \cdot \sigma'(z^L), \quad \delta^{(l)} = \sigma'(z^{(l)}) \cdot (\delta^{(l+1)} \cdot W^{(l+1)T})$$

где  $Z^{(l)} = \{z_1, \dots, z_r\}$  – активационный потенциал слоя  $l$ .

По результатам вычислений рассчитываются градиенты весов и смещений

$$\nabla_W C^{(l)} = \delta^{(l)} \cdot A^{(l-1)}, \quad \nabla_B C^{(l)} = \delta^{(l)},$$

где  $A^{l-1} = \{a_1, \dots, a_h\}$  – вектор выходных сигналов слоя  $l - 1$ .

# Q-ДЕТЕРМИНАНТЫ

Q-детерминант метода СГС состоит из двух множеств безусловных Q-терм:

$$w_{ij}^{(l)*} = w_{ij}^{(l)} - \varepsilon \cdot \delta_i^{(l)} \cdot a_j^{(l-1)},$$
$$b_i^{(l)*} = b_i^{(l)} - \varepsilon \cdot \delta_i^{(l)},$$

где  $i \in H$  – количество нейронов на слое  $l - 1$  ;  
 $j \in R$  – количество нейронов на слое  $l$ .

Q-детерминант метода обратного распространения ошибки состоит из одного множества безусловных Q-терм:

$$\delta_i^{(l)} = \sigma'(z_i^{(l)}) \cdot (\delta_j^{(l+1)} \cdot w_{ji}^{(l+1)}),$$

где  $j \in K$  – количество нейронов на слое  $l + 1$ .



# ОПИСАНИЕ Q-ЭФФЕКТИВНОЙ РЕАЛИЗАЦИИ

Для систем с общей памятью все Q-термы  $\{b_0, \dots, b_r\}$ ,  $\{\{w_0, \dots, w_r\}_0, \dots, \{w_0, \dots, w_r\}_h\}$  и  $\{\delta_0, \dots, \delta_r\}$  должны вычисляться одновременно, при этом операции должны выполняться по мере их готовности к выполнению.

Для систем с распределенной памятью Q-термы должны быть распределены так, чтобы каждому узлу  $S$  досталось несколько множеств  $\{b_i, \dots, b_{i+t}\}$ ,  $\{\{w_0, \dots, w_r\}_i, \dots, \{w_0, \dots, w_r\}_{i+t}\}$  а множество Q-терм  $\{\delta_0, \dots, \delta_r\}$  было передано всем узлам.

# СРЕДСТВА РАЗРАБОТКИ

- Для разработки Q-эффективных программ был использован язык программирования C++
- Для Q-эффективной программы для общей памяти применялась технология OpenMP
- Для Q-эффективных программ для распределенной памяти применялись технологии OpenMP и MPI
- Все вычисления проводились на суперкомпьютере «Торнадо ЮУрГУ»

# ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

- При проведении экспериментов оценивалось время выполнения Q-эффективных программ при изменении размера архитектуры нейронной сети.
- Для систем с общей памятью проводились эксперименты с количеством ядер от 2 до 12, а для систем с распределенной памятью – с количеством узлов от 2 до 12.
- Ускорение вычислялось по формуле  $S_p = \frac{T_{\Pi}}{T_p}$ .
- Эффективность рассчитывалась по формуле  $E_p = \frac{S_p}{p}$ .

# УСКОРЕНИЕ

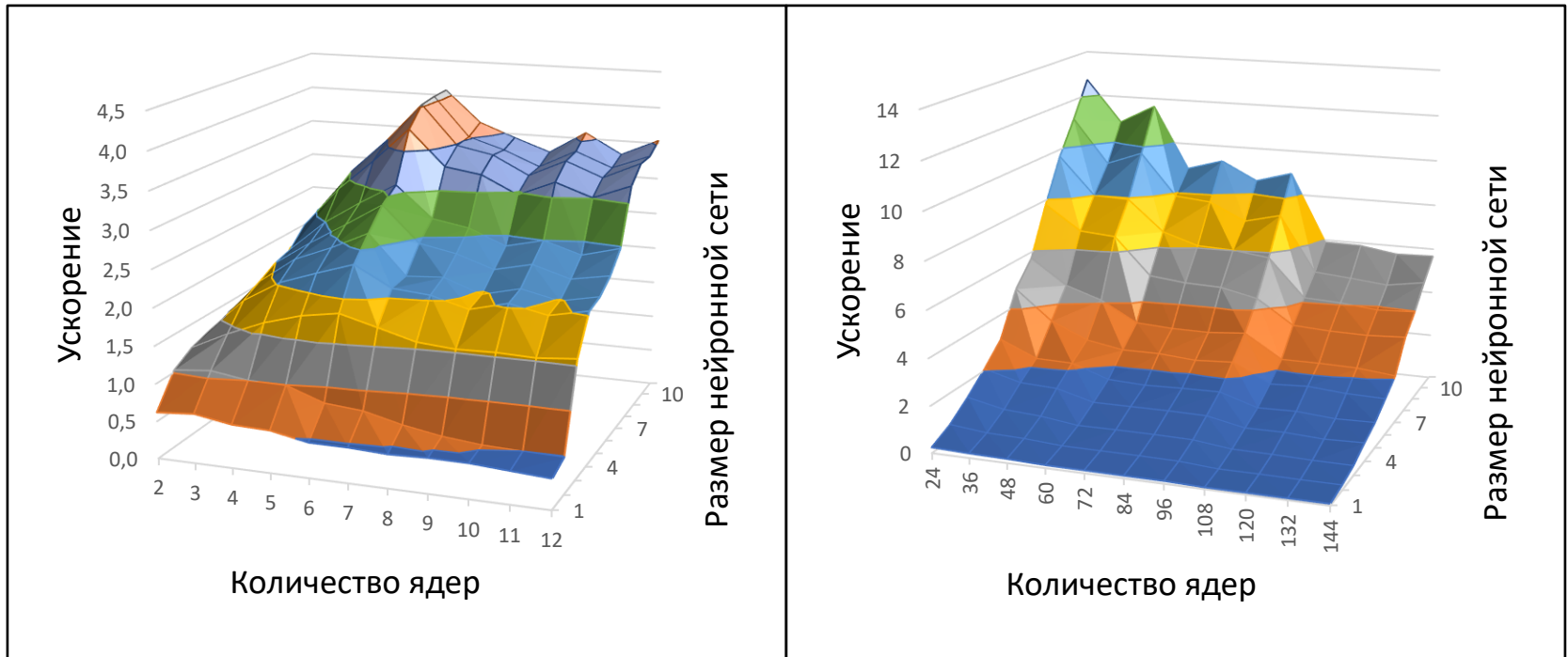


Рисунок 1 – Ускорение Q-эффективной программы для метода обратного распространения ошибки для системы с общей памятью

Рисунок 2 – Ускорение Q-эффективной программы для метода обратного распространения ошибки для системы с распределенной памятью

# УСКОРЕНИЕ

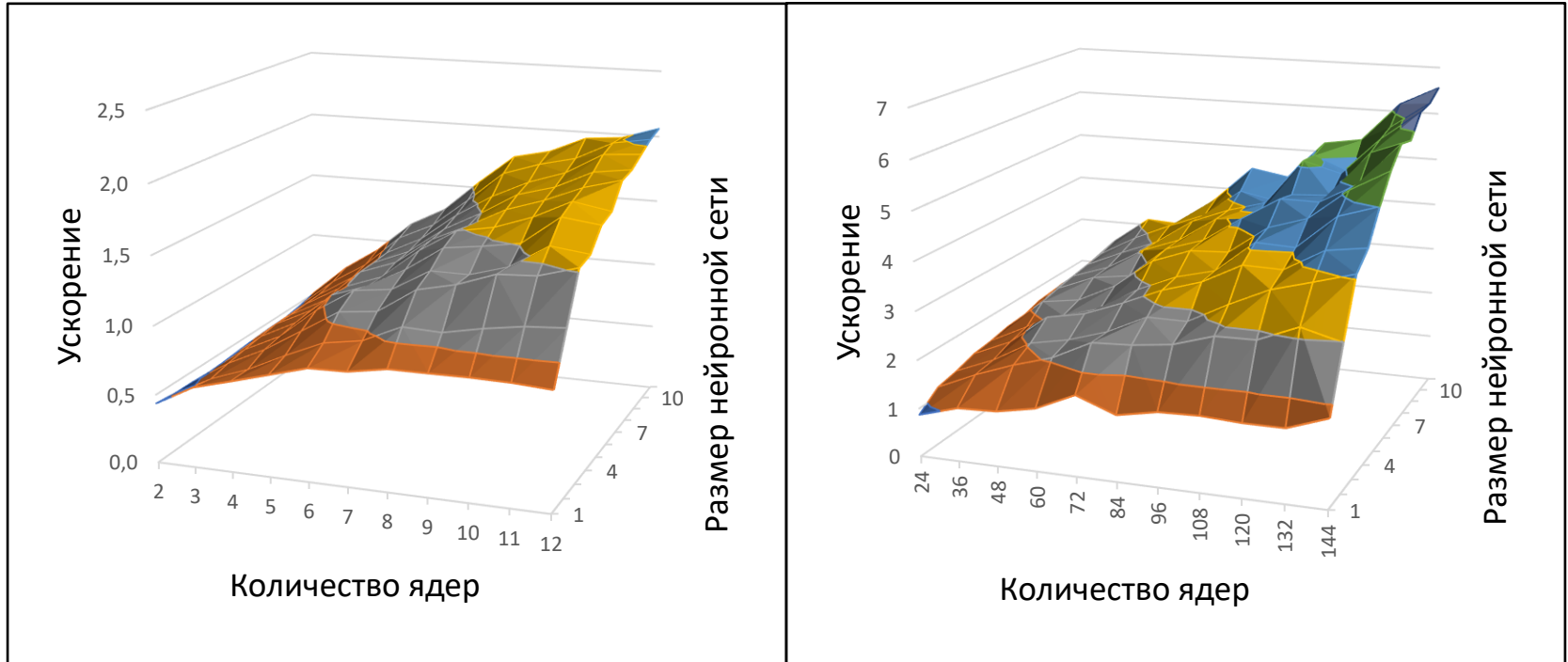


Рисунок 3 – Ускорение Q-эффективной программы для метода СГС для системы с общей памятью

Рисунок 4 – Ускорение Q-эффективной программы для метода СГС для системы с распределенной памятью

# ЭФФЕКТИВНОСТЬ

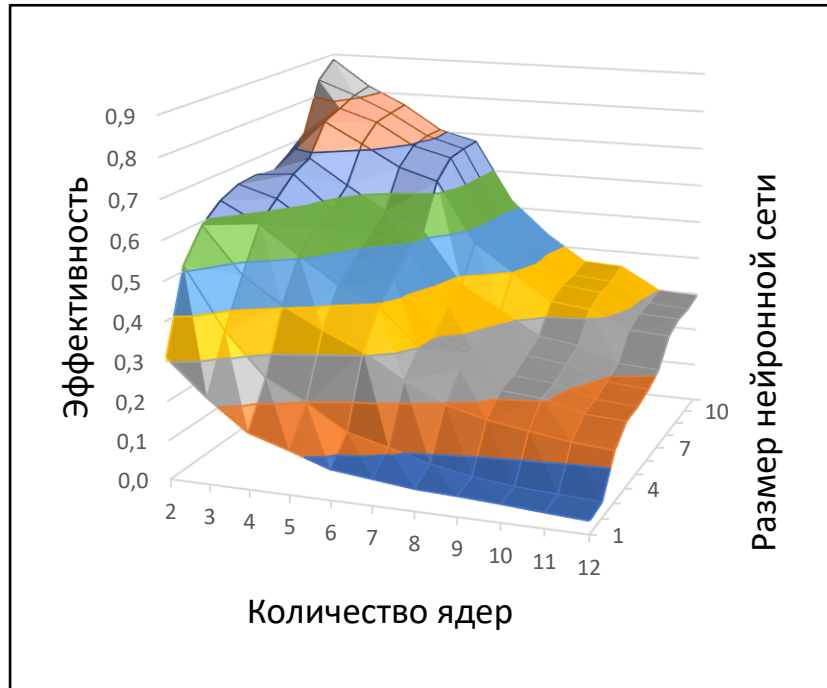


Рисунок 5 – Эффективность Q-эффективной программы для метода обратного распространения ошибки для системы с общей памятью

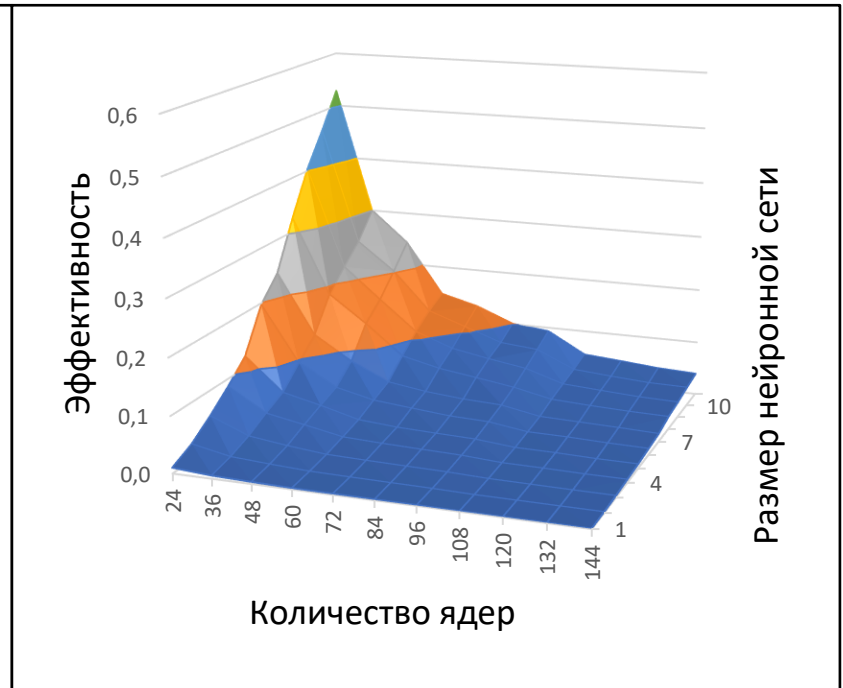


Рисунок 6 – Эффективность Q-эффективной программы для метода обратного распространения ошибки для системы с распределенной памятью

# ЭФФЕКТИВНОСТЬ

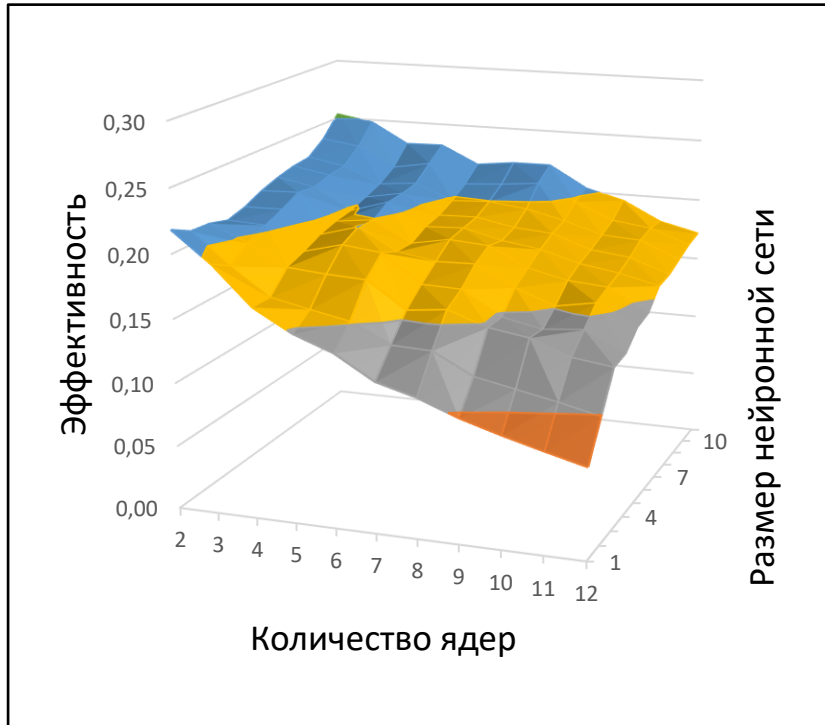


Рисунок 7 – Эффективность Q-эффективной программы для метода СГС для системы с общей памятью

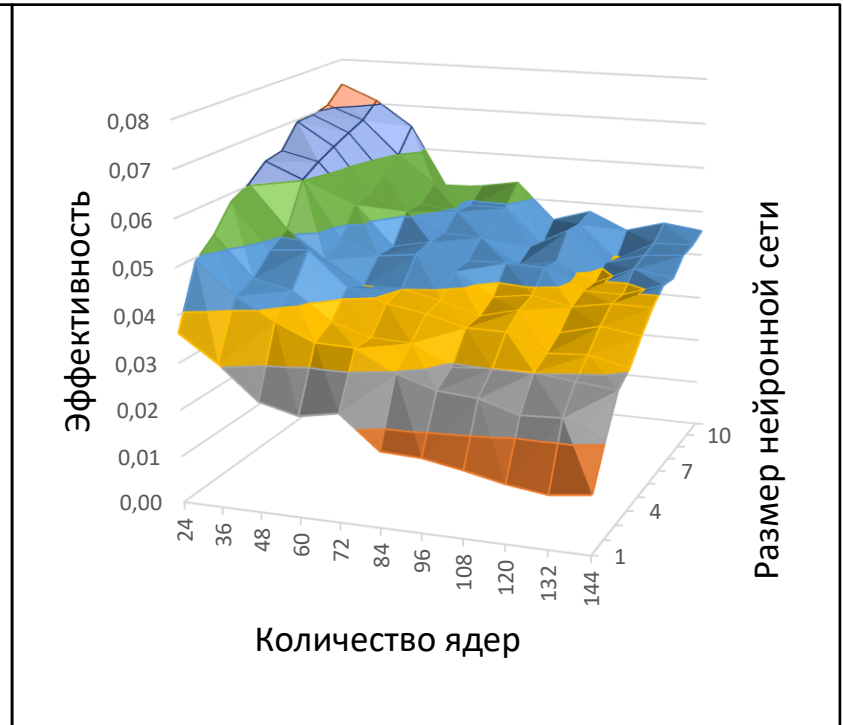


Рисунок 8 – Эффективность Q-эффективной программы для метода СГС для системы с распределенной памятью

# ОСНОВНЫЕ РЕЗУЛЬТАТЫ

1. Изучены параллельные компьютерные архитектуры и реализация ресурса параллелизма на основе концепции Q-детерминанта.
2. Изучен метод стохастического градиентного спуска, применяемый для оптимизации целевой функции нейронной сети.
3. Изучен метод обратного распространения ошибки, применяемого для вычисления градиентов нейронной сети.
4. Изучена архитектура суперкомпьютера «Торнадо ЮУрГУ».
5. Разработана Q-эффективная реализация алгоритма, выполняющего метод стохастического градиентного спуска для обучения нейронных сетей.
6. Разработана Q-эффективная реализация алгоритма, выполняющего метод обратного распространения ошибки, применяемого для вычисления градиентов нейронной сети.
7. Разработана Q-эффективная программа для системы с общей памятью.
8. Разработана Q-эффективная программа для системы с распределенной памятью.
9. Проведено тестирование разработанных Q-эффективных программ.
10. Проведено экспериментальное исследование разработанных Q-эффективных программ.