Problem Chosen	2024 MCM/ICM Summary Sheet	Team Control Number
B		2421255

my title

Summary

my abstract

Keywords: keyword1; keyword2

Contents

1 Introduction **2**

1.1 background 2

1.2 Restatement of the problem 2

1.3 Our work 2

2 Assumptions and Justifications **3**

3 Notations **3**

4 Model 1 : **4**

4.1 1 4

5 Sensitivity Analysis **4**

6 Evaluate of the Mode and Further Discussion **4**

7 Conclusions **4**

8 Strengths and weaknesses **4**

8.1 Strengths 4

8.2 How to cite? 4

Appendices **5**

Appendix A First appendix **5**

Appendix B Second appendix **5**

1 Introduction

1.1 background

In the 21st century, there has been a growing interest in the exploration of the ocean in various countries and regions. Maritime Cruises Mini-Submarines (MCMS), a Greek company, specializes in manufacturing submersibles to take people to the deepest parts of the ocean. They have set their sights on leading tourists on exciting adventures to explore sunken shipwrecks in the depths of the Ionian Sea.

To make their submersible operations a reality, we help MCMS to obtain regulatory approval and establish safety protocols to address potential communication loss and mechanical issues, such as propulsion failure. And we developing a predictive model that can track the submersible's location over time. This model needs to consider factors such as sea floor positioning, buoyancy, currents, sea density , and geography.

1.2 Restatement of the problem

Considering the background information and restricted conditions identified in the problem statement,we need to solve the following problems:

- Create models predicting submersible location over time.
- Explore and reduce the uncertainty of these models.Find out what information the submersible needs to send regularly to the host ship and what equipment is needed to do this
- Consider the costs associated with availability, maintenance, readiness, and usage when determining what additional search equipment to carry on both the host ship and rescue ship if necessary.
- Develop a model using location data to recommend deployment points and search patterns, aiming to minimize the time to locate a lost submersible.
- Establish a function that relates the probability of finding the submersible to both time and the accumulated search results.
- Extend the model to cover other tourist destinations and adapt it for multiple submersibles moving in the same general vicinity.

1.3 Our work

to my eq (**??**)

- the angular velocity of the bat,
- the velocity of the ball, and
- the position of impact along the bat.

center of percussion [Brody 1986]

Theorem 1.1. *LaTeX*

Lemma 1.2. *TeX.*

Proof. The proof of theorem. □

2 Assumptions and Justifications

Considering that practical problems always contain many complex factors, first of all, we need to make reasonable assumptions to simplify the model, and each hypothesis is closely followed by its corresponding explanation

- **Assumptions 1: Rationalize the assumptions for the submersible by considering it as a point mass.**

  When moving in the deep sea, the size and shape of the submersible are negligible compared to the problem we are studying, and the movement of each point on the submersible can be considered to be the same. Therefore, the submersible can be represented by a mass point, following the principle of the center of mass theorem.

- **Assumptions 2: Assume that the submarine is unable to provide propulsion.**

  Unless under extreme special circumstances, the submersible pilot will cease all operations and await rescue after experiencing distress. Therefore, it can be inferred that the submersible is powerless on its own once in distress.

- **Assumptions 3: Excluding the submarine itself, the decision factors include ocean currents, density, and topography.**

  Based on the principles of fluid mechanics, the motion of the submersible is primarily influenced by water flow and its own dynamics. Ocean currents play a significant role in governing the movement of water bodies in the ocean, and both density and topography are factors associated with the flow of water bodies.so these factors are emphasized and other factors are ignored.

- **Assumptions 4: Assume that there are no other interfering factors except ocean currents, density, and topography.**

  The movement of the submersible is mainly affected by water flow and itself. Ocean currents are mainly related to factors such as the atmosphere and the rotation of the earth. Water density is usually related to factors such as water temperature and salinity, and these factors have minimal impact on the terrain.

3 Notations

The key mathematical notations used in this paper are listed in Table 1

Symbol	Definitions	Units

Table 1: Notations used in this paper

4 Model 1 :

4.1 1

5 Sensitivity Analysis

6 Evaluate of the Mode and Further Discussion

7 Conclusions

8 Strengths and weaknesses

8.1 Strengths

- **Applies widely**
  This system can be used for many types of airplanes, and it also solves the interference during the procedure of the boarding airplane,as described above we can get to the optimization boarding time.We also know that all the service is automate.

- **Improve the quality of the airport service**
  Balancing the cost of the cost and the benefit, it will bring in more convenient for airport and passengers.It also saves many human resources for the airline.

8.2 How to cite?

bibliography cite use [1, 2, 3]

AI cite use [AI: 1, 2, 3]

References

[1] D. E. KNUTH The TEXbook the American Mathematical Society and Addison-Wesley Publishing Company , 1984-1986.

[2] Lamport, Leslie, LATEX: “ A Document Preparation System ”, Addison-Wesley Publishing Company, 1986.

[3] `https://www.latexstudio.net/`

[4] as,disadvantages,dsaw,fdsa

Appendices

Appendix A First appendix

Dear, Mr. Alpha Chiang

Sincerely yours,

Your friends

Input matlab source:

```
function [t,seat,aisle]=OI6Sim(n,target,seated)
pab=rand(1,n);
for i=1:n
    if pab(i)<0.4
        aisleTime(i)=0;
    else
        aisleTime(i)=trirnd(3.2,7.1,38.7);
    end
end
```

Appendix B Second appendix

MEMORANDUM

To: MCM office

From: MCM Team 2421255

Subject: MCM

Date: February 3, 2024

This is a memorandum.

some more text **Input C++ source:**

```
//============================================================================
// Name        : Sudoku.cpp
// Author      : lst
// Version     : a.0
// Copyright   : Your copyright notice
// Description : Sudoku in C++.
//============================================================================

#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int table[9][9];

int main() {

    for(int i = 0; i < 9; i++){
        table[0][i] = i + 1;
    }

    srand((unsigned int)time(NULL));

    shuffle((int *)&table[0], 9);

    while(!put_line(1))
    {
        shuffle((int *)&table[0], 9);
    }

    for(int x = 0; x < 9; x++){
        for(int y = 0; y < 9; y++){
            cout << table[x][y] << " ";
        }

        cout << endl;
    }

    return 0;
}
```

Report on Use of AI

1. Bing AI
   Query1: write a sort algorithm
   Output: Sure, I can help you with that. Here is a simple implementation of the selection sort algorithm in Python:

```
def selection_sort(arr):
    for i in range(len(arr)):
        min_idx = i
        for j in range(i + 1, len(arr)):
            if arr[min_idx] > arr[j]:
                min_idx = j
        arr[i], arr[min_idx] = arr[min_idx], arr[i]
    return arr
```

   This algorithm sorts an array by repeatedly finding the minimum element from the unsorted part of the array and putting it at the beginning. The time complexity of this algorithm is O(n^2).

   There are many other sorting algorithms available such as bubble sort, insertion sort, merge sort, quick sort, heap sort, and more. Each of these algorithms has its own advantages and disadvantages. You can learn more about them on GeeksforGeeks and Programiz.

   I hope this helps!

2. OpenAl Enie (Nov 5, 2023 version, Ernie 4.0)
   Query1: <insert the exact wording of any subsequent input into the Al tool>
   Output: <insert the complete output from the second query>

3. Github CoPilot (Feb 3, 2024 version)
   Query1: <insert the exact wording you input into the Al tool>
   Output: <insert the complete output fron the Al tool>

4. Google Bard (Feb 2,2024 version)
   Query: <insert the exact wording of your query>
   Output: <irsert the complete output from the Al tool>

5. Bing AI
   Query: write a FFT algorithm
   Output: Sure, here is a Python implementation of the Fast Fourier Transform (FFT) algorithm:

```
import numpy as np

def fft(x):
    n = len(x)
    if n == 1:
        return x
    even = fft(x[0::2])
    odd = fft(x[1::2])
    T = [np.exp(-2j * np.pi * k / n) * odd[k] for k in range(n // 2)]
    return [even[k] + T[k] for k in range(n // 2)] + [even[k] - T[k] for k
    ↪ in range(n // 2)]
```

The FFT is a widely used algorithm for computing the **Discrete Fourier Transform (DFT)** of a sequence. It reduces the number of computations needed from $O(n^2)$ to $O(n \log n)$ 12. The basic idea of the FFT is to apply divide and conquer. We divide the coefficient vector of the polynomial into two vectors, recursively compute the DFT for each of them, and combine the results to compute the DFT of the complete polynomial 3.

I hope this helps!