**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
 РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФГБОУ ВО «БРЯНСКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Кафедра** «Информатика и программное обеспечение»

**КУРСОВАЯ РАБОТА**

По дисциплине **«**Системы искусственного интеллекта**»**

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ   
НЕИНФОРМИРОВАННОГО ПОИСКА   
И ЭВРИСТИЧЕСКОГО ПОИСКА

Выполнил студ. гр. О-21-ИВТ-ПО-Б

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Бахтин Г. А.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_к.т.н., доц. Исаев Р.А.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Брянск 2024

Содержание

[1. Описание задачи поиска 3](#_Toc183517846)

[2. Исследуемые алгоритмы поиска 4](#_Toc183517847)

[2.1. Эвристика 1 4](#_Toc183517848)

[2.2. Эвристика 2 4](#_Toc183517849)

[2.3. Сравнение эвристик 5](#_Toc183517850)

[2.3.1. Доминирование 5](#_Toc183517851)

[3. Исследование зависимости показателей поиска от сложности задачи 7](#_Toc183517852)

[3.1. Примеры начальных состояний 7](#_Toc183517853)

[3.2. Таблицы 9](#_Toc183517854)

[3.3. Графики 10](#_Toc183517855)

[3.4. Выводы 11](#_Toc183517856)

[4. Дополнительное исследование эффективности эвристик 12](#_Toc183517857)

[4.1. Исследование эффективности эвристики 1 12](#_Toc183517858)

[4.2. Исследование эффективности эвристики 2 13](#_Toc183517859)

[5. Выводы 15](#_Toc183517860)

# Описание задачи поиска

В качестве решаемой задачи поиска выступает головоломка «Сокобан».

Игроку необходимо расставить ящики по обозначенным местам лабиринта (меткам), толкая их по одному в разные стороны. Карта представлена в виде решётки, в ячейках которой могут находиться стены или ящики. Игрок может перемещаться только в ячейки, которые имеют общую сторону с текущей.

На рис. 1 и 2 приведены одно из возможных начальных состояний и целевое состояние соответственно.

Изображение выглядит как снимок экрана, шаблон, Прямоугольник, прямоугольный

Автоматически созданное описание

Рис. 1. Возможное начальное состояние

Изображение выглядит как снимок экрана, Прямоугольник, прямоугольный, шаблон

Автоматически созданное описание

Рис. 2. Целевое состояние

# Исследуемые алгоритмы поиска

Были рассмотрены следующие алгоритмы поиска:

* Поиск в ширину;
* Поиск с итеративным углублением;
* Эвристический поиск А\*.
* В эвристическом поиске использовались эвристики 1 и 2.

## Эвристика 1

Принцип работы данной эвристики заключается в подсчёте количества ящиков, которые не находятся на метках.

Листинг

Эвристика 1

|  |
| --- |
| static public int MidHeuristic(State state)  {  int counter = 0;  foreach ((byte x, byte y) b in state.boxes)  {  if (Sokoban.map.GetCell(b.y, b.x) != (byte)Block.Type.Mark)  {  counter++;  }  }  return counter;  } |

Данная эвристика является допустимой, так как сумма ящиков не на метках никогда не сможет превзойти количество ходов, требуемое для расставления данных ящиков на метки.

Эвристика получена из ослабленной задачи с правилом: коробка А может быть поставлена на целевую метку Б за 1 ход

## Эвристика 2

Принцип работы данной эвристики заключается в подсчёте суммы кратчайших расстояний городских кварталов от коробок до меток.

Листинг

Эвристика 2

|  |
| --- |
| static public int BetterHeuristic(State state)  {  int res = 0;  (byte x, byte y)[] marks = state.map.marks;  foreach ((byte x, byte y) b in state.boxes)  {  List<int> dists = new();  foreach ((byte x, byte y) m in marks)  {  dists.Add(Sphere.Dist((m.x, m.y), (b.x, b.y)));  }  res += dists.Min();  }  return res;  } |

Реализация подсчёта расстояния городских кварталов приведено в листинге 3.

Листинг 3

Расстояние городских кварталов

public class Sphere

{

public static int Dist((byte x, byte y) p, (byte x, byte y) c)

{

return Math.Abs(p.x - c.x) + Math.Abs(p.y - c.y);

}

}

Данная эвристика является допустимой, так как сумма кратчайших расстояний городских кварталов от коробок до меток никогда не превосходит количество ходов, требуемое для расставления ящиков на метки.

Эвристика получена из ослабленной задачи с правилом: коробка А может быть поставлена грузчиком на целевую метку Б, при этом стены игнорируются.

## Сравнение эвристик

Эвристика 1 даёт слишком маленькое количество различных оценок, из-за чего большое количество времени поиск, использующий данную эвристику, будет вести себя как поиск в ширину. Тем временем эвристика 2 даёт гораздо большее количество различных оценок, и они больше приближены к настоящему количеству ходов, необходимому для решения задачи.

### Доминирование

Если *h*1(*n*), *h*2(*n*) — допустимые эвристические функции, и для любого узла *n* верно неравенство *h*1(*n*) ≥ *h*2(*n*), то *h*1 доминирует над *h*2.

Неравенство выполняется при *h*2=эвристика 1, *h*1=эвристика 2, так как:

1. В случаях, где кратчайшие расстояния от коробок до меток хотя бы в одном случае больше 1, оценка *h*1 будет больше *h*2.
2. В случаях, где кротчайшие расстояния от коробок до меток везде равно 1, оценка h1 будет равна h2.

Следовательно, эвристика 2 доминирует над эвристикой 1.

# Исследование зависимости показателей поиска от сложности задачи

Для сравнения алгоритмов проведем ряд тестов, отличающихся сложностью задачи, под которой понимается стоимость ее оптимального решения (*d*). Будут исследованы следующие значения *d*: 2, 7, 12, 17, 22. Для каждого значения *d* будет проведено по 10 тестов, отличающихся начальными состояниями, и будут получены средние значения следующих показателей.

1. Число итераций поиска, необходимых для нахождения решения.
2. Максимальное количество узлов, одновременно хранимых в памяти в ходе выполнения поиска.
3. Эффективный коэффициент ветвления *b\** (из тестов находится среднее значение N, и используя его с помощью WolframAlpha вычисляется *b\**).

Тесты проводятся на игровом поле размером 12х12. Максимальный коэффициент ветвления в игре «Сокобан» равен 4-ём не зависимо от размеров поля. Также проведём отдельный тест на проверку работы алгоритмов в случае нерешаемой задачи.

Начальные состояния для тестов создавались вручную.

## Примеры начальных состояний

Ниже приведены примеры начальных состояний для каждого *d*.

A screenshot of a video game

Description automatically generated

Рис. 3. Пример начального состояния, где d=2

A screenshot of a video game

Description automatically generated

Рис. 4. Пример начального состояния, где d=7

A screenshot of a video game

Description automatically generated

Рис. 5. Пример начального состояния, где d=12

A screenshot of a video game

Description automatically generated

Рис. 6. Пример начального состояния, где d=17

A screenshot of a video game

Description automatically generated

Рис. 7. Пример начального состояния, где d=22

A screenshot of a video game

Description automatically generated

Рис. 8. Начальное состояние нерешаемого уровня

Для нерешаемого уровня количество итераций всех поисков было одинаковым и составило 992, решение найдено не было.

## Таблицы

Ниже приведена таблица, в которой отображены средние значения показателей поисков для каждого d.

Таблица

Показатели для каждого поиска с различными d

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Поиск | Показатель | d | | | | |
| 2 | 7 | 12 | 17 | 22 |
| Поиск в ширину | Итерации | 8.1 | 258.5 | 638.9 | 4332.444 | 8274.6 |
| N | 17.6 | 503.2 | 988 | 6610.222 | 11533.8 |
| b\* | 3.725 | 2.236 | 1.643 | 1.582 | 1.451 |
| Поиск с итеративным углбением | Итерации | 8.444 | 169.111 | 681.222 | 6057.571 | 8195.2 |
| N | 10.667 | 272.222 | 759.556 | 6913.571 | 9909.5 |
| b\* | 2.804 | 2.023 | 1.602 | 1.587 | 1.44 |
| Эвристика 1 | Итерации | 3.5 | 91.2 | 258.9 | 2543.111 | 4318.6 |
| N | 6.8 | 165.4 | 398.3 | 3876.111 | 6038.6 |
| b\* | 2.155 | 1.862 | 1.505 | 1.527 | 1.404 |
| Эвристика 2 | Итерации | 3.5 | 66.8 | 197.1 | 1187.1 | 1734 |
| N | 6.8 | 114.6 | 283 | 1947 | 2352 |
| b\* | 2.155 | 1.749 | 1.454 | 1.459 | 1.337 |

## Графики

Ниже приведены графики, построенные на основе предыдущей таблицы.

График 1

График 2

График 3

## Выводы

Исходя из графиков можно сделать следующие выводы:

* Нельзя однозначно сказать, что поиск с итеративным углублением лучше поиска в ширину или наоборот.
* Поиск A\* с эвристикой 1 лучше поиска в ширину.
* Поиск A\* с эвристикой 1 лучше поиска с итеративным углублением.
* Теоретическое заключение превосходства эвристики 2 над эвристикой 1 было подтверждено на практике.

# Дополнительное исследование эффективности эвристик

## Исследование эффективности эвристики 1

Эвристика 1 эффективна в тех случаях, когда коробки, поставленные на метки, не нужно перемещать в последующих ходах.

Ситуации, в которых коробки, поставленные на метки, нужно перемещать в последующих ходах, для эвристики 1 будут менее «подходящими». Эвристика будет запутываться, потому что поиск с ней сначала обойдёт все состояния, где коробка стоит на метке, решение найдено не будет, алгоритм сдвинет эту коробку и поиск целевого состояния продолжится.

На рис. 9 приводится пример с уровнем, для решения которого коробки на метках не нужно двигать в последующих ходах.

A screenshot of a video game

Description automatically generated

Рис. 9. «Подходящий» уровень

На рис. 10 приводится пример с уровнем, для решения которого приходится двигать коробку на метке.

A screenshot of a video game

Description automatically generated

Рис. 10. «Неподходящий» уровень

Значения показателей приведены в таблице ниже.

|  |  |  |
| --- | --- | --- |
|  | «Подходящий» | «Неподходящий» |
| d | 18 | 18 |
| Количество итераций | 3955 | 28526 |
| N | 7285 | 47312 |
| b\* | 1.54 | 1.73 |

Полученные данные подтверждают описанную выше гипотезу.

## Исследование эффективности эвристики 2

Эвристика 2, которая считает сумму кратчайших расстояний городских кварталов от коробок до меток, будет эффективной в ситуациях, когда для расстояния городских кварталов между коробками и метками не будут стоять стены.

Ситуации, в которых стены «мешают» расстояниям городских кварталов, для эвристики 2 будут менее «подходящими». Эвристика будет запутываться, потому что алгоритм будет двигать коробку к ближайшей метке, однако целевое состояние не будет достигнуто из-за преграды в виде стены.

На рис. 11 приводится пример с уровнем, где нет стен.

A screenshot of a video game

Description automatically generated

Рис. 11. «Подходящий» уровень

На рис. 12 приводится пример с уровнем, где стены мешают эвристике 2.

A screenshot of a video game

Description automatically generated

Рис. 12. «Неподходящий» уровень

Значения показателей приведены в таблице ниже.

|  |  |  |
| --- | --- | --- |
|  | «Подходящий» | «Неподходящий» |
| d | 19 | 19 |
| Количество итераций | 205 | 1597 |
| N | 539 | 1978 |
| b\* | 1.21 | 1.40 |

Полученные данные подтверждают описанную выше гипотезу.

# Выводы

В курсовой работе было проведено исследование четырех алгоритмов поиска в пространстве состояний: поиск в ширину, поиск с итеративным углублением, поиск A\* с использованием эвристики 1, поиск A\* с использованием эвристики 2. В качестве задачи поиска выступала головоломка «Сокобан».

В результате исследования сделаны следующие выводы:

1. Эвристика 2 доминирует над эвристикой 1, что было подтверждено на практике.
2. Поиск с итеративным углублением не лучше поиска в ширину.
3. Поиск A\* с эвристикой 1 лучше поиска в ширину.
4. Поиск A\* с эвристикой 1 лучше поиска с итеративным углублением.
5. Эвристика 1 показывает себя хуже, если коробку на метке нужно двигать в последующих ходах.
6. Эвристика 2 показывает себя хуже, если между коробками и метками стоят стены.