

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра ВТ**

**КУРСОВАЯ РАБОТА**  
**по дисциплине «Архитектура Параллельных Вычислительных**  
**Систем»**  
**Тема: Решение задачи о восьми ферзях с использованием**  
**мультиагентной системы**

Студенты гр. 6307	_____	Киварин Д.М. Новиков Б.М. Ходос А.А.
Преподаватель	_____	Костичев С.В.

Санкт-Петербург  
2020

## **СОДЕРЖАНИЕ**

<b>ВВЕДЕНИЕ</b>	<b>4</b>
<b>1. МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ О ВОСЬМИ ФЕРЗЯХ</b>	<b>5</b>
1.1. Метод перебора	5
1.2. Метод с использованием мультиагентной системы	6
<b>2. МУЛЬТИАГЕНТНАЯ СИСТЕМА РЕШЕНИЯ ЗАДАЧИ О ВОСЬМИ ФЕРЗЯХ</b>	<b>8</b>
2.1. Сервер	10
2.2. Агент-фигура	11
2.3. Агент-судья	12
2.4. Монитор	14
<b>3. ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ</b>	<b>15</b>
<b>ВЫВОД</b>	<b>17</b>
<b>СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ</b>	<b>18</b>

## **ВВЕДЕНИЕ**

Задача о восьми ферзях — широко известная задача по расстановке фигур на шахматной доске. Исходная формулировка: «Расставить на стандартной 64-клеточной шахматной доске 8 ферзей так, чтобы ни один из них не находился под боем другого». Подразумевается, что ферзь бьёт все клетки, расположенные по вертикалям, горизонталям и обеим диагоналям. Обобщение задачи — расставить максимальное количество взаимно не бьющих друг друга ферзей на прямоугольном поле, в частности, квадратном поле, со стороной  $n$ .

Целью работы является сравнить классический метод перебора для поиска решения данной задачи и метод с использованием мультиагентной системы.

## 1. МЕТОДЫ РЕШЕНИЯ ЗАДАЧИ О ВОСЬМИ ФЕРЗЯХ

Задача о восьми ферзях заключается в том, чтобы расставить на стандартной 64-клеточной шахматной доске 8 ферзей так, чтобы ни один из них не находился под боем другого. Пример такой расстановки представлен на рисунке 1.

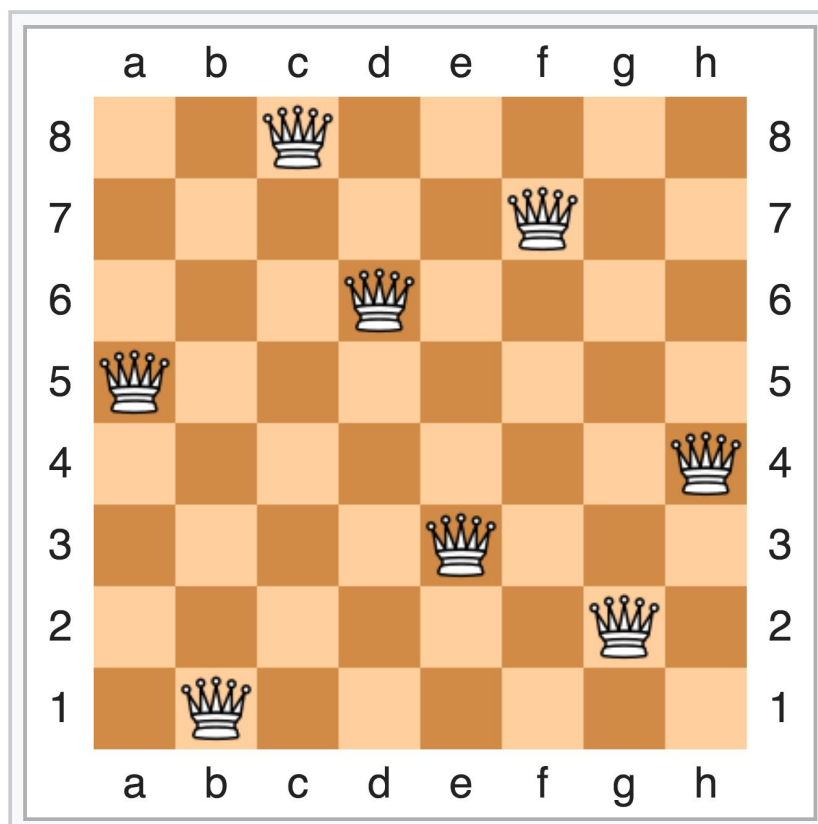


Рисунок 1 - пример решения задачи о 8 ферзях

### 1.1. Метод перебора

В литературе известны различные классические алгоритмы, реализующие традиционный комбинаторный перебор для поиска

решений задачи о расстановке восьми ферзей. Эту задачу решил больше 200 лет тому назад великий математик Леонард Эйлер.

Очевидно, на каждой из 8 вертикалей должно стоять по ферзю. Каждую такую расстановку можно закодировать одномерным массивом  $X[1], \dots, X[8]$ , где  $X[i]$  – номер горизонтали для  $i$ -го ферзя. Поскольку никакие два ферзя не могут стоять на одной горизонтали (тогда они бьют друг друга), то все  $X[i]$  различны, т.е. образуют перестановку из чисел  $1..8$ . Можно, конечно, перебрать все  $8!$  таких перестановок и выбрать среди них те 92, которые нас интересуют. Но число  $8!=40320$  довольно большое. Поэтому обычно используют алгоритм перебора с возвратом, который позволяет значительно сократить перебор и дает ответ намного быстрее. Таким образом, классические алгоритмы имеют ряд ограничений, в частности, долго работают, весьма сложны и трудно модифицируемые

Нами было реализовано классическое решение задачи и использовался алгоритм поиска с возвратом. На каждом этапе данный алгоритм пытается поставить новую фигуру на доску таким образом, чтобы ее не бил ни один ферзь, установленный ранее. Поэтому нет смысла рассматривать в качестве возможных полей горизонтали, вертикали и диагонали, на которых уже есть другие фигуры. Если же поставить ферзя не удалось, происходит возврат к этапу установки предыдущей фигуры и для нее ищется новое свободное поле.

## **1.2. Метод с использованием мультиагентной системы**

Мультиагентная система (МАС) состоит из автономных программных агентов, способных воспринимать ситуацию, принимать решения и взаимодействовать с себе подобными.

Новый этап в развитии информационных технологий связывается с мультиагентными технологиями, которые по своему

потенциалу призваны в скором времени выйти на уровень нано и биотехнологий. Вместе с тем, как показывает практика, разработка такого рода систем распределенного интеллекта, где каждый агент может воспринимать ситуацию, принимать решения и коммуницировать с другими, сама по себе является сложной задачей и, во многом, контр-интуитивной для человеческого мышления.

За счет применения МАС при решении данной задачи были дополнительно введены следующие возможности:

- без кардинальной перестройки метода и перепрограммирования системы, была изменена постановка задачи, введя классы новых фигур (ладьи и слона);
- при желании возможно введение новой фигуры или выведение одной из фигур, участвующей в решении, прямо во время выполнения программы;
- возможно изменение стратегии решения, изменив агентов;

## 2. МУЛЬТИАГЕНТНАЯ СИСТЕМА РЕШЕНИЯ ЗАДАЧИ О ВОСЬМИ ФЕРЗЯХ

Рассмотрим особенности реализации метода с использованием мультиагентной системы к решению данной задачи.

В нашей системе были спроектированы четыре программы-компонента:

- сервер;
- агент-фигура;
- агент-судья;
- монитор.

Коммуникация программ происходит по протоколу пользовательских датаграмм UDP. Диаграммы последовательности запуска системы и решения задачи представлены на рисунках 2 и 3, соответственно.

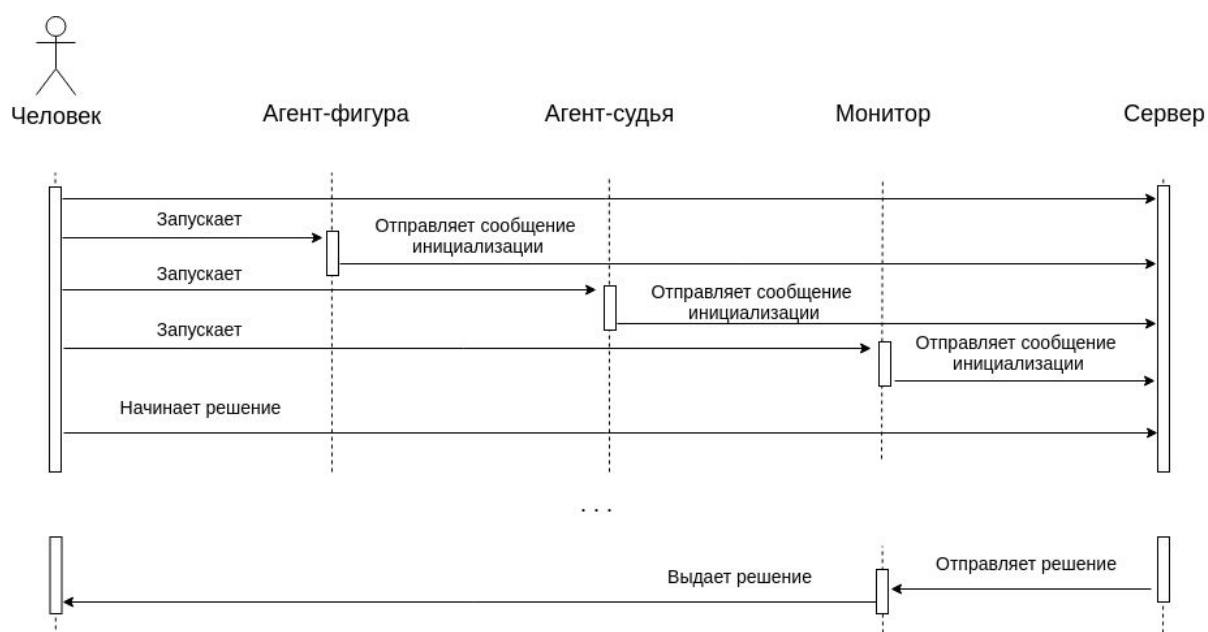


Рисунок 2 - диаграмма последовательности запуска системы

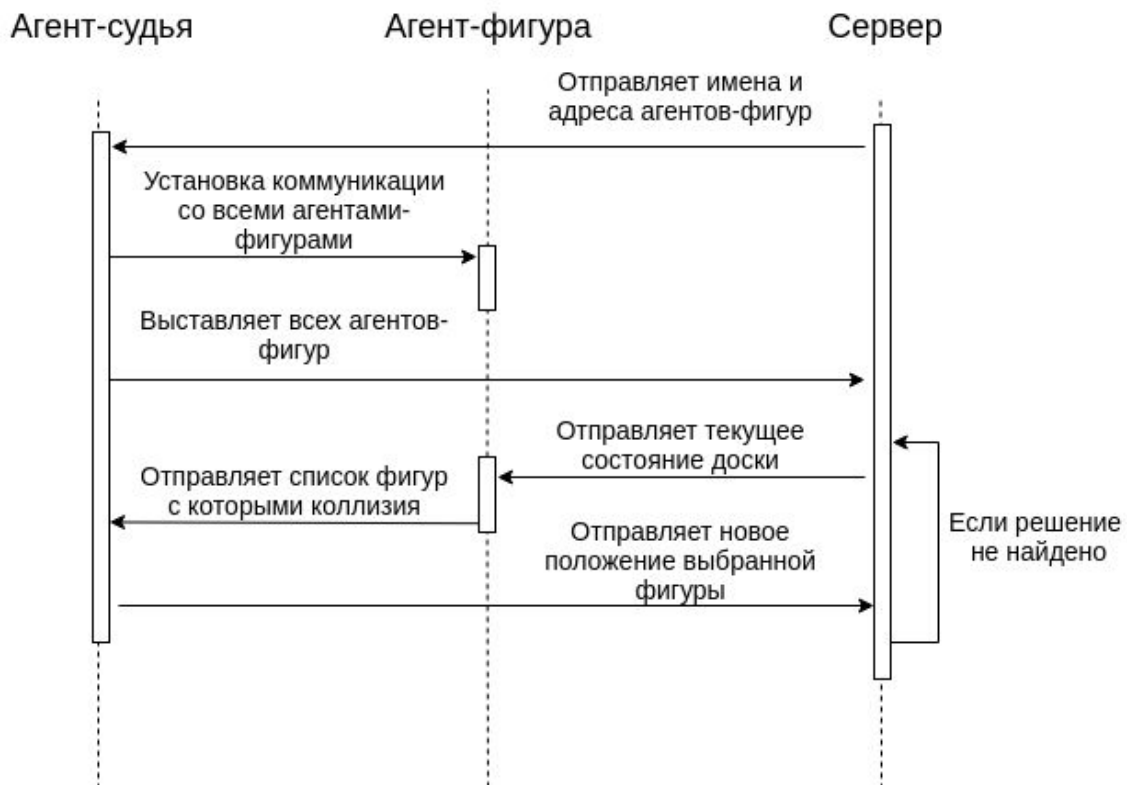


Рисунок 3 - диаграмма последовательности решения задачи

Схема связей компонентов представлена на рисунке 4.

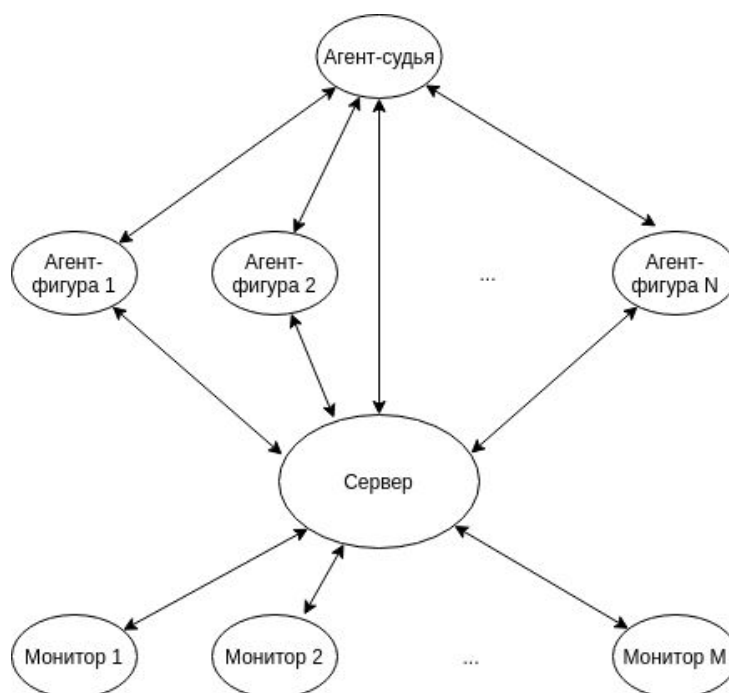


Рисунок 4 - схема связей компонентов системы



## 2.1. Сервер

Сервер представляет собой двухпоточную программу, состоящий из следующих потоков:

- поток чтения запросов - чтение запросов и добавление их в очередь;
- поток исполнения - последовательно берет из очереди запрос, выполняет парсинг команд, исполняет, формирует и отправляет ответы.

Список всех обрабатываемых запросов представлен в таблице 1.

Таблица 1. Обрабатываемые сервером запросы

Запрос	Описание
<i>init &lt;figure_type&gt;</i>	Запрос на инициализацию агента-фигуры. При успешной инициализации отправляет обратно сообщение со сгенерированным именем агента-фигуры:  <i>init ok &lt;figure_name&gt;</i>
<i>init_judge</i>	Запрос на инициализацию агента-судьи. При успешной инициализации в ответ отправляет сообщение:  <i>init_judge ok</i>
<i>set_params &lt;board_width&gt; &lt;board_height&gt;</i>	Запрос на изменение параметров доски. При успешном изменении параметров отправляет обратно сообщение:

	<i>set_params ok</i>
<i>start_solving</i>	<p>Запрос на начало решения. После начала решения отправляет агенту-судье список всех агентов-фигур с их адресами:</p> <p><i>agents (&lt;figure1_name&gt; &lt;figure1_port&gt;) ...</i></p>
<i>change_pos (&lt;figure1_name&gt; &lt;figure1_x&gt; &lt;figure1_y&gt;) ...</i>	<p>Запрос на изменение позиции агента-фигуры. После смены позиции отправляет всем агентам-фигурам обновленное состояние доски:</p> <p><i>board (&lt;figure_name1&gt; &lt;figure_x1&gt; &lt;figure_y1&gt;) ...</i></p>
Остальные	<p>Во всех остальных случаях, когда не удалось определить запрос, в ответ отправляется сообщение об ошибке:</p> <p><i>bad_language</i></p>

## 2.2. Агент-фигура

Агент-фигура - последовательная программа, который работает по алгоритму представленному блок-схемой на рисунке 5.



Рисунок 5 - блок-схема алгоритма работы агента-фигуры

Суть этого агента заключается в поиске всех фигур, которые находятся под боем фигуры, закрепленной за этим агентом. За счет работы агентов на разных процессорах, достигается параллелизм, что особенно эффективно при большом количестве фигур на доске.

### 2.3. Агент-судья

Агент-судья - это программа, цель которой после каждого обновления позиций фигур на доске собрать данные о коллизиях от всех агентов-фигур, а затем выбрать и отправить на сервер фигуру, которую

следует передвинуть на заранее выбранное место (в зависимости от стратегии). К серверу может подключиться только один агент-судья.

Блок-схема алгоритма работы агента-судьи представлена на рисунке 5.

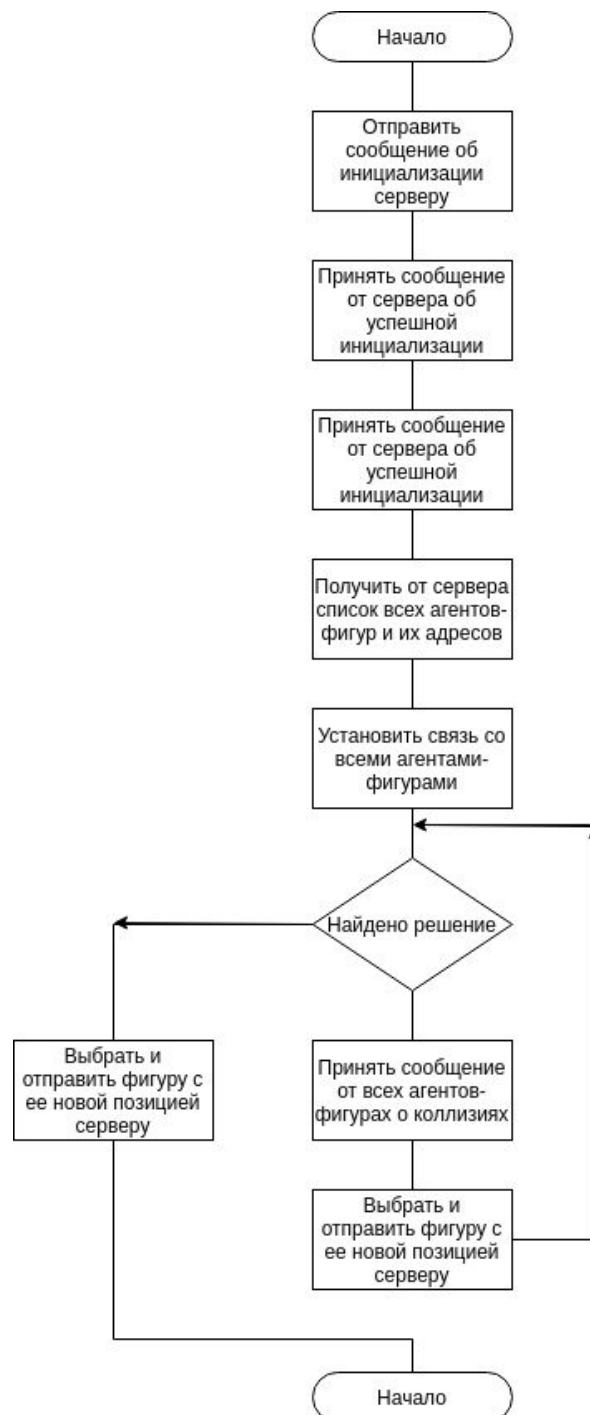


Рисунок 6 - блок-схема алгоритма работы агента-судьи

## **2.4. Монитор**

Монитор - программа для отображения решения системы, может быть подключено одновременно несколько мониторов. Реализован на двух потоках:

- Поток взаимодействия с пользователем: визуализирует доску, считывает нажатие клавиш, отправляет запросы серверу;
- Поток взаимодействия с сервером: принимает от сервера информацию об изменении доски, обновляет доску.

### 3. ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ

Для подведения итогов, были проведены эксперименты, которые должны показать корректность работы системы при решении задачи о восьми ферзях.

Скриншоты решений, которые нашла система представлены на рисунках 7-9.

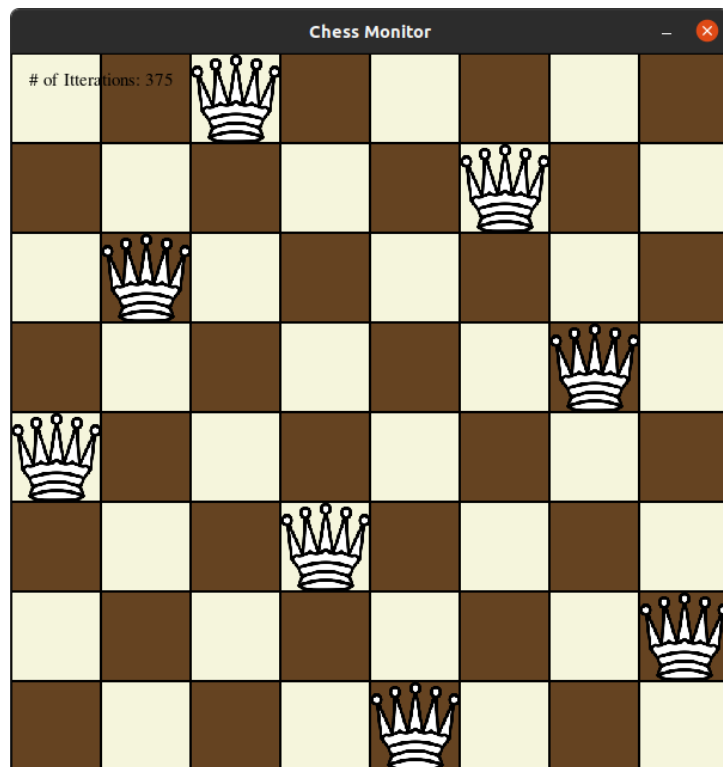


Рисунок 7 - решение задачи о 8 ферзях на доске 8x8

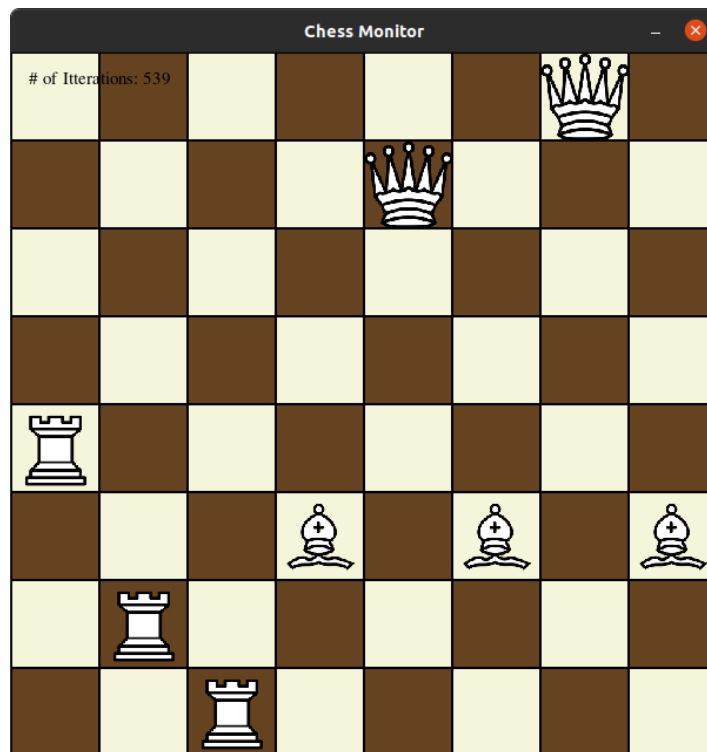


Рисунок 8 - решение задачи о 2 ферзях, 3 слонах и 3 ладьях на доске 8x8

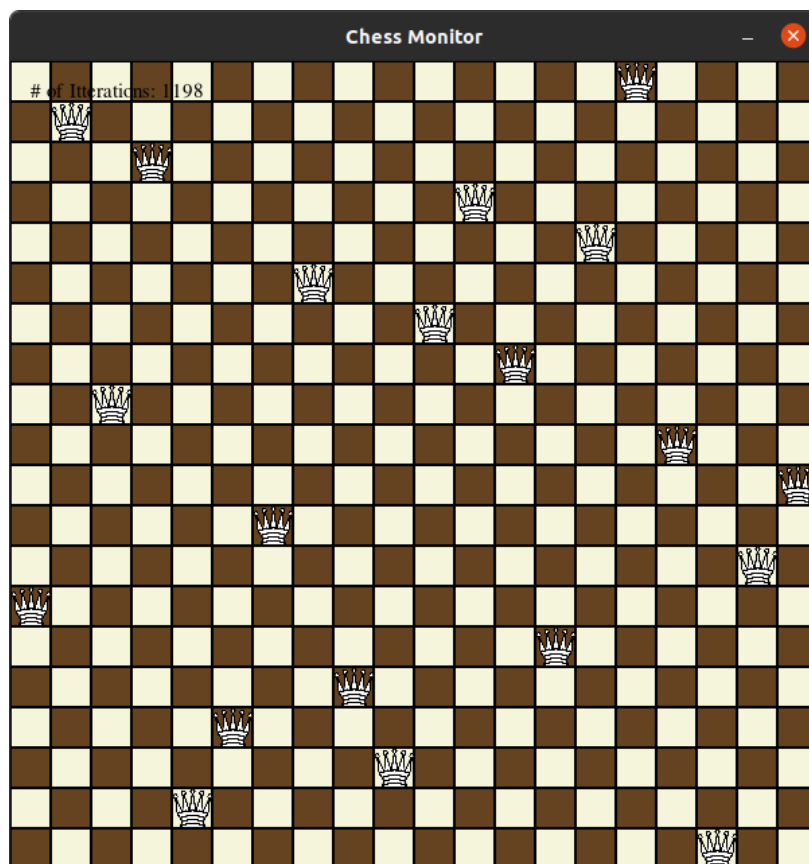


Рисунок 9 - решение задачи о 20 ферзях на доске 20x20

## **ВЫВОД**

В ходе выполнения курсовой работы был разработан и реализован подход к решению классической задачи расстановки ферзей, основанный на использовании мультиагентных систем.

Данный метод является достаточно сложным в плане программной реализации из-за лежащей в основе клиент-серверной архитектуры, однако его использование позволяет добиться значительного увеличения скорости решения задачи по сравнению с классическими алгоритмами. Так, на наборах с 8, 10 и 12 ферзями реализованная система работала в среднем быстрее алгоритма поиска с возвратом в 10-20 раз. А для нахождения взаимной расстановки 20 ферзей на поле была затрачена 1 минута, когда стандартный метод не дал результата после 30 минут работы и был приостановлен.

Стоит отметить, что рассматриваемый подход, ввиду индивидуальной программной реализации и автономной работы каждой отдельной фигуры, имеет очень широкие возможности по дальнейшему усовершенствованию и введению новых стратегий. С целью наглядной демонстрации этого преимущества, в курсовой работе была решена задача по взаимной расстановке 2 ферзей, 3 ладей и 3 слонов.

Для удобного восприятия результатов поиска взаимоположения фигур на доске, в рамках курсовой работы, был также реализован графический интерфейс пользователя.



## **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. В. Олифер, Н. Олифер. Компьютерные сети. Принципы, технологии, протоколы. Учебник, 2016 - 822 с.
2. С. Рассел, П. Норвиг. Искусственный интеллект: современный подход, 2-е изд. : Пер с англ - М. : ООО “И. Д. Вильямс”, 2016 - 1408 с.
3. M. Wooldridge and N. R. Jennings. Agent Theories, Architectures, and Languages: A Survey. In: Intelligent Agents. ECAI-94 Workshop on Agent Theories, Architecture and Languages. Amsterdam, The Netherlands, August 8-9, 1994, (Eds. M. J. Wooldridge and N. R. Jennings). Proceedings. Springer Verlag: 3-39, 1994